Programmable Logic Devices

DATA HANDBOOK

BOOK IC13 1994

Philips Semiconductors

PHILIPS

Programmable Logic Devices

CONTENTS

Preface

The 1994 Philips Semiconductors Programmable Logic Devices Data Handbook contains all the information a designer needs to implement virtually any logic design in a PLD.

Philips offers a variety of programmable logic devices which have been tailored to implement a wide range of applications.

In addition to a full complement of industry standard PAL-type devices, Philips offers several application specific devices.

For ultra high speed applications, the Philips PLAs are the right solution. The logical power of two programmable arrays makes complex decoding easy. The PLUS153 and the PLUS173 offer 32 input wide AND-OR decoding, with a worst case T_{PD} of 10ns.

Designers using high performance DRAM, VRAM and graphics devices will appreciate the variety of complex state machines which include the PLC42VA12, PLC415, PLUS405 and the PLUS105. Check out the new 70 MHz PLUS105 in Section 5.

The PLC18V8Z can replace virtually all 20 PALs and GALs. The zero standby power of 20 micro amps is the lowest of any available PLD.

For maximum density in truly compact systems, the Programmable Macro Logic family is the right choice. These field programmable gate arrays are ideal for bus interface applications which require very wide gates.

The newest addition to the Philips PLD line is a family of low voltage devices (2.7 to 3.6 volts). Refer to Section 6 for details on the zero standby power P3C18V8Z Universal PAL device and the 3 very high speed BiCMOS devices.

Designing and programming of Philips PLDs is easy; many third party software and programmer vendors support the entire line of Philips devices. Section 10 details information on the approved software and programmer vendors, as well as all the latest software and firmware revisions.

For ideas on how to implement your designs, take a look at all the application notes in Section 11 of this manual.

Philips provides a high level of customer service via our 24-hour computer bulletin board and our staff of applications engineers who are ready to answer all your design and software related questions. The toll free bulletin board number (from the USA only) is (800) 451-6644. It may also be reached at (408) 991-2406.

Section 1
General Information

CONTENTS

Contents

Contents

Contents

Alphanumeric index

Product Status

DEFINITIONS

Data Sheet Identification	Product Status	Definition
Objective Specification	Formative or in Design	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	Preproduction Product	This data sheet contains preliminary data, and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	Full Production	This data sheet contains Final Specifications. Signetics reserves the right to make changes at any time without notice, in order to improve design and supply the best possible product.

Selection guide

PHILIPS SEMICONDUCTORS PART NUMBER	ARCHITECTURE (Inputs × Terms* × Outputs)	PACKAGE	TOTAL INPUTS (# Dedicated)	PRODUCT TERMS PER OR GATE	INTERNAL STATE REGISTERS (# Dedicated)	OUTPUTS C, I/O, R, R I/O	t_{PD} (Max)	f_{MAX}	I_{CC} (Max)
PAL DEVICES									
10H20EV8–4/ 10020EV8–4	20 × 90 × 8	24-Pin	20 (12)	8 to 12	0	8 varied	4.5ns	208MHz	–250mA
PHD16N8–5	16 × 16 × 8	20-Pin	16 (10)	1♦	0	2 C, 6 I/O	5ns		180mA
PHD48N22–7	48 × 73 × 22	68-Pin	48 (36)	7 to 12	0	10 C, 12 I/O	7.5ns		420mA
PLUS16L8–7	16 × 64 × 8	20-Pin	16 (10)	7	0	2 C, 6 I/O	7.5ns		180mA
PLUS16R4–7	16 × 64 × 8	20-Pin	16 (8)	7 to 8	4 (0)	4 I/O, 4 R	7.5ns	74MHz	180mA
PLUS16R6–7	16 × 64 × 8	20-Pin	16 (8)	7 to 8	6 (0)	2 I/O, 6 R	7.5ns	74MHz	180mA
PLUS16R8–7	16 × 64 × 8	20-Pin	16 (8)	8	8 (0)	8 R		74MHz	180mA
PLUS16L8D	16 × 64 × 8	20-Pin	16 (10)	7	0	2 C, 6 I/O	10ns		180mA
PLUS16R4D	16 × 64 × 8	20-Pin	16 (8)	7 to 8	4 (0)	4 I/O, 4 R	10ns	60MHz	180mA
PLUS16R6D	16 × 64 × 8	20-Pin	16 (8)	7 to 8	6 (0)	2 I/O, 6R	10ns	60MHz	180mA
PLUS16R8D	16 × 64 × 8	20-Pin	16 (8)	8	8 (0)	8 R		60MHz	180mA
PLUS20L8–7	20 × 64 × 8	24-Pin	20 (14)	7	0	2 C, 6 I/O	7.5ns		210mA
PLUS20R4–7	20 × 64 × 8	24-Pin	20 (12)	7 to 8	4 (0)	4 I/O, 4 R	7.5ns	74MHz	210mA
PLUS20R6–7	20 × 64 × 8	24-Pin	20 (12)	7 to 8	6 (0)	2 I/O, 6 R	7.5ns	74MHz	210mA
PLUS20R8–7	20 × 64 × 8	24-Pin	20 (12)	8	8 (0)	8 R		74MHz	210mA
PLUS20L8D	20 × 64 × 8	24-Pin	20 (14)	7	0	2 C, 6 I/O	10ns		210mA
PLUS20R4D	20 × 64 × 8	24-Pin	20 (12)	7 to 8	4 (0)	4 I/O, 4R	10ns	60MHz	210mA
PLUS20R6D	20 × 64 × 8	24-Pin	20 (12)	7 to 8	6 (0)	2 I/O, 6 R	10ns	60MHz	210mA
PLUS20R8D	20 × 64 × 8	24-Pin	20 (12)	8	8 (0)	8 R		60MHz	210mA
ABT22V10-7	22 × 130 × 10	24-Pin	22 (12)	8 to 16	10 (0)	10 varied	7.5ns	87MHz	210mA
PL22V10-15/I15	22 × 130 × 10	24-Pin	22 (12)	8 to 16	10 (0)	10 varied	15ns	53MHz	90mA, 0.5mA/MHz
PL22V10–12	22 × 130 × 10	24-Pin	22 (12)	8 to 16	10 (0)	10 varied	12ns	67MHz	90mA, 0.5mA/MHz
PL22V10–10	22 × 130 × 10	24-Pin	22 (12)	8 to 16	10 (0)	10 varied	10ns	77MHz	110mA, 0.5mA/MHz
PLC18V8Z35 PLC18V8ZI	18 × 74 × 8	20-Pin	18 (8)	8	8 (0)	8 varied	35, 40ns	21MHz	100µA, 1.5mA/MHz
PLC18V8Z25 PLC18V8ZAI	18 × 74 × 8	20-Pin	18 (8)	8	8 (0)	8 varied	25ns	30MHz	100µA, 1.5mA/MHz
P3C18V8Z25 # P3C18V8ZAI #	18 × 74 × 8	20-Pin	18 (8)	8	8 (0)	8 varied	35ns 40ns	25MHz 20MHz	60µA, 0.8mA/MHz
P3C16V8-7 + #	18 × 64 × 8	20-Pin	18 (10)	7	8 (0)	8 varied	7.5ns	100MHz	150mA
P3C20V8-7 + #	22 × 64 × 8	24-Pin	22 (14)	7	8 (0)	8 varied	7.5ns	100MHz	150mA
LVT22V10-7 + #	22 × 130 × 10	28-Pin	22 (12)	8 to 16	10 (0)	10 varied	7.5ns	100MHz	150mA
PLA									
PLS100/101	16 × 48 × 8	28-Pin	16 (16)	Up to 48	0	8 C	50ns		170mA
PLS153	18 × 42 × 10	20-Pin	18 (8)	Up to 32	0	10 I/O	40ns		155mA
PLS153A	18 × 42 × 10	20-Pin	18 (8)	Up to 32	0	10 I/O	30ns		155mA
PLUS153B	18 × 42 × 10	20-Pin	18 (8)	Up to 32	0	10 I/O	15ns		200mA
PLUS153D	18 × 42 × 10	20-Pin	18 (8)	Up to 32	0	10 I/O	12ns		200mA
PLUS153–10	18 × 42 × 10	20-Pin	18 (8)	Up to 32	0	10 I/O	10ns		200mA
PLS173	22 × 42 × 10	24-Pin	22 (12)	Up to 32	0	10 I/O	30ns		170mA
PLUS173B	22 × 42 × 10	24-Pin	22 (12)	Up to 32	0	10 I/O	15ns		200mA
PLUS173D	22 × 42 × 10	24-Pin	22 (12)	Up to 32	0	10 I/O	12ns		200mA
PLUS173–10	22 × 42 × 10	24-Pin	22 (12)	Up to 32	0	10 I/O	10ns		210mA

Selection guide

PHILIPS SEMICONDUCTORS PART NUMBER	ARCHITECTURE (Inputs × Terms* × Outputs)	PACKAGE	TOTAL INPUTS (# Dedicated)	PRODUCT TERMS PER OR GATE	INTERNAL STATE REGISTERS (# Dedicated)	OUTPUTS C, I/O, R, R I/O	t_{PD} (Max)	f_{MAX}	I_{CC} (Max)
PLS									
PLS105	22 × 48 × 8	28-Pin	22 (16)	Up to 48	6 (6)	8 R		14MHz	180mA
PLS105A	22 × 48 × 8	28-Pin	22 (16)	Up to 48	6 (6)	8 R		20MHz	180mA
PLUS105–45	22 × 48 × 8	28-Pin	22 (16)	Up to 48	6 (6)	8 R		45MHz	200mA
PLUS105–55	22 × 48 × 8	28-Pin	22 (16)	Up to 48	6 (6)	8 R		55MHz	200mA
PLUS405–37	24 × 64 × 8	28-Pin	24 (16)	Up to 64	8 (8)	8 R		37MHz	225mA
PLUS405–45	24 × 64 × 8	28-Pin	24 (16)	Up to 64	8 (8)	8 R		45MHz	225mA
PLUS405–55	24 × 64 × 8	28-Pin	24 (16)	Up to 64	8 (8)	8 R		55MHz	225mA
PLS155	16 × 45 × 12	20-Pin	16 (4)	Up to 32	4 (0)	8 I/O, 4 R I/O	50ns	14MHz	190mA
PLS157	16 × 45 × 12	20-Pin	16 (4)	Up to 32	6 (0)	6 I/O, 6 R I/O	50ns	14MHz	190mA
PLS159A	16 × 45 × 12	20-Pin	16 (4)	Up to 32	8 (0)	4 I/O, 8 R I/O	35ns	18MHz	190mA
PLS167	22 × 48 × 6	24-Pin	22 (14)	Up to 48	8 (6)	6 R		14MHz	180mA
PLS167A	22 × 48 × 6	24-Pin	22 (14)	Up to 48	8 (6)	6 R		20MHz	180mA
PLS168	22 × 48 × 8	24-Pin	22 (12)	Up to 48	10 (6)	8 R		14MHz	180mA
PLS168A	22 × 48 × 8	24-Pin	22 (12)	Up to 48	10 (6)	8 R		20MHz	180mA
PLS179	20 × 45 × 12	24-Pin	20 (8)	Up to 32	8 (0)	4 I/O, 8 R I/O	35ns	18MHz	210mA
PLC42VA12/I	42 × 105 × 12	24-Pin	42 (10)	Up to 64	10 (0)	10 I/O or R I/O, 2 I/O	35ns	25MHz	135mA
PLC415–16	25 × 68 × 8	28-Pin	25 (17)	Up to 64	8 (8)	8 R		16MHz	100µA/ 80mA
PML									
PLHS501	104 × 116 × 24	52-Pin	32 (24)	Up to 136♦	0	16 C, 8 I/O	22ns		295mA
PML2552–35	205 × 210 × 24	68-Pin	53 (29)	Up to 258♦	36 (20)	8 I/O, 16 R I/O	35ns	50MHz	10mA/ 100mA
PML2552–50	205 × 210 × 24	68-Pin	53 (29)	Up to 258♦	36 (20)	8 I/O, 16 R I/O	50ns	35MHz	10mA/ 100mA
PML2852–35	205 × 210 × 40	84-Pin	53 (29)	Up to 258♦	36 (20)	16 C, 8 I/O, 16 R I/O	35ns	50MHz	10mA/ 100mA
PML2852–50	205 × 210 × 40	84-Pin	53 (29)	Up to 258♦	36 (20)	16 C, 8 I/O, 16 R I/O	50ns	35MHz	10mA/ 100mA

PAL Device = Programmable Array Logic (Fixed OR Array)-Type
PHD = Programmable High-Speed Decoder
PLA = Programmable Logic Array
PLS = Programmable Logic Sequencer
PML = Programmable Macro Logic

OUTPUTS:
C = Combinatorial output
R = Registered output
I/O = Combinatorial I/O
R I/O = Registered I/O

NOTES:
$f_{MAX} = 1/(t_{IS} + t_{CKO})$ worst case
* Includes control product terms

♦ Product terms per NAND gate
PAL is a registered trademark of AMD.
PML is a trademark of Philips Semiconductors.
+ Under development
# 3 Volt devices

All packages refer to DIP configurations except PHD48N22, PML2552 and PML2852, which are offered in PLCC only.

Ordering Information

PLD PRODUCTS

Example: P(L) XX YYYY Q P

P(L) — Indicator for Philips Programmable Logic
(Can be either P, PL, P3, or blank)
(P for PHD and PML and blank for ECL devices)

XX — Process/Architecture Indicator
S = Bipolar Junction Isolated Schottky – Nichrome fuses
C = CMOS – EPROM cells
HS = High Speed Bipolar Oxide Isolated – Vertical Fuse
US = High Speed Bipolar Oxide Isolated – Lateral Fuse
HD = High Speed Decoder
ML = Macro Logic
(Blank for ECL devices)

YYYY — Basic Part Number
(3 to 8 characters)
(e.g., 100, 105, 153, 168, 173, 18P8, 42VA12)

Q — Performance indicator
Z = Zero standby power devices
blank, A, B, D, -35, -7, etc. = propagation delay (ns)
-37, -45, -55, etc. = operating frequency (MHz)

P — Package Code
A = 20-, 28-, 52-, 68-, 84-lead Plastic Leaded Chip Carrier (PLCC)
D = 20-, 24-Pin Plastic SO (300-mil)
F = 20-, 24-, 28-Pin Ceramic Dual In-Line
N = 20-, 24-, 28-Pin Plastic Dual In-Line
FA = 20-, 24-Pin Ceramic Dual In-Line with Quartz Window
KA = CerQuad (window)

Operating temperature range = 0 to +70°C
EXCEPT: PLC18V8ZI = –40 to +85°C
PL22V10I15 = –40 to +85°C

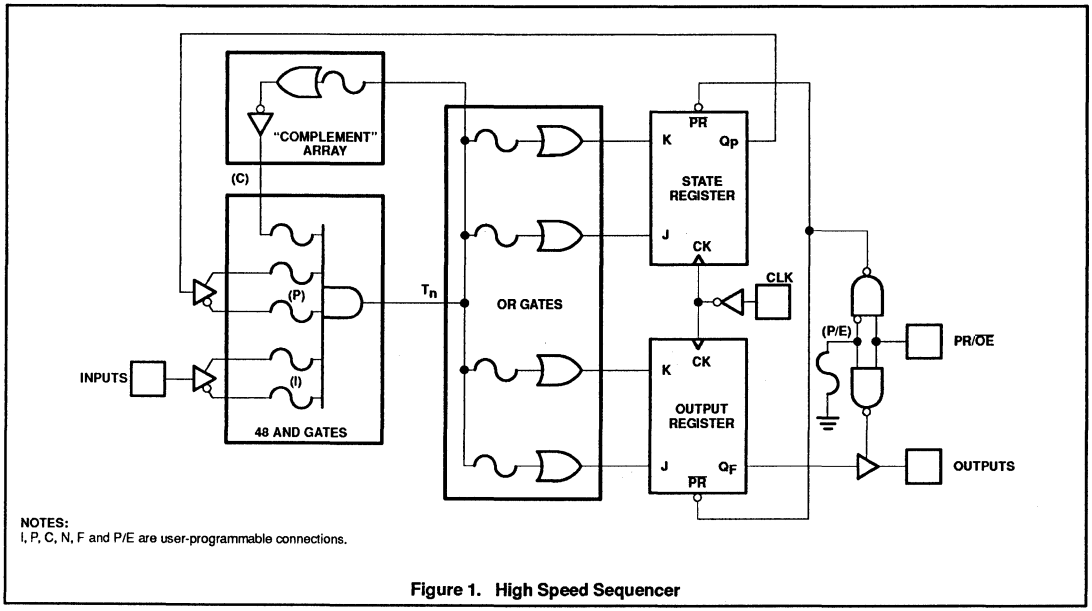
GAL is a registered trademark of Lattice Corp.
PAL is a registered trademark of MMI, Corp., a wholly-owned subsidiary of Advanced Micro Devices (AMD), Inc.

Section 2
Introduction

WHAT IS PROGRAMMABLE LOGIC?

In 1975, Philips Semiconductors developed a new product family by combining its expertise in semi-custom gate array products and fuse-link Programmable Read Only Memories (PROMs). Out of this marriage came Philips Semiconductors Programmable Logic Family. The PLS100 Field-Programmable Logic Array (FPLA) was the first member of this family. The FPLA was an important industry first in two ways. First, the AND/OR/INVERT architecture allowed the custom implementations of Sum of Product logic equations. Second, the three-level fusing allows complete flexibility in the use of this device family. All logic interconnections from input to output are programmable.

Figure 1 shows the architecture of a high performance sequencer combining a PLA architecture with JK flip-flops. The Selection Guide shown on pages 12 and 13 of this data handbook shows the current spectrum of Philips Semiconductors PLDs. Parts for every need are available in nearly every architecture and across at least three technologies. The PLUS and PLHS prefixes describe bipolar parts, the PLC and P3C prefix describes EPLD (CMOS) parts and the PLQ and P3Q prefix refers to the new Philips Semiconductors QUBiC BiCMOS process. Figure 2 shows a shorthand image of the PLUS153 programmable logic array (PLA), which was derived from the original PLS100.



NOTES:
I, P, C, N, F and P/E are user-programmable connections.

Figure 1. High Speed Sequencer

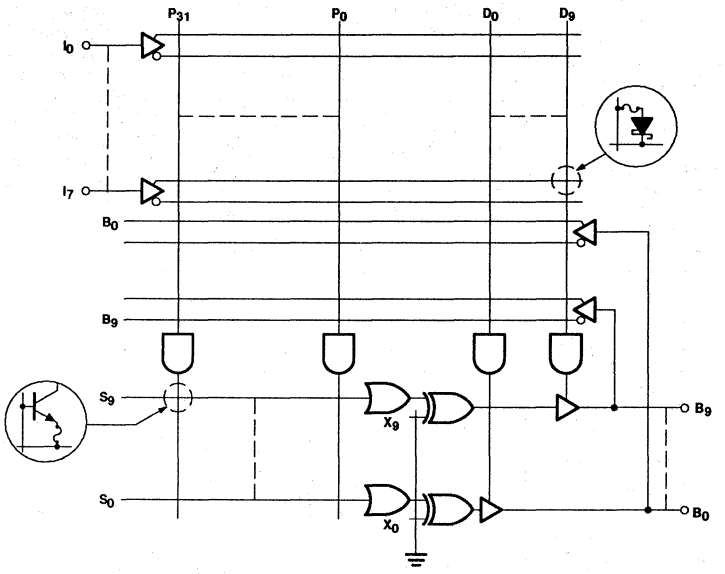


Figure 2. PLUS153 20-Pin Functional Diagram

PLD LOGIC SYNTHESIS

No intermediate step is required to implement Boolean Logic Equations with PLDs. Each term in each equation simply becomes a direct entry into the Logic Program Table. The following example illustrates this straightforward concept:

$X_0 = AB + \overline{C}D + B\overline{D}$

$\overline{X}_1 = \overline{A}B + \overline{C}D + EFG$

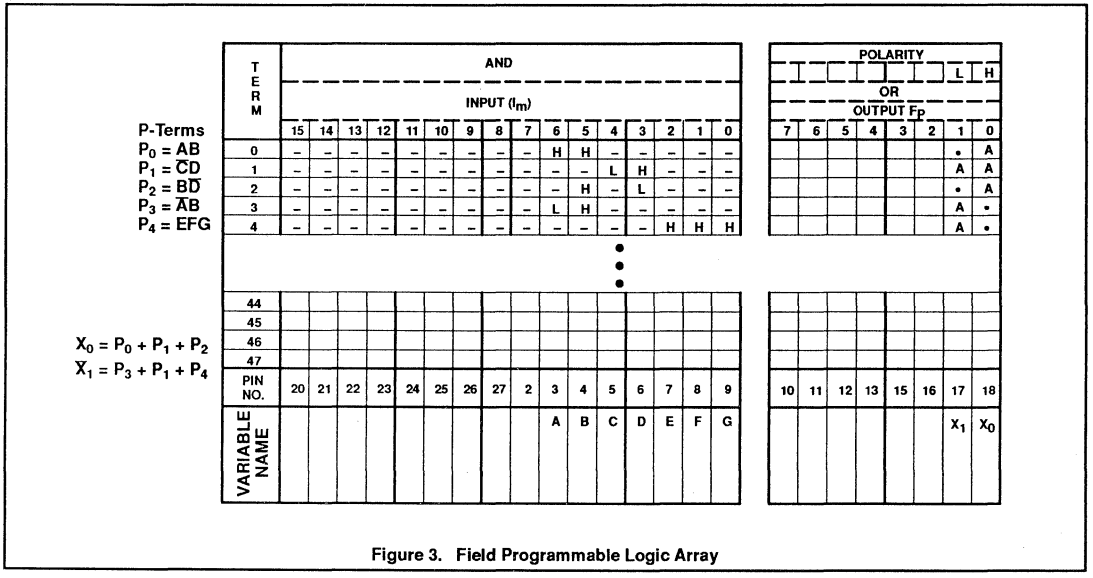


Figure 3. Field Programmable Logic Array

Figure 4. Equivalent Fixed Logic Diagram

In the previous example, the two Boolean Logic equations were broken into Product terms. Each P-term was then programmed into the P-term section of the PLA Program Table. This was accomplished in the following manner:

Step 1

Select which input pins $I_0 - I_{15}$ will correspond to the input variables. In this case A – G are the input variable names. I_6 through I_0 were selected to accept inputs A – G respectively.

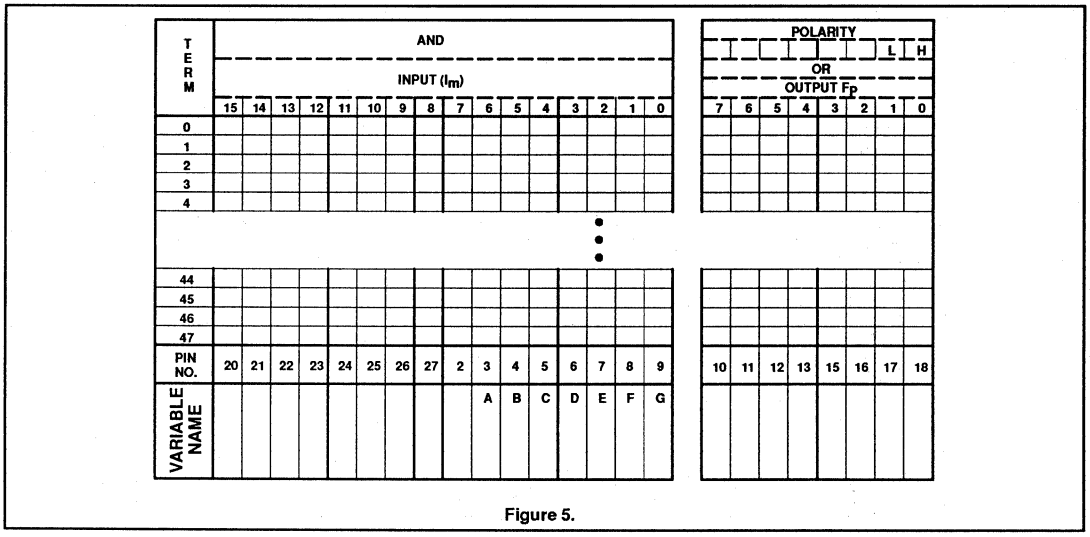


Figure 5.

Step 2

Transfer the Boolean Terms to the PLA Program Table. This is done simply by defining each term and entering it on the Program Table.

e.g., $P_0 = AB$

This P-term translates to the Program Table by selecting $A = I_6 = H$ and $B = I_5 = H$ and entering the information in the appropriate column.

$P_1 = \overline{C}D$

This term is defined by selecting $C = I_4 = L$ and $D = I_3 = H$, and entering the data into the Program Table. Continue this operation until all P-terms are entered into the Program Table.

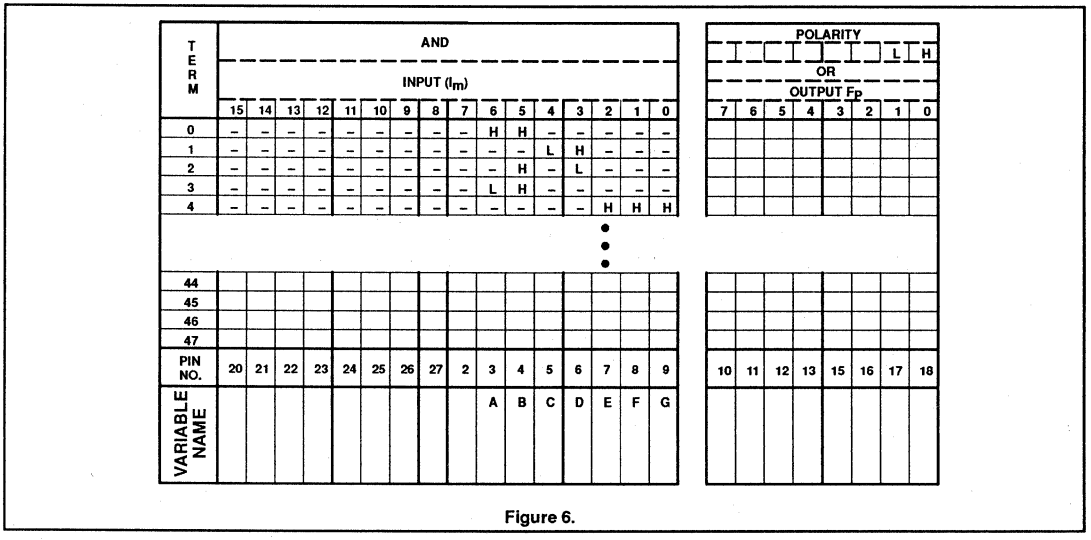
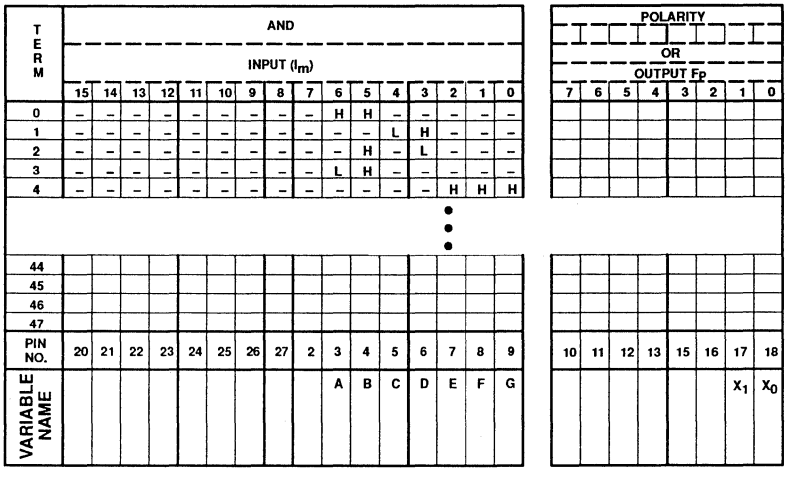


Figure 6.

Step 3

Select which output pins correspond to each output function. In this case F_0 = Pin 18 = X_0, and F_1 = Pin 17 = X_1.

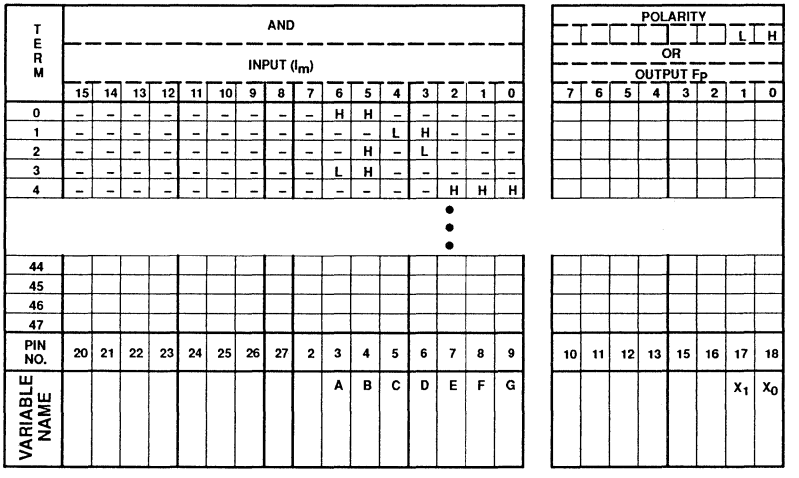


TERM	AND: INPUT (I_m) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POLARITY / OR: OUTPUT Fp 7	6	5	4	3	2	1	0
0	–	–	–	–	–	–	–	–	–	H	H	–	–	–	–	–								
1	–	–	–	–	–	–	–	–	–	–	–	L	H	–	–	–								
2	–	–	–	–	–	–	–	–	–	–	H	–	L	–	–	–								
3	–	–	–	–	–	–	–	–	–	L	H	–	–	–	–	–								
4	–	–	–	–	–	–	–	–	–	–	–	–	–	H	H	H								
• • •																								
44																								
45																								
46																								
47																								
PIN NO.	20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9	10	11	12	13	15	16	17	18
VARIABLE NAME										A	B	C	D	E	F	G							X_1	X_0

Figure 7.

Step 4

Select the Output Active Level desired for each Output Function. For X_0 the active level is high for a positive logic expression of this equation. Therefore, it is only necessary to place an (H) in the Active Level box above Output Function 0, (F_0). Conversely, X_1 can be expressed as $\overline{X}_1$ by placing an (L) in the Active Level box above Output Function 1, (F_1).



TERM	AND: INPUT (I_m) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	POLARITY L H / OR: OUTPUT Fp 7	6	5	4	3	2	1	0
0	–	–	–	–	–	–	–	–	–	H	H	–	–	–	–	–								
1	–	–	–	–	–	–	–	–	–	–	–	L	H	–	–	–								
2	–	–	–	–	–	–	–	–	–	–	H	–	L	–	–	–								
3	–	–	–	–	–	–	–	–	–	L	H	–	–	–	–	–								
4	–	–	–	–	–	–	–	–	–	–	–	–	–	H	H	H								
• • •																								
44																								
45																								
46																								
47																								
PIN NO.	20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9	10	11	12	13	15	16	17	18
VARIABLE NAME										A	B	C	D	E	F	G							X_1	X_0

Figure 8.

Step 5

Select the P-Terms you wish to make active for each Output Function. In this case $X_0 = P_0 + P_1 + P_2$, so an A has been placed in the intersection box for P_0 and X_0, P_1 and X_0 and P_2 and X_0.

Terms which are not active for a given output are made inactive by placing a (•) in the box under that P-term. Leave all unused P-terms unprogrammed.

Continue this operation until all outputs have been defined in the Program Table.

Step 6

Enter the data into a Philips Semiconductors approved programmer. The input format is identical to the Philips Semiconductors Program Table. You specify the P-terms, Output Active Level, and which P-terms are active for each output exactly the way it appears on the Program Table.

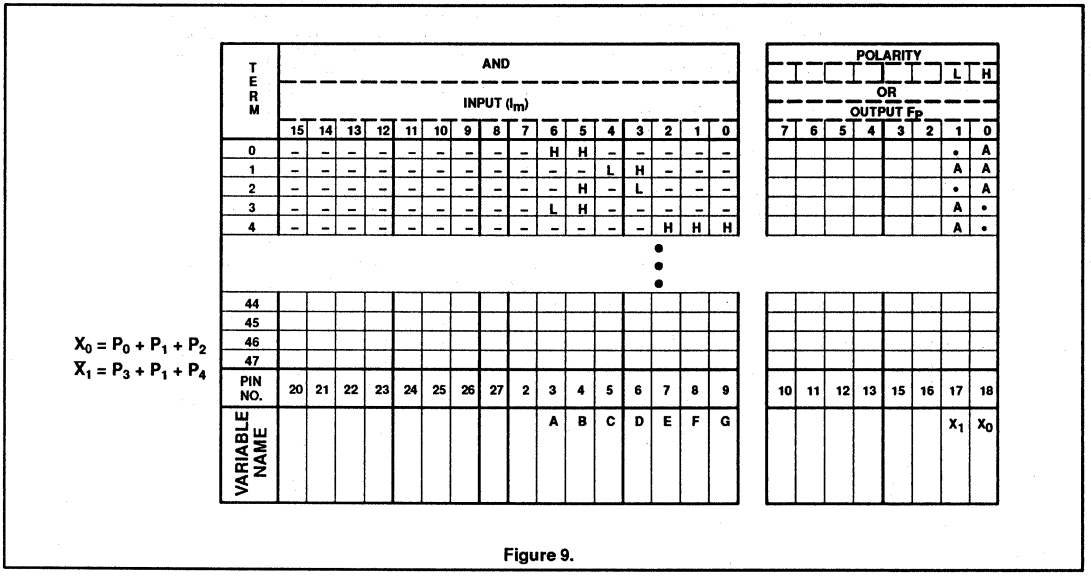


Figure 9.

PLD LOGIC SYNTHESIS
(Continued)

When fewer inputs and outputs are required in a logic design and low cost is most important, the Philips Semiconductors 20-pin PLD should be considered first choice. The PLUS153 is a PLA with 8 inputs, 10 I/O pins, and 42 product terms. The user can configure the device by defining the direction of the I/O pins. This is easily accomplished by using the direction control terms $D_0 - D_9$ to establish the direction of pins $B_0 - B_9$. The D-terms control the 3-State buffers found on the outputs of the Ex-OR gates. Figures 10 and 11 show how the D-term configures each B_X pin.

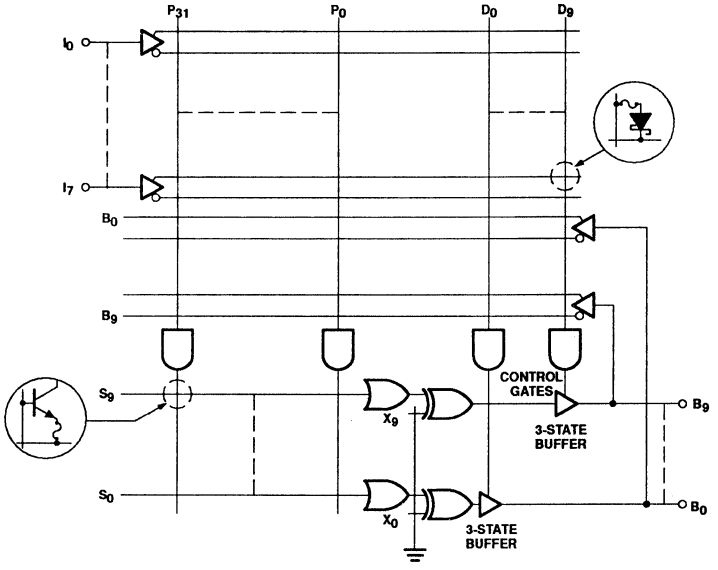
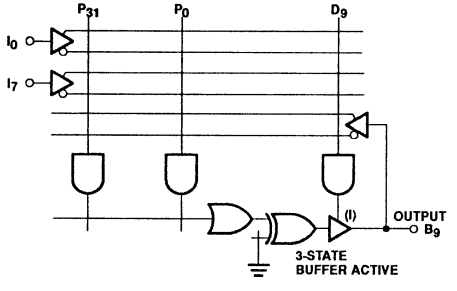
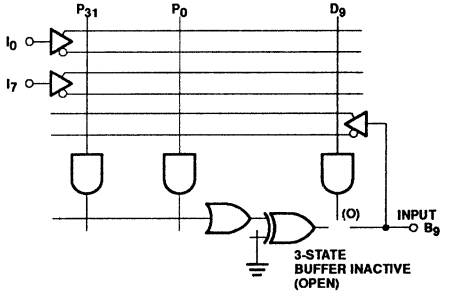


Figure 10. PLUS153 Functional Diagram



a. D_9 Active Makes B_9 Appear as an Output with Feedback



b. D_9 Inactive Makes B_9 Appear as an Input

Figure 11.

To control each D-term, it is necessary to understand that each control gate is a 36-input AND gate. To make the 3-State buffer active (B_x pin an output), the output of the control gate must be at logic HIGH (1). This can be accomplished in one of two ways. A HIGH can be forced on all control gate input nodes, or fuses can be programmed. When a fuse is programmed, that control gate input node is internally pulled up to HIGH (1). See Figure 12 and Figure 13.

Programming the fuse permanently places a HIGH (1) on the input to the control gate. The input pin no longer has any effect on that state.

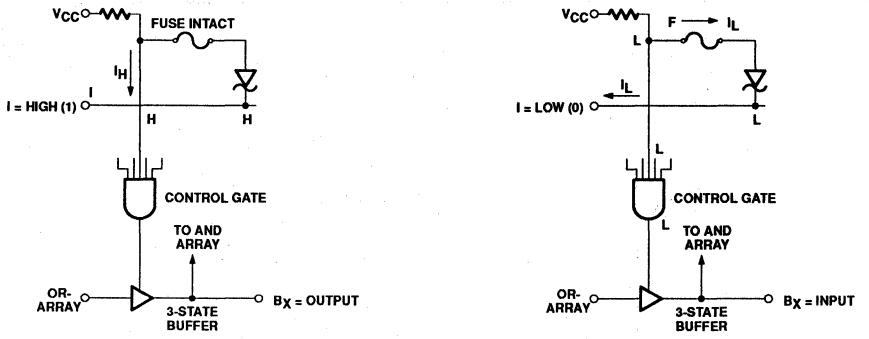


Figure 12. Input Effect on Control Gates (Fuse Intact)

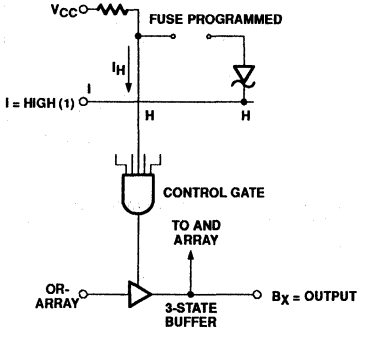


Figure 13. Effect on Control Gate if Fuse is Programmed

DEDICATING B_X PIN DIRECTION

Since each input to the D-terms is true and complement buffered (see Figure 11), when the device is shipped with all fuses intact, all control gates have half of the 36 input lines at logic low (0). The result of this is all Control Gate outputs are low (0) and the 3-State buffers are inactive. This results in all B_X pins being in the input condition. the resultant device is, therefore, an 18-input, 0-output FPLA. While useful as a bit bucket or Write-Only-Memory (WOM), most applications require at least one output. Clearly, the first task is to determine which of the B_X pins are to be outputs. The next step is to condition the control gate to make the 3-State buffer for those gates active. To dedicate B_0 and B_1 as outputs, it is necessary to program all fuses to the inputs to Control Gates D_0 and D_1. This internally pulls all inputs to those gates to HIGH (1) permanently. since all inputs to the Control Gates are HIGH (1), the output is HIGH (1) and the 3-State buffers for B_0 and B_1 are active. This permanently enables B_0 and B_1 as outputs. Note that even though B_0 and B_1 are outputs, the output data is available to the AND array via the internal feedback (see Figure 11a).

To program this data, the PLUS153 Program Table is used as shown in Figure 14.

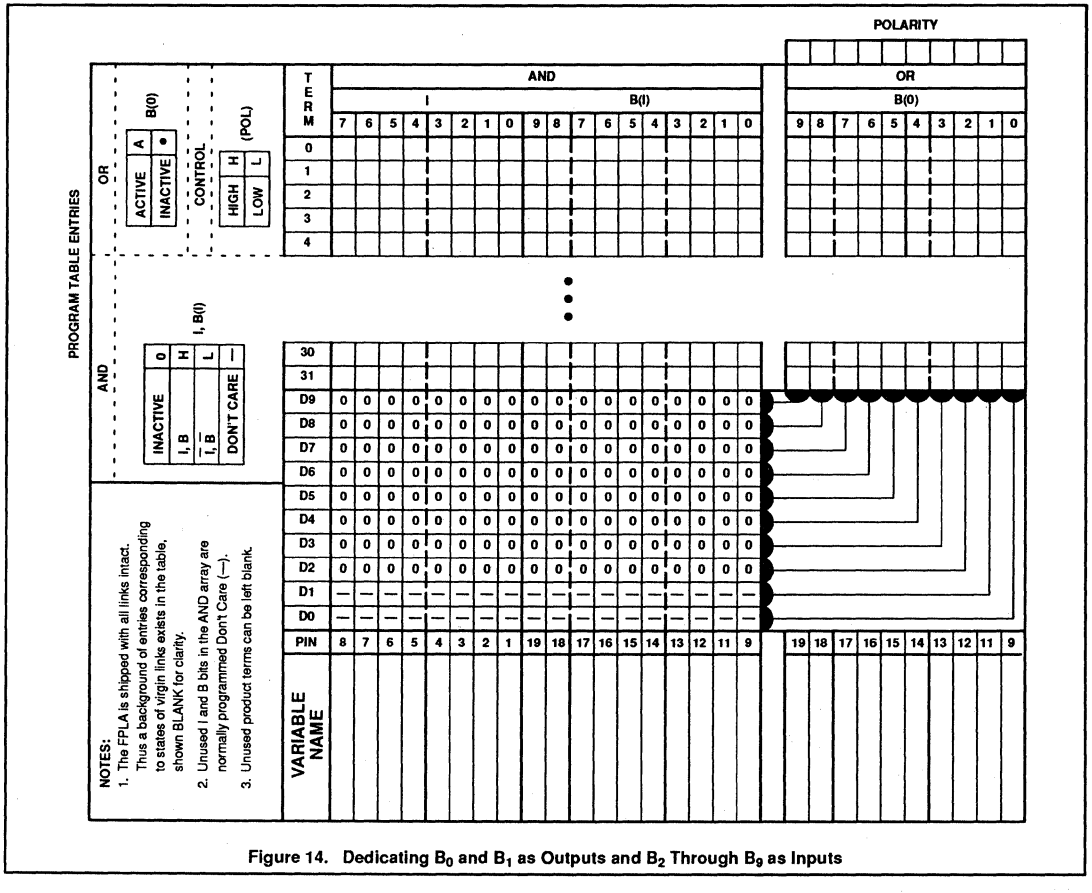


Figure 14. Dedicating B_0 and B_1 as Outputs and B_2 Through B_9 as Inputs

By placing a (—) Don't Care in each input box you are specifying that the True and Complement fuses are programmed on each Control Gate, thus permanently dedicating the B_0 and B_1 pins as outputs. By placing a (0) in all input boxes for B_2 - B_9, you are specifying that both True and Complement fuses are intact. This causes a low (0) to be forced on half of the Control Gate inputs, guaranteeing the output of the Control Gate will be low (0). When the Control Gate outputs are low (0), the 3-State buffer is inactive and the B_2 - B_9 pins are enabled as inputs. All B_X pin directions can be controlled in this manner.

ACTIVE DIRECTION CONTROL

Sometimes it is necessary to be able to actively change the direction of the B_X pins without permanently dedicating them. Some applications which require this include 3-State bus enable, multi-function decoding, etc. This can easily be done by programming the Control Gate to respond to one or more input pins. It is only necessary to select which I_X and B_X pins will control the pin directions and the active level HIGH (H) or LOW (L) that will be used. The PLUS153 Program Table in Figure 15 shows the method of controlling $B_0 - B_9$ with I_7. When I_7 is LOW (L), pins $B_0 - B_9$ are outputs; when I_7 is HIGH (H), pins $B_0 - B_9$ are inputs. Note that by programming all other I_X and B_X pins as DON'T CARE (—), they are permanently disconnected from control of B_X pin direction.

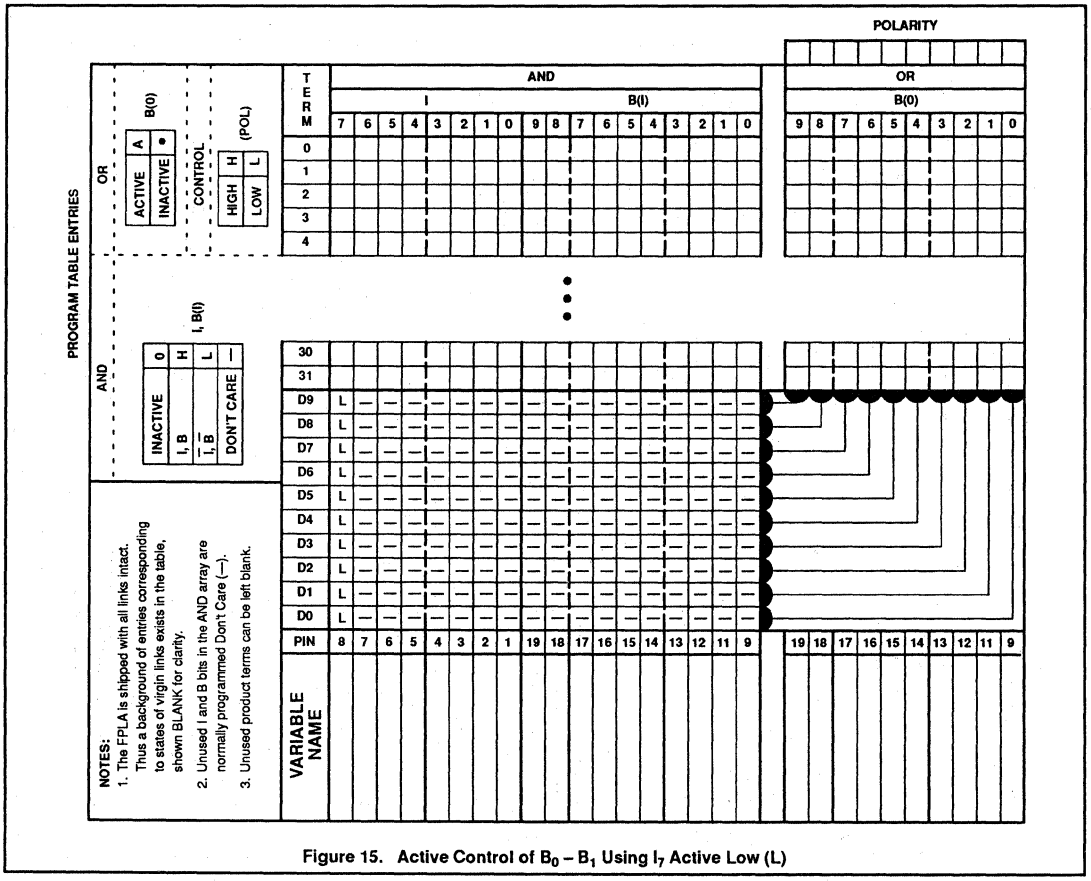


Figure 15. Active Control of $B_0 - B_1$ Using I_7 Active Low (L)

The previous 28-pin logic synthesis example could be done on the PLUS153 as follows:

$X_0 = AB + \overline{C}D + B\overline{D}$

$\overline{X}_1 = \overline{A}B + \overline{C}D + EFG$

Note that B_0 was used as a CHANGE input. When B_0 is HIGH (H) the outputs appear on B_8 and B_9. When B_0 is LOW (L), the outputs appear on B_6 and B_7. B_1 through B_5 are not used and therefore left unprogrammed.

Philips Semiconductors offers two packages for user-friendly design assistance. The first package, AMAZE, has evolved over 10 years to support Philips Semiconductors programmable products with logic equation, state equation, and schematic entry. AMAZE can compile designs quite well for Philips Semiconductors lower density parts. However, to satisfy the needs of Programmable Macro Logic users, Philips Semiconductors developed an additional software package called SNAP. SNAP expands upon the capabilities of AMAZE in its approach to design implementation, more closely resembling a gate array methodology. Both of these products are described in more depth at a later point in this handbook.

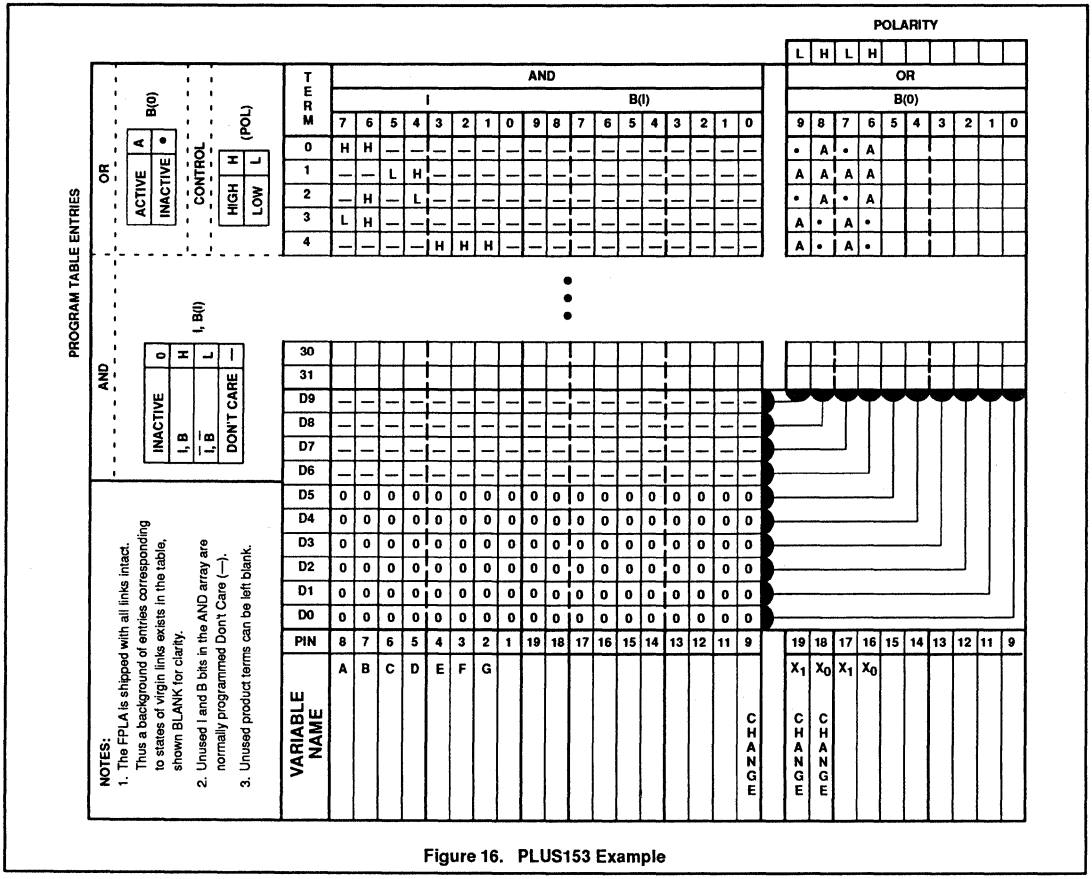


Figure 16. PLUS153 Example

SEQUENTIAL LOGIC CONSIDERATIONS

The PLUS405, PLUS105 and PLC42VA12 represent significant increases in complexity when compared to the combinatorial logic devices previously discussed. By combining the AND/OR combinatorial logic with clock output flip-flops and appropriate feedback, Philips Semiconductors has created the first family of totally flexible sequential logic machines.

The PLUS405 (Programmable Logic Sequencer) is an example of a high-order machine whose applications are many. Application areas for this device include VRAM, DRAM, Bus and LAN control. The PLUS405 is fully capable of performing fast sequential operations in relatively high-speed processor systems. By placing repetitive sequential operations on the PLUS405, processor overhead is reduced.

The following pages summarize the PLUS405 architecture and features.

Sequencer Architecture

The PLUS405 Logic Sequencer is a programmable state machine, in which the output is a function of the present state and the present input.

With the PLUS405, a user can program any logic sequence expressed as a series of jumps between stable states, triggered by a valid input condition (I) at clock time (t). All stable states are stored in the State Register. The logic output of the machine is also programmable, and is stored in the Output Register. The PLUS105 is a subset of the PLUS405.

Clocked Sequence

A synchronous logic sequence can be represented as a group of circles interconnected with arrows. The circles represent stable states, labeled with an arbitrary numerical code (binary, hex, etc.) corresponding to discrete states of a suitable register. The arrows represent state transitions, labeled with symbols denoting the jump condition and the required **change** in output. The number of states in the sequence depends on the length and complexity of the desired algorithm.

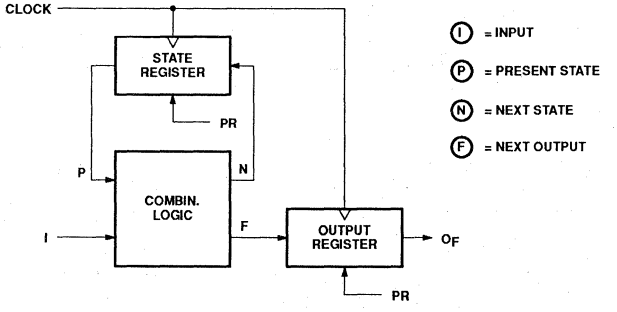
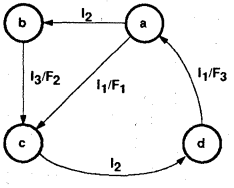
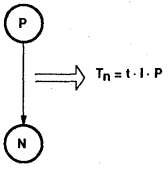


Figure 17. Basic Architecture of PLS105 FPLS. I, P, N, and F are Multi-line Paths Denoting Groups of Binary Variables Programmed by the User.



I_{1-3} are jump conditions which must be satisfied before any transitions take place.
F_r are changes in output triggered by I_m, and stored in the output register.
State transitions a → b and c → d involve no output change.

Figure 18. Typical State Diagram.



The arrow connecting the two states gives rise to a transition term T_n. I is the jump condition.

Figure 19. Typical State Transition Between Any Two States of Figure 18.

State Jumps

The state from which a jump originates is referred to as the Present state (P), and the state to which a jump terminates is defined as the Next state (N). A state jump always causes a change in state, but may or may not cause a change in machine output (F).

State jumps can occur only via "transition terms" T_n. These are logical AND functions of the clock (t), the Present state (P), and a valid input (I). Since the clock is actually applied to the State Register, $T_n = I \bullet P$. When T_n is "true", a control signal is generated and used at clock time (t) to force the contents of the State Register from (P) to (N), and to change the contents of the Output Register (if necessary). The simple state jump in Figure 20, involving 2 inputs, 1 state bit, and 1 output bit, illustrates the equivalence of discrete and programmable logic implementations.

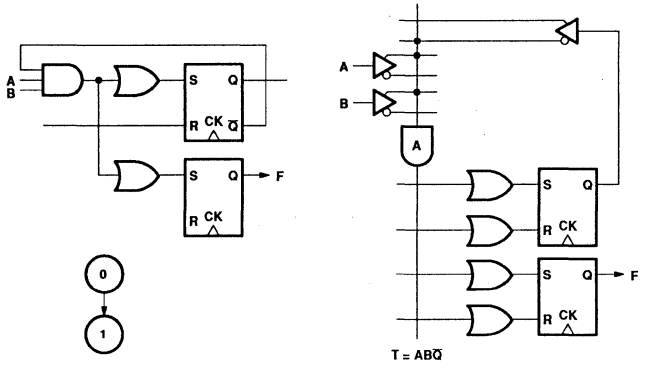


Figure 20. Typical State Jump From State (0) to State (1), if Inputs A = B = "1". The Jump Also Forces F = "1", as Required.

Sequencer Logic Structure

The Sequencer consists of programmable AND and OR gate arrays which control the Set and Reset inputs of a State Register, as well as monitor its output via an internal feedback path. The arrays also control an independent Output Register, added to store output commands generated during state transitions, and to hold the output constant during state sequences involving no output changes. If desired, any number of bits of the Output Register can be used to extend the width of the State Register, via external feedback.

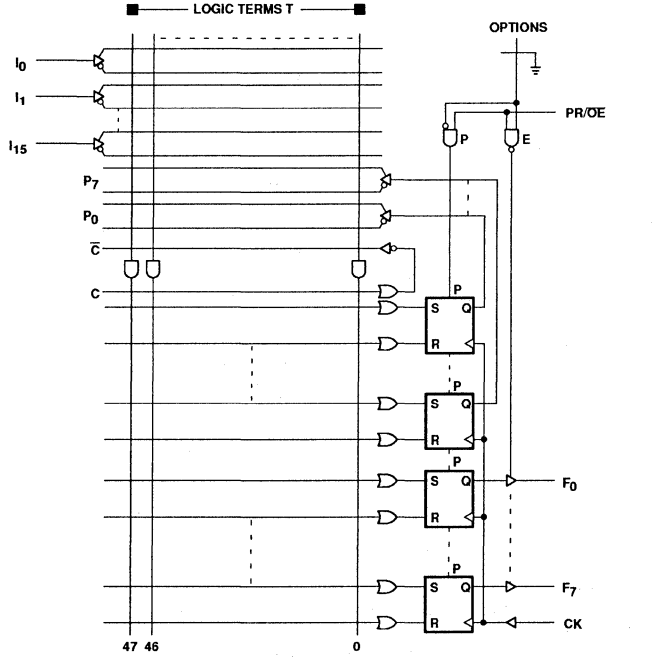


Figure 21. Simplified Logic Diagram of PLUS105

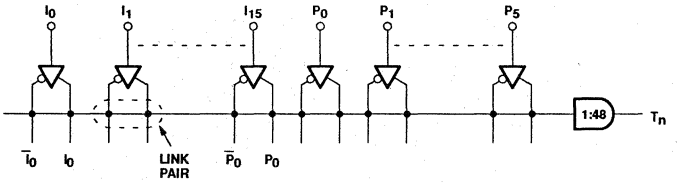


Figure 22. Typical AND Gate Coupled to (I) and (P) Inputs. If at Least One Link Pair Remains Intact, T_n is Unconditionally Forced Low.

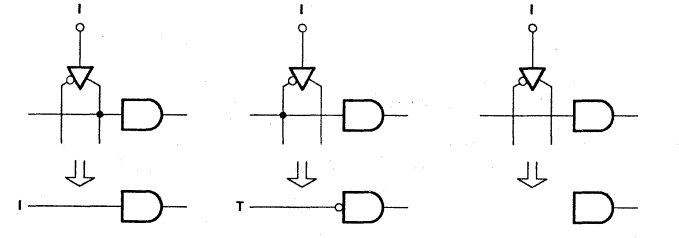


Figure 23. Choice of Input Polarity Coupling to a Typical AND Gate. With Both Links Open, (I) is Logically Don't Care.

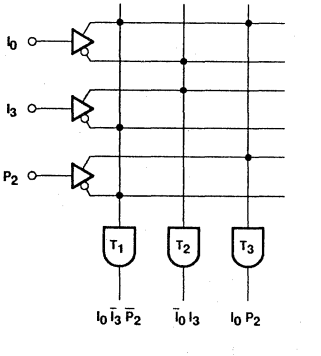


Figure 24. Typical Transition Terms Involving Arbitrary Inputs and State Variables. All Remaining Gate Inputs Are Programmed Don't Care. Note That T_2 Output is State Independent.

Input Buffers

16 external inputs (I_m) and 6 internal inputs (P_s), fed back from the State Register, are combined in the AND array through two sets of True/Complement (T/C) buffers. There are a total of 22 T/C buffers, all connected to multi-input AND gates via fusible links which are initially intact.

Selective fusing of these links allows coupling either True, Complement, or Don't Care values of (I_m) and (P_s).

"AND" Array

State jumps and output changes are triggered at clock time by valid transition terms T_n. These are logical AND functions of the present state (P) and the present input (I).

The PLUS105 AND Array contains a total of 48 AND gates. Each gate has 45 inputs – 44 connected to 22 T/C input buffers, and 1 dedicated to the Complement Array. The outputs of all AND gates are propagated through the OR Array, and used at clock time (t) to force the contents of the State Register from (P) to (N). they are also used to control the Output Register, so that the FPLS 8-bit output F_r is a function of the inputs and the present state. The PLUS405 contains 64 AND gates in its' AND array.

"OR" Array

In general, a clocked sequence will consist of several stable states and transitions, as determined by the complexity of the desired algorithm. All state and output changes in the state diagram imply changes in the contents of State and Output Registers.

Thus, each flip-flop in both registers may need to be conditionally set or reset several times with T_n commands. This is accomplished by selectively ORing through a programmable OR Array all AND gate outputs T_n necessary to activate the proper flip-flop control inputs.

The PLUS105 OR Array consists of 14 pairs of OR gates, controlling the S/R inputs of 14 State and Output Register stages, and a single NOR gate for the Complement Array. All gates have 48 inputs for connecting to all 48 AND gates. The PLUS405 uses 64 input gates.

The PLUS405 contains 16 pairs of OR gates controlling state transitions and output stages and two additional NOR gates for dual complement arrays.

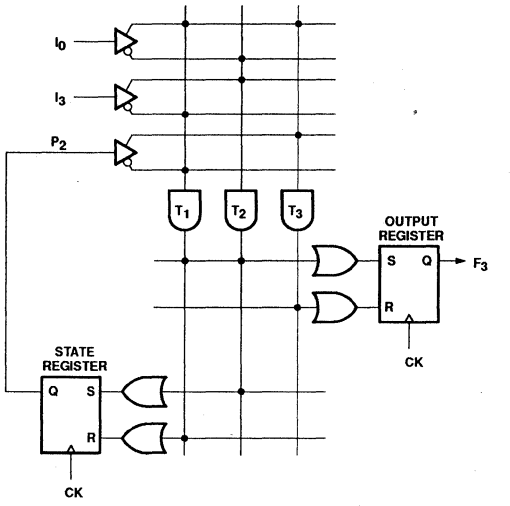


Figure 25. Typical OR Array Gating of Transition Terms $T_{1,2,3}$ Controlling Arbitrary State and Output Register Stages.

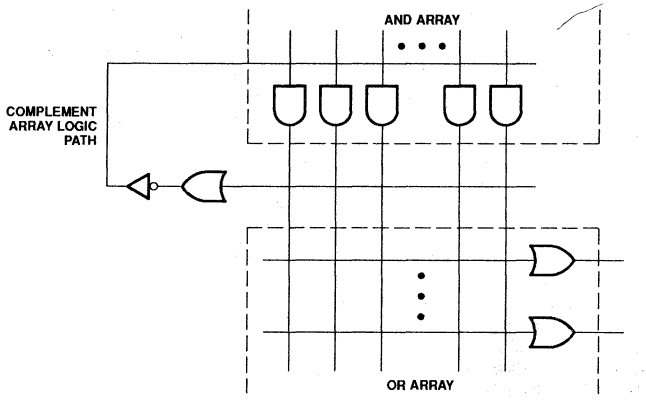
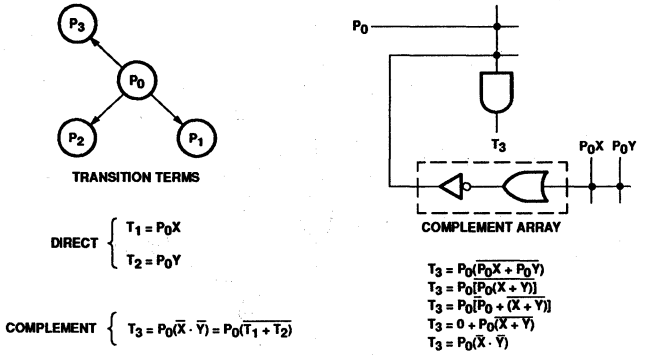


Figure 26. The COMPLEMENT Array is Logically Constructed from a Multiple Input Programmable NOR Gate. All AND Terms Coupled to the OR Gate are Complemented at the Inverter Output, and Can be Fed Back as Inputs to the AND Array.



a. Typical State Sequence b. Complement Jump

Figure 27.

a. **X and Y Specify the Conditional Logic for Direct Jump Transition Terms T_1 and T_2. The Complement Jump Term T_3 is True Only When Both T_1 and T_2 are False.**

b. **Note that the Complementary Logic Expression for T_3, $\overline{T_1 + T_2}$, Corresponds Exactly to the Logic Structure of the Complement Array.**

Complement Array

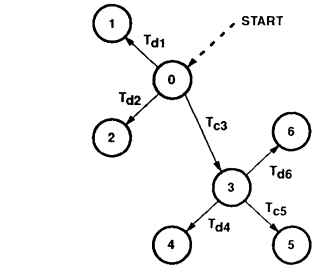
The Complement Array provides an asynchronous feedback path from the OR Array back to the AND Array.

This structure enables the sequencer to perform both direct and complement sequential state jumps with a minimum of transition (AND) terms.

Typically direct jumps, such as T_1 and T_2 in Figure 27 require only a single AND gate each.

But a complement jump such as T_3 generally requires many AND gates if implemented as a direct jump. However, by using the Complement Array, the logic requirements for this type of jump can be handled with just one more gate from the AND Array. Because it can be split into separate machines (2 clocks), the PLUS405 incorporates two Complement Arrays.

As indicated in Figure 28, the single Complement Array gate may be used for many states of the state diagram. This happens because all transition terms linked to the OR gate include the present state as a part of their conditional logic. In any particular state, only those transition terms which are a function of that state are enabled; all other terms coupled to different states are disabled and do not influence the output of the Complement Array. As a general rule of thumb, the Complement Array can be used as many times as there are states.



a. State Diagram

$T_{d1} = I_0 \overline{I_1} P_0$

$T_{d2} = \overline{I_2} P_0$

$T_{c3} = (\overline{T_{d1} + T_{d2}}) P_0 = (\overline{I_0 \overline{I_1} + I_2}) P_0$

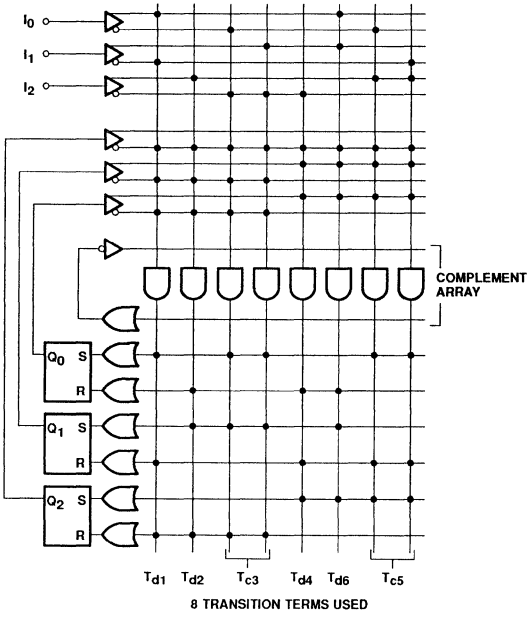
$T_{d4} = I_2 P_3$

$T_{d6} = I_0 \overline{I_1} P_0$

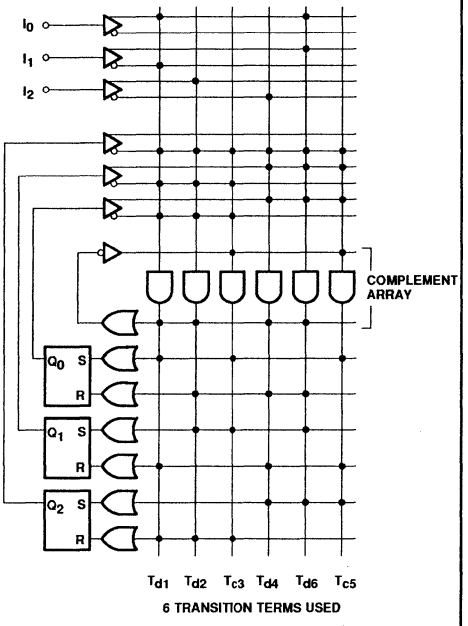
$T_{c5} = (\overline{T_{d4} + T_{d6}}) P_3 = (\overline{I_0 \overline{I_1} + \overline{I_2}}) P_3$

T_{cn} = COMPLEMENT STATE TRANSITION TERM
T_{dn} = DIRECT STATE TRANSITION TERM
P_s = PRESENT STATE

b. Logic Definition



c. State Logic without Using the Complement Array



d. State Logic Using the Complement Array

Figure 28. Logic Reduction with the Complement Array. The Logic State Diagram in (a) Includes Complement Jumps T_{C3} and T_{C5} Defined in (b). When Using the Complement Array, a Savings of 2 Transition Terms Results, as Shown in (c) and (d).

Additional features are available depending on a specific part. In particular, the PLC42VA12 has everything mentioned here, and more. More details on PLAs, PAL devices and Sequencers can be found in the application section later in the manual.

Programmable Macro Logic, Philips Semiconductors very high density logic is fully described in detail in its own section.

Section 3
PAL Devices

CONTENTS

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8-5

DESCRIPTION

The PHD16N8-5 is an ultra fast Programmable High-speed Decoder featuring a 5ns maximum propagation delay. The architecture has been optimized using Philips Semiconductors state-of-the-art bipolar oxide isolation process coupled with titanium-tungsten fuses to achieve superior speed in any design.

The PHD16N8-5 is a single level logic element comprised of 10 fixed inputs, 8 AND gates, and 8 outputs of which 6 are bidirectional. This gives the device the ability to have as many as 16 inputs. Individual 3-State control of all outputs is also provided.

The device is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment. Proprietary designs can be protected by programming the security fuse.

The SLICE software package from Philips Semiconductors supports easy design entry for the PHD16N8-5 as well as other PLD devices.

Order codes are listed below.

FEATURES

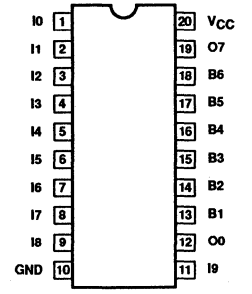
- Ideal for high speed system decoding
- Super high speed at 5ns t_{PD}
- 10 dedicated inputs
- 8 outputs
  - 6 bidirectional I/O
  - 2 dedicated outputs
- Security fuse to prevent duplication of proprietary designs.
- Individual 3-State control of all outputs
- Field-programmable on industry standard programmers
- Available in 20-pin Plastic Dual In-Line and 20-Pin PLCC

APPLICATIONS

- High speed memory decoders
- High speed code detectors
- Random logic
- Peripheral selectors
- Machine state decoders
- Footprint compatible to 16L8
- Fuse/Footprint compatible to TIBPAD

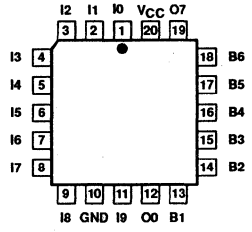
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

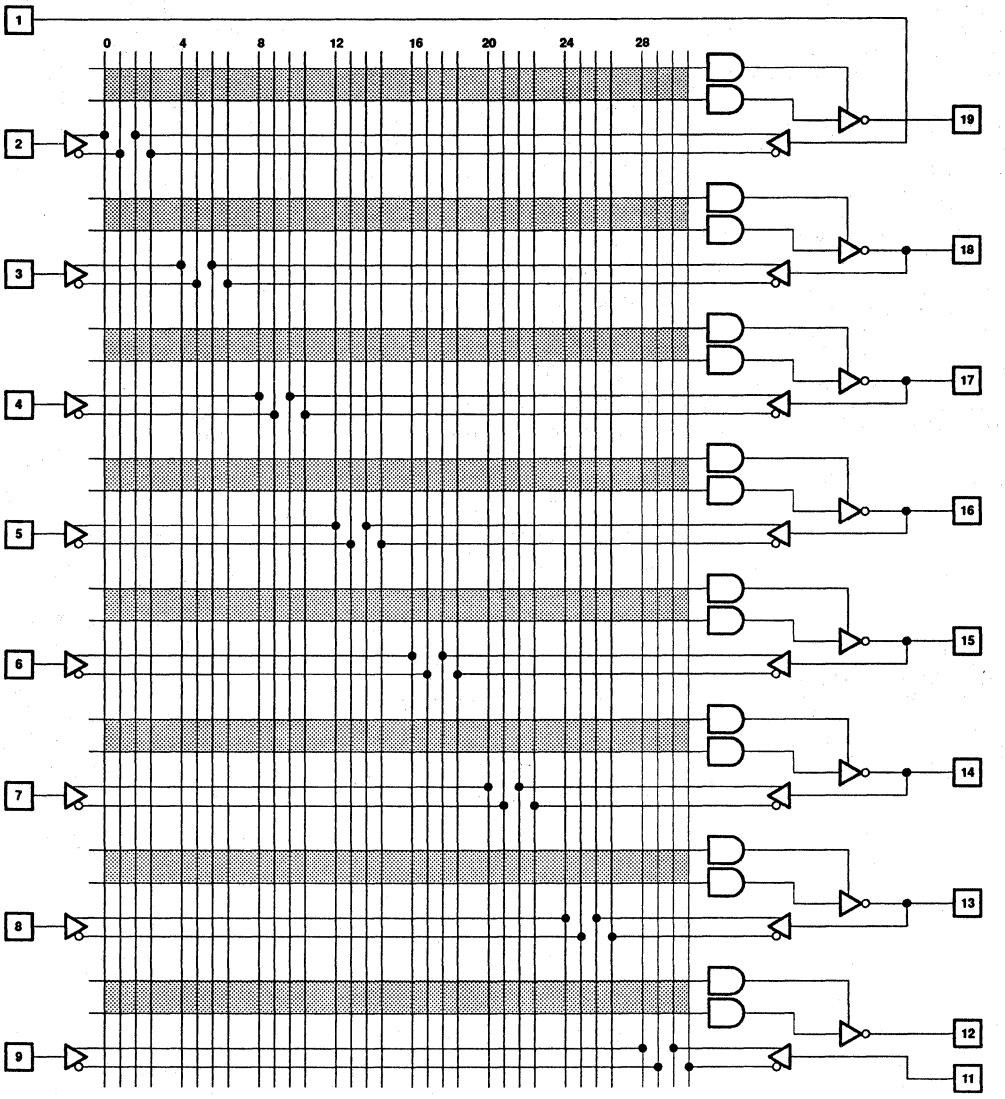
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual In Line Package; (300 mil-wide)	PHD16N8-5N	0173D
20-Pin Plastic Leaded Chip Carrier; (350 mil square)	PHD16N8-5A	0400E

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8-5

LOGIC DIAGRAM



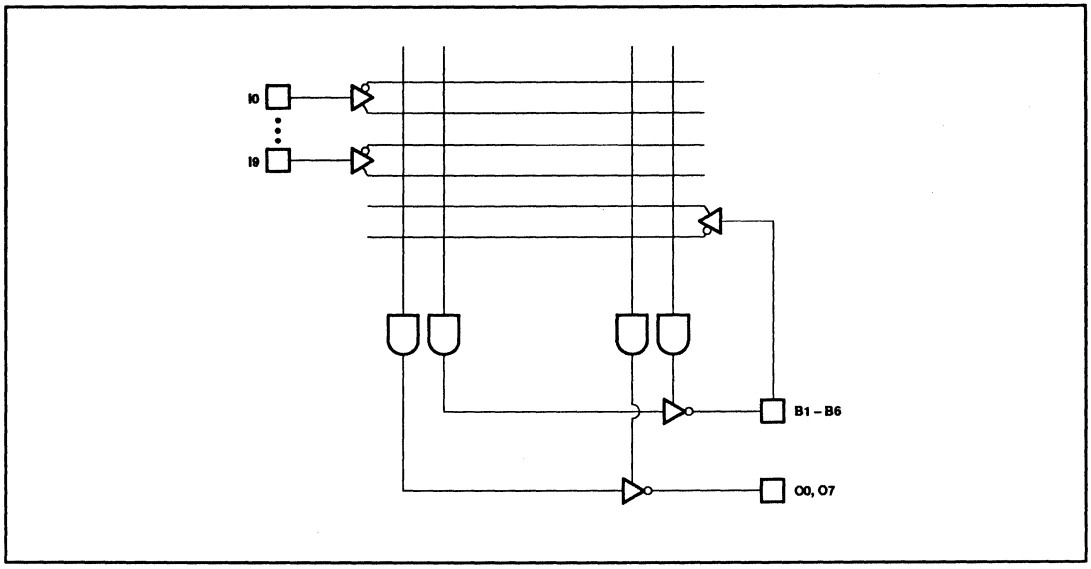
NOTES:

1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8-5

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		Min	Max	
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–0.5	+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS		UNIT
		Min	Max	
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = +24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = +0.40V		–20	–250	µA
I_{IH}	High	V_{IN} = +2.7V			25	µA
I_I	High	V_{IN} = V_{CC} = V_{CC} MAX			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage[3]	V_{OUT} = +2.7V			100	µA
I_{OZL}	Output leakage[3]	V_{OUT} = +0.40V			–100	µA
I_{OS}	Short circuit[4]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		115	180	mA
Capacitance[5]						
		V_{CC} = +5V				
C_{IN}	Input	V_{IN} = 2.0V @ f = 1MHz		8		pF
C_{OUT}	I/O (B)	V_{OUT} = 2.0V @ f = 1MHz		8		pF

NOTES:

1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second.
5. These parameters are not 100% tested, but are periodically sampled.

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8-5

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS MIN	LIMITS MAX	UNIT
t_{PD}[1]	Propagation delay	(I, B) ±	Output ±	C_L = 50pF		5	ns
t_{OE}[2]	Output Enable	(I, B) ±	Output enable	C_L = 50pF		10	ns
t_{OD}[2]	Output Disable	(I, B) ±	Input disable	C_L = 5pF		10	ns

NOTES:
1. t_{PD} is tested with switch S_1 closed and C_L = 50pF.
2. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

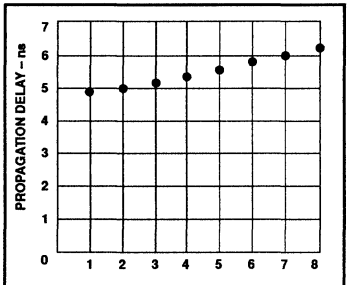
VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All outputs are disabled.
2. All p-terms are disabled in the AND array.

TIMING DEFINITIONS

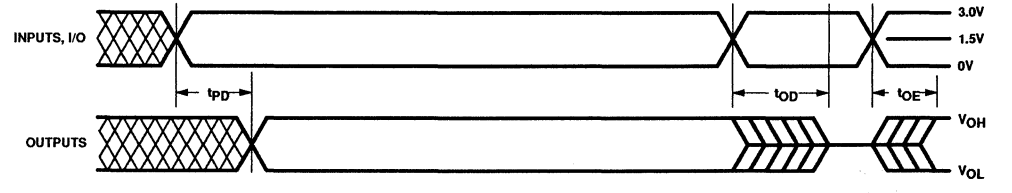
SYMBOL	PARAMETER
t_{PD}	Input to output propagation delay.
t_{OD}	Input to Output Disable (3-State) delay (Output Disable).
t_{OE}	Input to Output Enable delay (Output Enable).



TEST CONDITIONS: T_{amb} = 75°C; V_{CC} = 4.75V; C_L = 50pF; R_1 = 200Ω; R_2 = 390Ω

Worst-Case Propagation Delay vs. Number of Outputs Switching

TIMING DIAGRAM



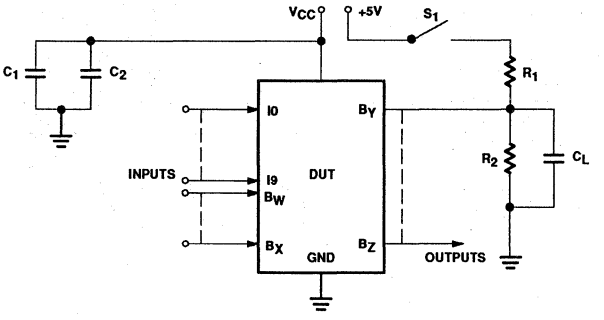
WAVEFORM	INPUTS	OUTPUTS
———	MUST BE STEADY	WILL BE STEADY
———		

WAVEFORM	INPUTS	OUTPUTS
	DON'T CARE; ANY CHANGE PERMITTED	CHANGING; STATE UNKNOWN
	DOES NOT APPLY	CENTER LINE IS HIGH IMPEDANCE "OFF" STATE

Programmable high-speed decoder logic (16 × 16 × 8)

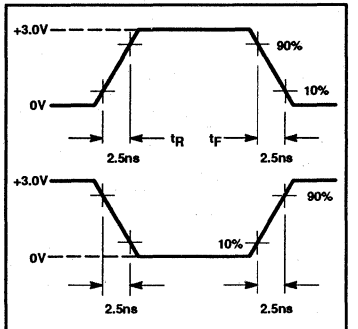
PHD16N8-5

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

The PHD16N8-5 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PHD16N8–5 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

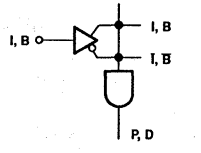
PHD16N8-5 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, each logic variable (I, B, P and D) from the logic equations is assigned a symbol. TRUE (High), COMPLEMENT (Low), DON'T CARE and INACTIVE symbols are defined below.

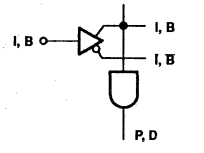
PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information.

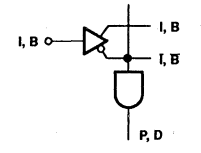
"AND" ARRAY – (I, B)



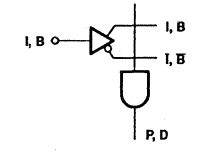
STATE	CODE
INACTIVE[1]	O



STATE	CODE
TRUE	H



STATE	CODE
COMPLEMENT	L



STATE	CODE
DON'T CARE	–

NOTE:
1. This is the initial state.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable high-speed decoder logic (16 × 16 × 8)

PHD16N8-5

PROGRAM TABLE

OR (FIXED): D = DIRECTION, A = ACTIVE OUTPUT, NOT USED

AND: I, B(I): 0 = INACTIVE; H = I, B; L = $\overline{I}$, $\overline{B}$; — = DON'T CARE

TERM	AND: INPUT (I) 9	8	7	6	5	4	3	2	1	0	AND: INPUTS (B) 6	5	4	3	2	1	OR (FIXED): OUTPUTS (B, O) 7	6	5	4	3	2	1	0
0																	D							
1																	A							
2																		D						
3																		A						
4																			D					
5																			A					
6																				D				
7																				A				
8																					D			
9																					A			
10																						D		
11																						A		
12																							D	
13																							A	
14																								D
15																								A
PIN	11	9	8	7	6	5	4	3	2	1	18	17	16	15	14	13	19	18	17	16	15	14	13	12
VARIABLE NAME																								

CUSTOMER NAME _____

PURCHASE ORDER # _____

PHILIPS DEVICE # _____ CF(XXXX)

CUSTOMER SYMBOLIZED PART # _____

TOTAL NUMBER OF PARTS _____

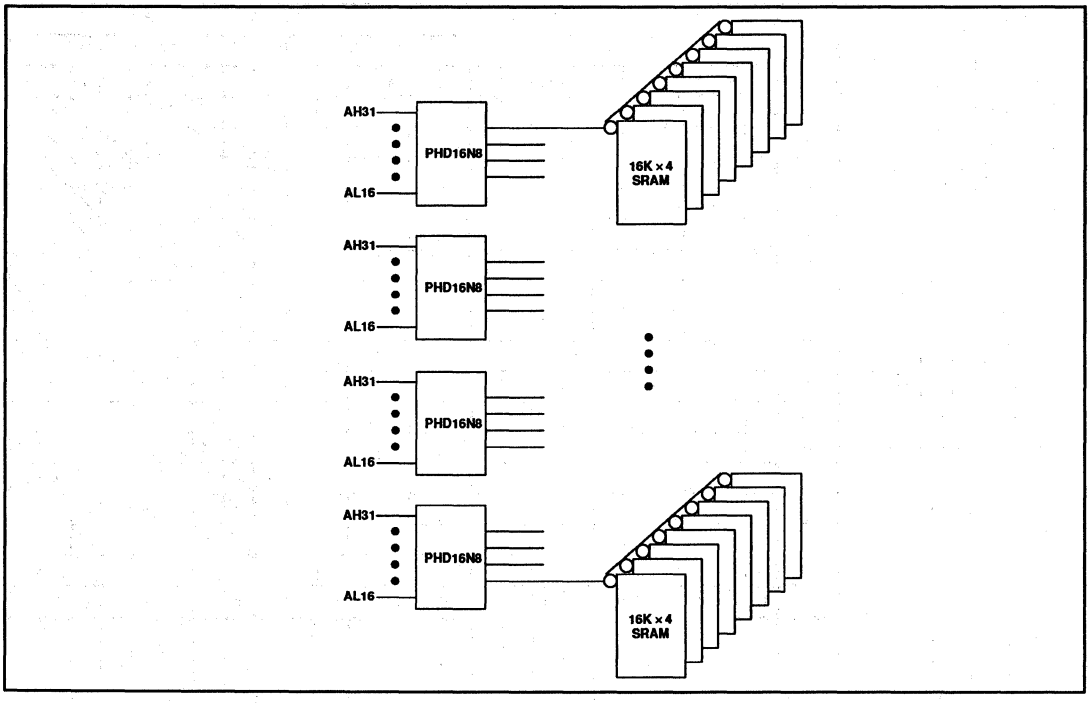
PROGRAM TABLE # _____ REV ___ DATE ___

NOTES:

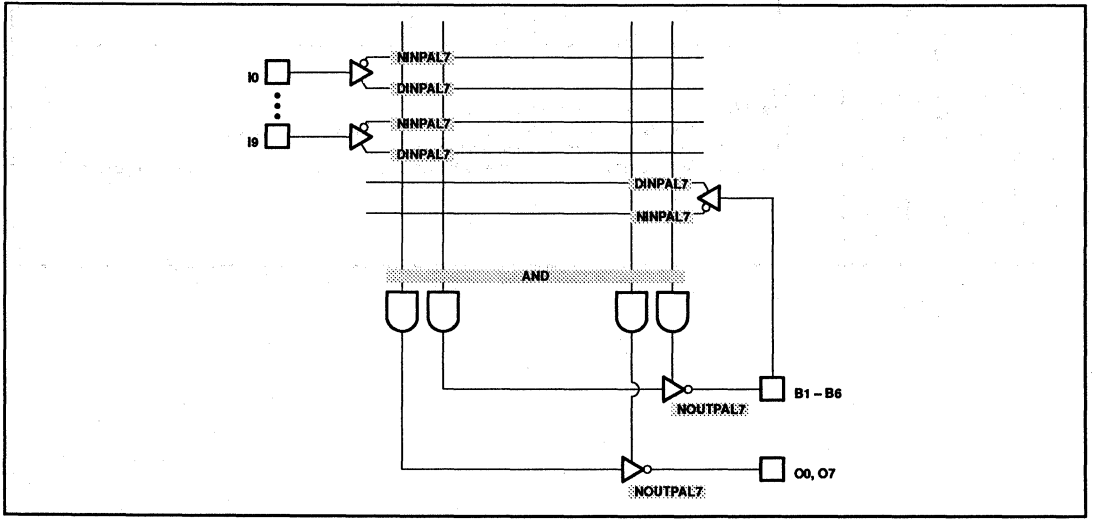
1. The PHD16N8-5 is shipped with all links intact.
2. Unused I and B bits in the AND array exist as INACTIVE in the virgin state.
3. All p-terms are inactive until programmed otherwise.
4. Data cannot be entered into the OR array field due to the fixed nature of the device architecture.

Programmable high-speed decoder logic (16 × 16 × 8) — PHD16N8-5

DECODING 1/2 MEG STATIC MEMORY



SNAP RESOURCE SUMMARY DESIGNATIONS



Zero standby power CMOS versatile PAL devices

PLC18V8Z35/PLC18V8ZI

DESCRIPTION

The PLC18V8Z35 and PLC18V8ZI are universal PAL®[1] devices featuring high performance and virtually zero-standby power for 0power sensitive applications. They are reliable, user-configurable substitutes for discrete TTL/CMOS logic. While compatible with TTL and HCT logic, the PLC18V8ZI can also replace HC logic over the V_{CC} range of 4.5 to 5.5V.

The PLC18V8Z is a two-level logic element comprised of 10 inputs, 74 AND gates (product terms) and 8 output Macro cells.

Each output features an "Output Macro Cell" which can be individually configured as a dedicated input, a combinatorial output, or a registered output with internal feedback. As a result, the PLC18V8Z is capable of emulating all common 20-pin PAL devices to reduce documentation, inventory, and manufacturing costs.

A power-up reset function and a Register Preload function have been incorporated in the PLC18V8Z architecture to facilitate state machine design and testing.

With a standby current of less than 100µA and active power consumption of 1.5mA/MHz, the PLC18V8Z is ideally suited for power sensitive applications in battery operated/backed portable instruments and computers.

The PLC18V8Z is also processed to industrial requirements for operation over an extended temperature range of -40°C to +85°C and supply voltage of 4.5V to 5.5V.

Ordering information can be found below.

FEATURES

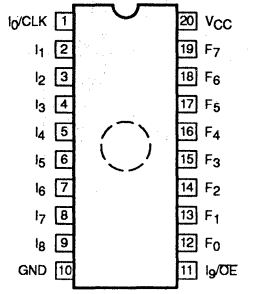
- 20-pin Universal Programmable Array Logic
- Virtually Zero-Standby-power
  - 20µA (typical)
- Available in 300mil-wide DIP with quartz window, plastic DIP (OTP), PLCC (OTP), and SOL (OTP)
- Functional replacement for Series 20 PAL devices
  - I_{OL} = 24mA
- High-performance CMOS EPROM cell technology
  - Erasable
  - Reconfigurable
  - 100% testable
- 35ns Max propagation delay (comm)
- 40ns Max propagation delay (Industrial)
- Up to 18 inputs and 8 input/output macro cells
- Programmable output polarity
- Power-up reset on all registers
- Register Preload capability
- Synchronous Preset/Asynchronous Reset
- Security fuse to prevent duplication of proprietary designs
- Design support provided using SLICE software development package and other CAD tools for PLDs

APPLICATIONS

- Battery powered instruments
- Laptop and pocket computers
- Industrial control
- Medical Instruments
- Portable communications equipment

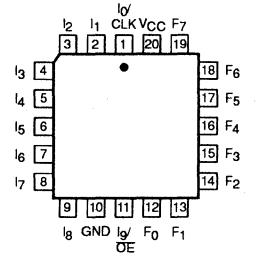
PIN CONFIGURATIONS

D, N, and FA Packages



N = Plastic Dual In-Line Package (DIP) (300mil-wide)
FA = Ceramic DIP with Quartz Window (300mil-wide)
D = Plasitc Small Outline Large Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

PIN LABEL DESCRIPTIONS

I	Dedicated input
B	Bidirectional input/output
O	Dedicated output
D	Registered output (D-type flip-flop)
F	Macrocell Input/Output
CLK	Clock input
OE	Output Enable
V_{CC}	Supply voltage
GND	Ground

ORDERING INFORMATION

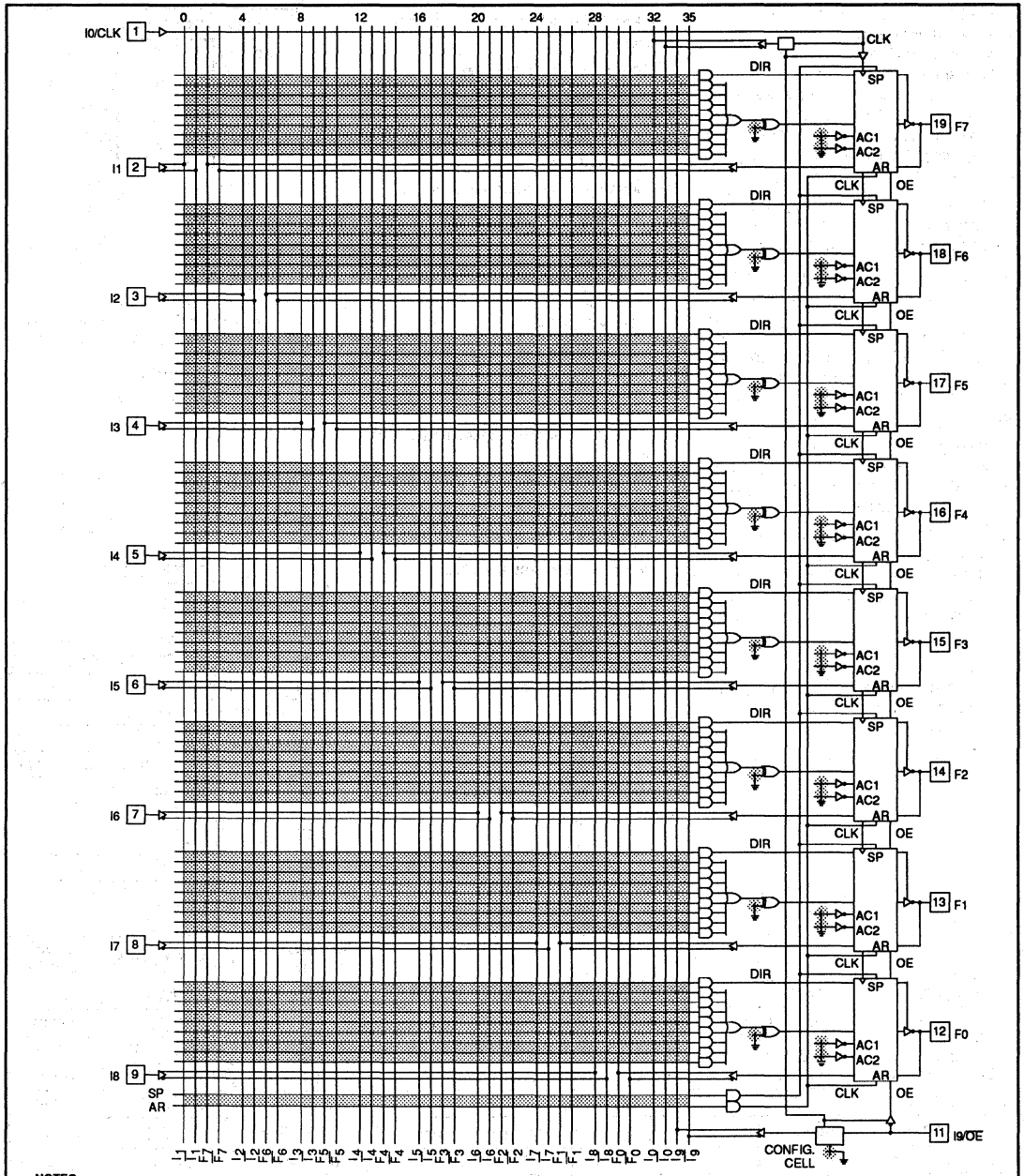
DESCRIPTION	OPERATING CONDITIONS	ORDER CODE	DRAWING NUMBER
20-Pin (300mil-wide) Plastic Dual In-Line Package (t_{PD} = 35ns)	Commercial Temperature Range ± 5% Power Supplies	PLC18V8Z35N	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window (t_{PD} = 35ns)		PLC18V8Z35FA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier Package (t_{PD} = 35ns)		PLC18V8Z35A	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package (t_{PD} = 35ns)		PLC18V8Z35D	0172D
20-Pin (300mil-wide) Plastic Dual In-Line Package (t_{PD} = 40ns)	Industrial Temperature Range ± 10% Power Supplies	PLC18V8ZIN	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window (t_{PD} = 40ns)		PLC18V8ZIFA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier Package (t_{PD} = 40ns)		PLC18V8ZIA	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package (t_{PD} = 40ns)		PLC18V8ZID	0172D

1. PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

Zero standby power CMOS versatile PAL devices

PLC18V8Z35/PLC18V8ZI

LOGIC DIAGRAM



NOTES:
In the unprogrammed or virgin state:

All cells are in a conductive state.
All AND gate locations are pulled to a logic "0" (Low).
Output polarity is inverting.

Pins 1 and 11 are configured as inputs 0 and 9, respectively, via the configuration cell. The clock and OE functions are disabled.
All output macro cells (OMC) are configured as bidirectional I/O, with the outputs disabled via the direction term.
▒ Denotes a programmable cell location.

Zero standby power CMOS versatile PAL devices

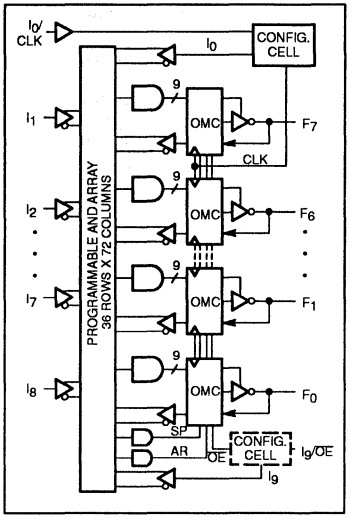
PLC18V8Z35/PLC18V8ZI

PAL DEVICE TO PLC18V8Z OUTPUT PIN CONFIGURATION CROSS REFERENCE

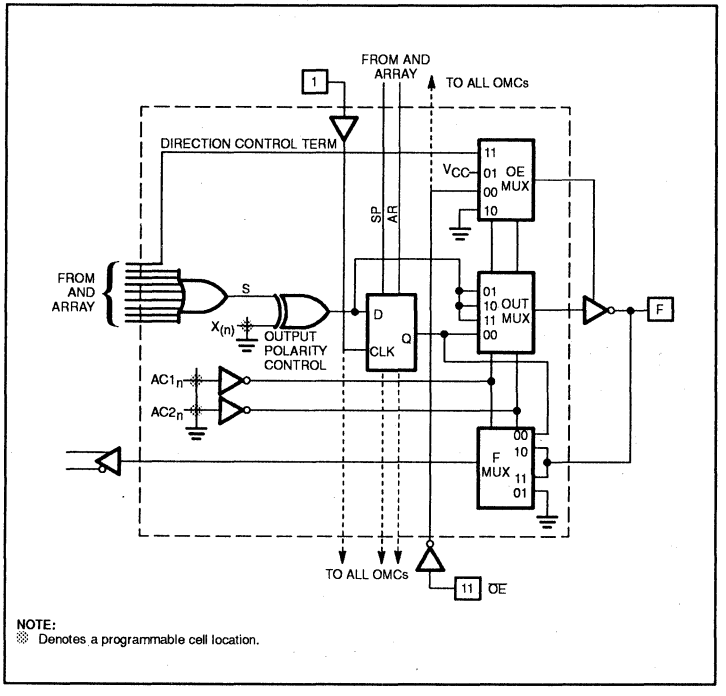
PIN NO.	PLC 18V8Z	16L8 16H8 16P8 16P8	16R4 16RP4	16R6 16RP6	16R8 16RP8	16L2 16H2 16P2	14L4 14H4 14P4	12L6 12H6 12P6	10L8 10H8 10P8
1	I_0/CLK	I	CLK	CLK	CLK	I	I	I	I
19	F7	B	B	B	D	I	I	I	O
18	F6	B	B	D	D	I	I	O	O
17	F5	B	D	D	D	I	O	O	O
16	F4	B	D	D	D	O	O	O	O
15	F3	B	D	D	D	O	O	O	O
14	F2	B	D	D	D	I	O	O	O
13	F1	B	B	D	D	I	I	O	O
12	F0	B	B	B	D	I	I	I	O
11	I_9/$\overline{OE}$	I	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	I	I	I	I

The Philips Semiconductors' state-of-the-art Floating-Gate CMOS EPROM process yields bipolar equivalent performance at less than one-quarter the power consumption. The erasable nature of the EPROM process enables Philips Semiconductors to functionally test the devices prior to shipment to the customer. Additionally, this allows Philips Semiconductors to extensively stress test, as well as ensure the threshold voltage of each individual EPROM cell. 100% programming yield is subsequently guaranteed.

FUNCTIONAL DIAGRAM



OUTPUT MACRO CELL (OMC)



NOTE:
▒ Denotes a programmable cell location.

THE OUTPUT MACRO CELL (OMC)

The PLC18V8Z series devices have 8 individually programmable Output Macro Cells. The 72 AND inputs (or product terms) from the programmable AND array are connected to the 8 OMCs in groups of 9. Eight of the AND terms are dedicated to logic functions; the ninth is for asynchronous direction control, which enables/disables the respective bidirectional I/O pin. Two product terms are dedicated for the Synchronous Preset and Asynchronous Reset functions.

Each OMC can be independently programmed via 16 architecture control bits, $AC1_n$ and $AC2_n$ (one pair per macro cell). Similarly, each OMC has a programmable output polarity control bit (Xn). By configuring the pair of architecture control bits according to the configuration cell table, 4 different configurations may be implemented. Note that the configuration cell is automatically programmed based on the OMC configuration.

DESIGN SECURITY

The PLC18V8Z series devices have a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

CONFIGURATION CELL

A single configuration cell controls the functions of Pins 1 and 11. Refer to Functional Diagram. When the configuration cell is programmed, Pin 1 is a dedicated clock and Pin 11 is dedicated for output enable. When the configuration cell is unprogrammed, Pins 1 and 11 are both dedicated inputs. Note that the output enable for all registered OMCs is common—from Pin 11 only. Output enable control of the bidirectional I/O OMCs is provided from the AND array via the direction product term.

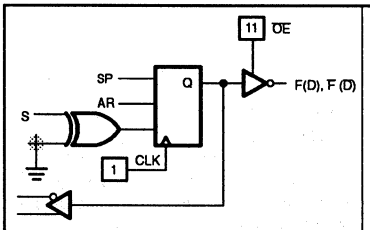
If any one OMC is configured as registered, the configuration cell will be automatically configured (via the design software) to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively. If none of the OMCs are registered, the configuration cell will be programmed such that Pins 1 and 11 are dedicated inputs. The programming codes are as follows:

Pin 1 = CLK, Pin 11 = $\overline{OE}$	L
Pin 1 and Pin 11 = Input	H

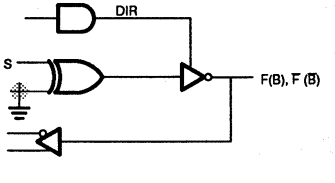
FUNCTION	CONTROL CELL CONFIGURATIONS			COMMENTS
	$AC1_1$	$AC2_N$	CONFIG. CELL	
Registered mode	Programmed	Programmed	Programmed	Dedicated clock from Pin 1. $\overline{OE}$ Control for all registerd OMCs from Pin 11 only.
Bidirectional I/O mode[1]	Unprogrammed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. 3-State control from AND array only.
Fixed input mode	Unprogrammed	Programmed	Unprogrammed	Pins 1 and 11 are dedicated inputs.
Fixed output mode	Programmed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. The feedback path (via F_{MUX}) is disabled.

NOTE:
1. This is the virgin state as shipped from the factory.

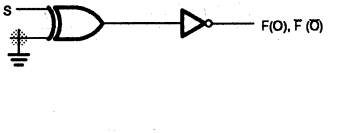
ARCHITECTURE CONTROL—AC1 and AC2



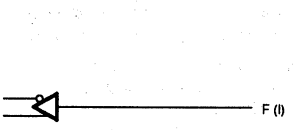
OMC CONFIGURATION	CODE
REGISTERED (D–TYPE)	D



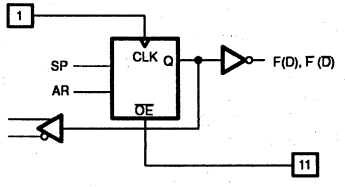
OMC CONFIGURATION	CODE
BIDIRECTIONAL I/O[1] (COMBINATORIAL)	B



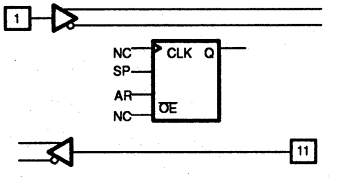
OMC CONFIGURATION	CODE
FIXED OUTPUT	O



OMC CONFIGURATION	CODE
FIXED INPUT	I



CONFIGURATION CELL	CODE
PIN 1 = CLK PIN 11 = $\overline{OE}$	L



CONFIGURATION CELL	CODE
PIN 1 = INPUT PIN 11 = INPUT	H[6]

NOTES:
A factory shipped unprogrammed device is configured such that:
1. This is the initial unprogrammed state. All cells are in a conductive state.
2. All AND gates are pulled to a logic "0" (Low).
3. Output polarity is inverting.
4. Pins 1 and 11 are configured as inputs 0 and 9. The clock and $\overline{OE}$ functions are disabled.
5. All Output Macro Cells (OMCs) are configured as bidirectional I/O, with the outputs disabled via the direction term.
6. This configuration cannot be used if any OMCs are configured as registered (Code = D). The configuration cell will be automatically configured to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively, if any one OMC is programmed as registered.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +7	V_{DC}
V_{CC}	Operating supply voltage	4.5 to 5.5 (Industrial) 4.75 to 5.25 (Commercial)	V_{DC}
V_{IN}	Input voltage	–0.5 to V_{CC} + 0.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} + 0.5	V_{DC}
$\Delta t/\Delta V$	Input/clock transition rise or fall[2]	250	ns/V maximum
I_{IN}	Input currents	–10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	–40 to +85 (Industrial) 0 to +75 (Commercial)	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

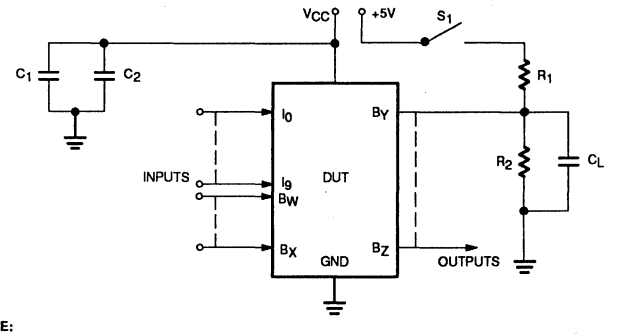
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

2. All digital circuits can oscillate or trigger prematurely when input rise and fall times are very long. When the input signal to a device is at or near the switching threshold, noise on the line will be amplified and can cause oscillation which, if the frequency is low enough, can cause subsequent stages to switch and give erroneous results. For this reason, Schmitt-triggers are recommended if rise/fall times are likely to exceed 250ns at V_{CC} = 4.5V.

THERMAL RATINGS

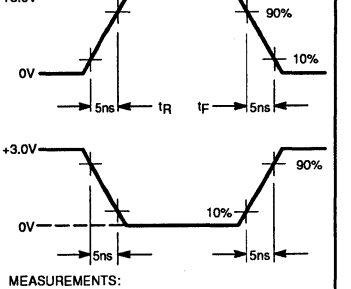
TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

DC ELECTRICAL CHARACTERISTICS

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 20µA V_{CC} = MIN, I_{OL} = 24mA			0.100 0.500	V V
V_{OH}	High	V_{CC} = MIN, I_{OH} = –3.2mA V_{CC} = MIN, I_{OH} = –20µA	2.4 V_{CC} – 0.1V			V V
Input current						
I_{IL}	Low[7]	V_{IN} = GND			–10	µA
I_{IH}	High	V_{IN} = V_{CC}			10	µA
Output current						
$I_{O(OFF)}$	Hi–Z state	V_{OUT} = V_{CC} V_{OUT} = GND			10 –10	µA µA
I_{OS}	Short-circuit[3]	V_{OUT} = GND			–130	mA
I_{CC}	V_{CC} supply current (Standby)	V_{CC} = MAX, V_{IN} = 0 or V_{CC}[8]		20	100	µA
I_{CC}/f	V_{CC} supply current (Active)[4]	V_{CC} = MAX (CMOS inputs)[5, 6]			1.5	mA/MHz
Capacitance						
C_I	Input	V_{CC} = 5V V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

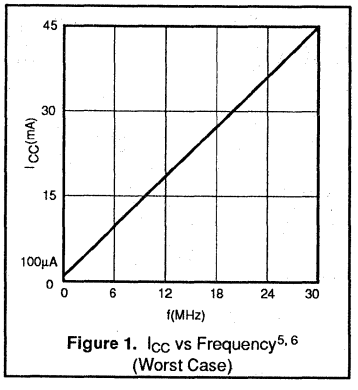


Figure 1. I_{CC} vs Frequency[5, 6] (Worst Case)

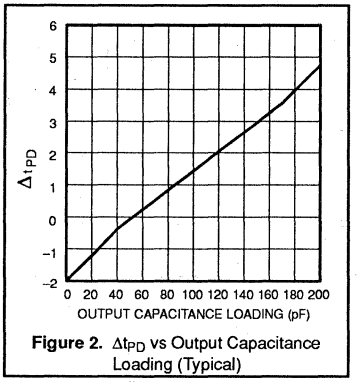


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with TTL input levels: V_{IL} = 0.45V, V_{IH} = 2.4V. Measured with all outputs switching.
5. ΔI_{CC}/TTL input = 2mA.
6. ΔI_{CC} vs frequency (registered configuration) = 2mA/MHz.
7. I_{IL} for Pin 1 (I_0/CLK) is ± 10µA with V_{IN} = 0.4V.
8. V_{IN} includes CLK and $\overline{OE}$ if applicable.

AC ELECTRICAL CHARACTERISTICS[4]

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V; R_2 = 390Ω

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION[1]		PLC18V8Z35 (Commercial)		PLC18V8ZI (Industrial)		UNIT
				R_1 (Ω)	C_L (pF)	MIN	MAX	MIN	MAX	
Pulse width										
t_{CKP}	Clock period (Minimum t_{IS} + t_{CKO})	CLK +	CLK +	200	50	47		57		ns
t_{CKH}	Clock width High	CLK +	CLK –	200	50	20		25		ns
t_{CKL}	Clock width Low	CLK –	CLK +	200	50	20		25		ns
t_{ARW}	Async reset pulse width	I ±, F±	I ∓, F ∓			35		40		ns
Hold time										
t_{IH}	Input or feedback data hold time	CLK +	Input ±	200	50	0		0		ns
Setup time										
t_{IS}	Input or feedback data setup time	I ±, F±	CLK +	200	50	25		30		ns
Propagation delay										
t_{PD}	Delay from input to active output	I ±, F±	F±	200	50		35		40	ns
t_{CKO}	Clock High to output valid access Time	CLK +	F±	200	50		22		27	ns
t_{OE1}[3]	Product term enable to outputs off	I ±, F±	F±	Active-High R = 1.5k Active-Low R = 550	50		35		40	ns
t_{OD1}[2]	Product term disable to outputs off	I ±, F±	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		35		40	ns
t_{OD2}[2]	Pin 11 output disable High to outputs off	OE –	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		25		30	ns
t_{OE2}[3]	Pin 11 output enable to active output	OE +	F±	Active-High R = 1.5k Active-Low R = 550	50		25		30	ns
t_{ARD}	Async reset delay	I ±, F±	F +				35		40	ns
t_{ARR}	Async reset recovery time	I ±, F±	CLK +			25		30		ns
t_{SPR}	Sync preset recovery time	I ±, F±	CLK +			25		30		ns
t_{PPR}	Power-up reset	V_{CC} +	F +				35		40	ns
Frequency of operation										
f_{MAX}	Maximum frequency	I/(t_{IS} + t_{CKO})		200	50		21		18	MHz

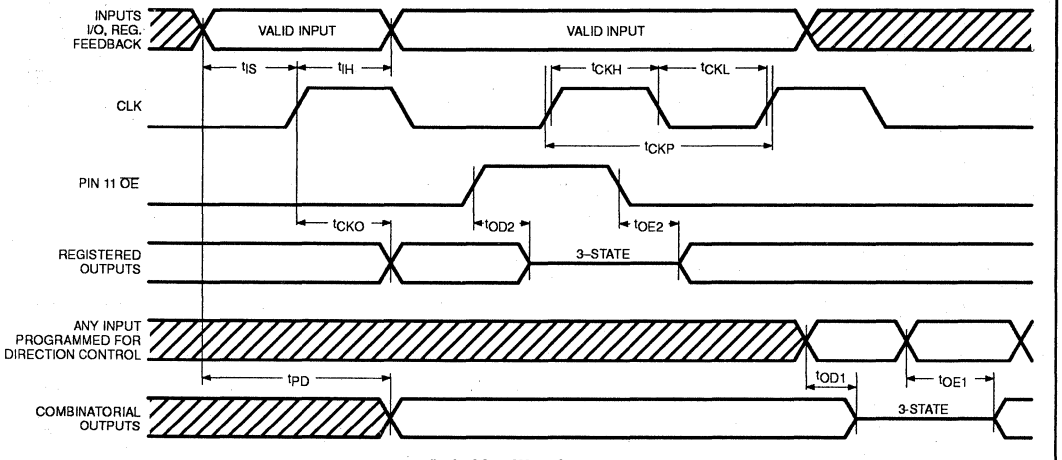
NOTES:
1. Refer also to AC Test Conditions. (Test Load Circuit)
2. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of V_T = (V_{OH} – 0.5V) with S_1 open, and Low-to-High impedance tests are made to the V_T = (V_{OL} + 0.5V) level with S_1 closed.
3. Resistor values of 1.5k and 550Ω provide 3-State levels of 1.0V and 2.0V, respectively. Output timing measurements are to 1.5V level.
4. Leave all the cells on unused product terms intact (unprogrammed) for all patterns.

POWER-UP RESET

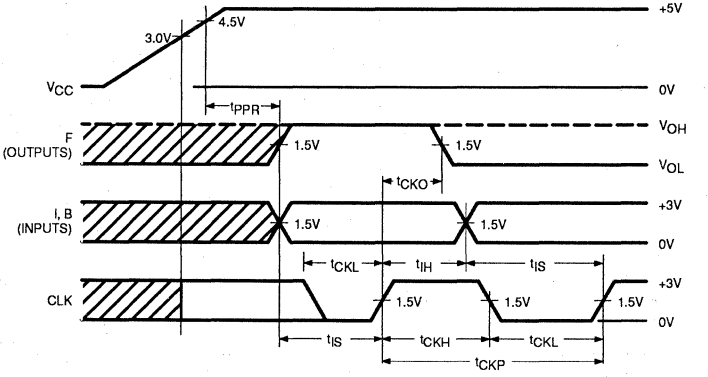
In order to facilitate state machine design and testing, a power-up reset function has been incorporated in the PLC18V8Z. All internal registers will reset to Active-Low (logical "0") after a specified period of time (t_{PPR}).

Therefore, any OMC that has been configured as a registered output will always produce an Active-High on the associated output pin because of the inverted output buffer. The internal feedback (Q) of a registered OMC will also be set Low. The programmed polarity of OMC will not affect the Active-High output condition during a system power-up condition.

TIMING DIAGRAMS



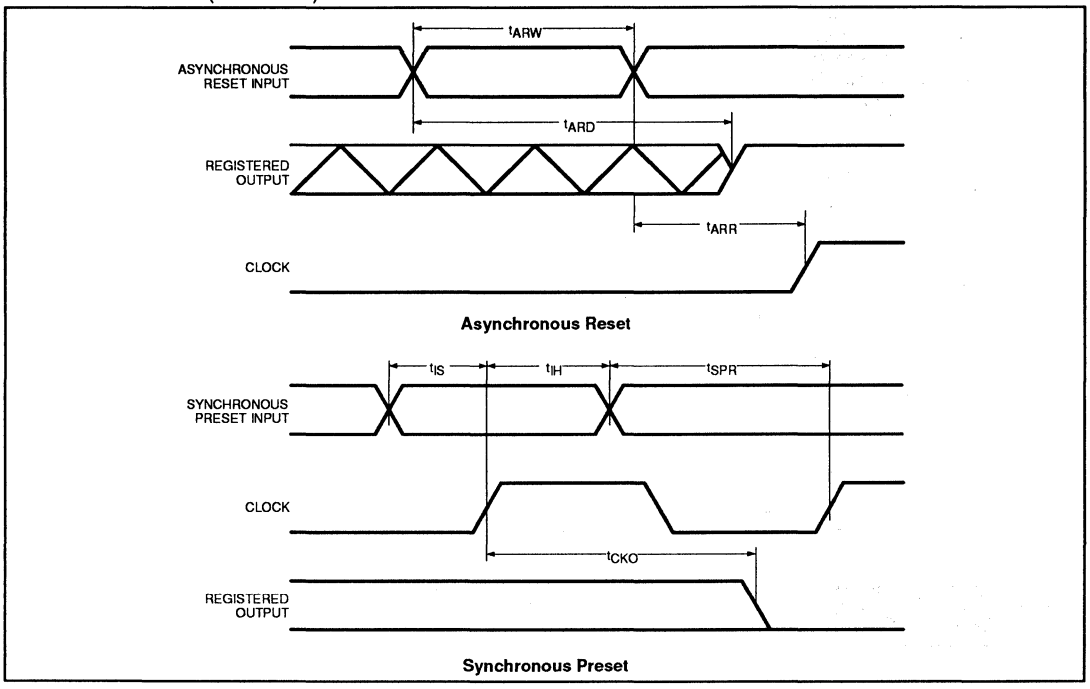
Switching Waveforms



NOTE:
Diagram presupposes that the outputs (F) are enabled. The reset occurs regardless of the output condition (enabled or disabled).

Power-Up Reset

TIMING DIAGRAMS (Continued)



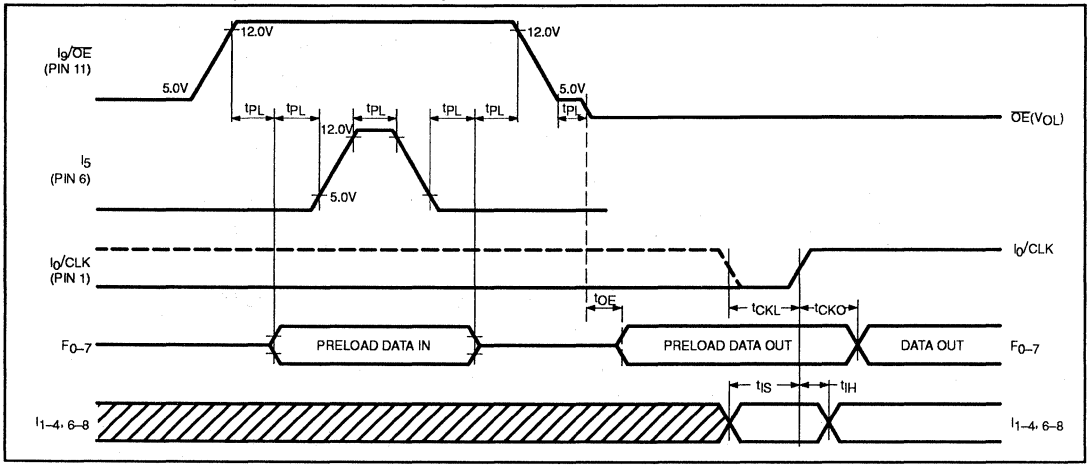
REGISTER PRELOAD FUNCTION (DIAGNOSTIC MODE ONLY)

In order to facilitate the testing of state machine/controller designs, a diagnostic mode register preload feature has been incorporated into the PLC18V8Z series device. This feature enables the user to load the registers with predetermined states while a super voltage is applied to Pins 11 and 6 ($I_9/\overline{OE}$ and I_5). (See diagram for timing and sequence.)

To read the data out, Pins 11 and 6 must be returned to normal TTL levels. The outputs, F_{0-7}, must be enabled in order to read data out. The Q outputs of the registers will reflect data in as input via F_{0-7} during preload. Subsequently, the register Q output via the feedback path will reflect the data in as input via F_{0-7}.

Refer to the voltage waveform for timing and voltage references. t_{PL} = 10µsec.

REGISTER PRELOAD (DIAGNOSTIC MODE)



LOGIC PROGRAMMING

The PLC18V8Z series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™ and CUPL™ 90 design software packages also support the PLC18V8Z architecture.

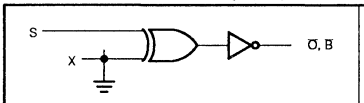
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC18V8Z logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

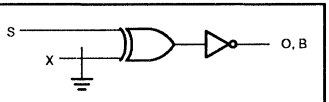
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table. Similarly, various OMC configurations are implemented by programming the Architecture Control bits AC1 and AC2. Note that the configuration cell is automatically programmed based on the OMC configuration.

In this table, the logic state of variables I, P and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

OUTPUT POLARITY – (O, B)

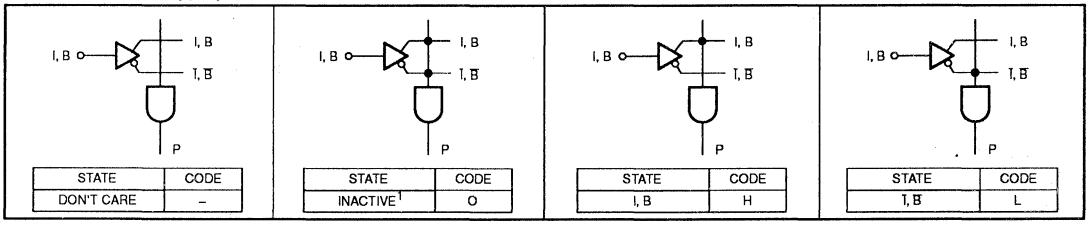


ACTIVE LEVEL	CODE
INVERTING[1]	L



ACTIVE LEVEL	CODE
NON-INVERTING	H

"AND" ARRAY – (I, B)



STATE	CODE
DON'T CARE	–

STATE	CODE
INACTIVE[1]	O

STATE	CODE
I, B	H

STATE	CODE
Ī, B̄	L

NOTE:
1. A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC18V8Z Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lighting could erase a typical PLC18V8Z in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC18V8Z is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PLC18V8Z is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING/SOFTWARE SUPPORT

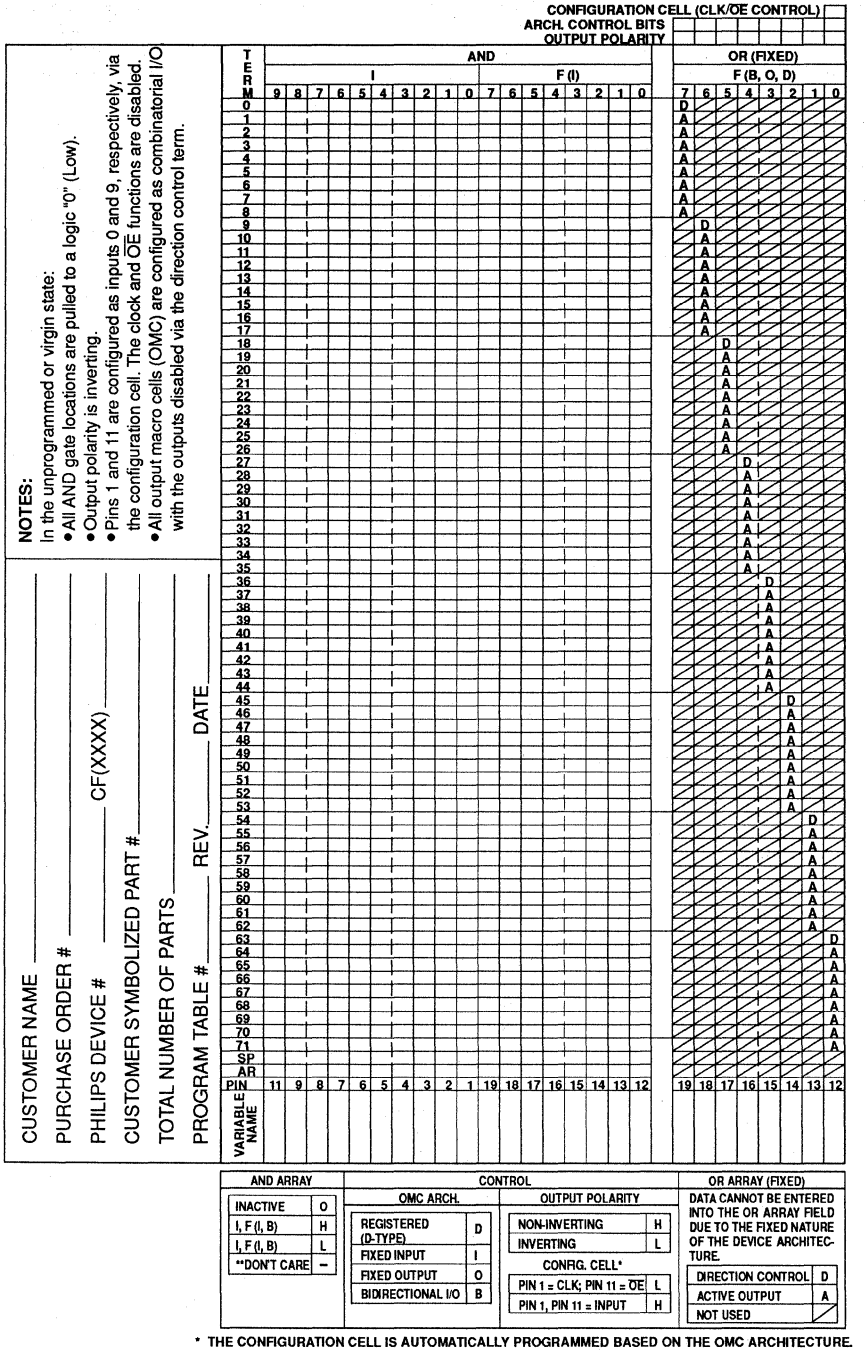
Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Zero standby power CMOS versatile PAL devices

PLC18V8Z35/PLC18V8ZI

PROGRAM TABLE



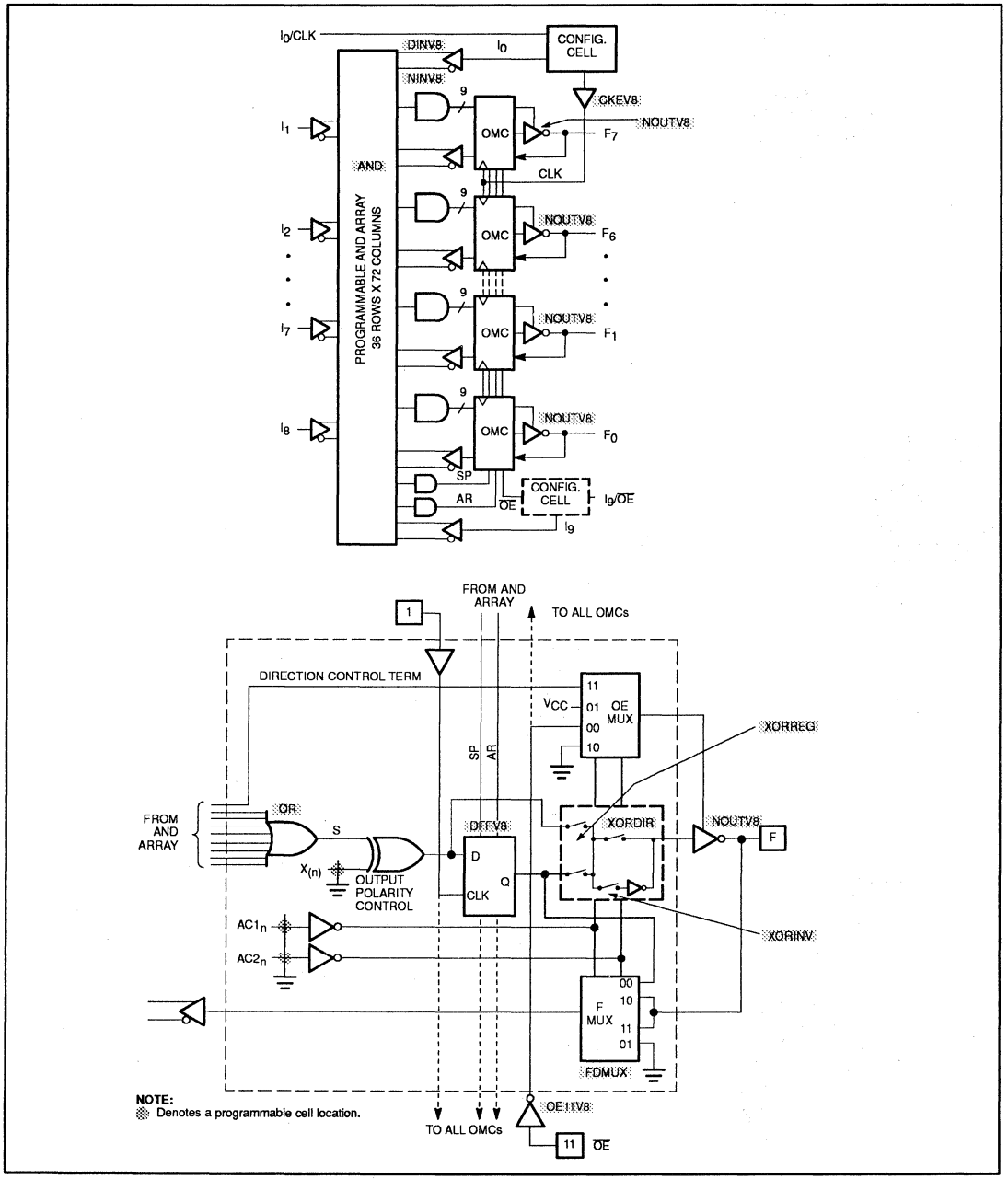
AND ARRAY	
INACTIVE	0
I, F (I, B)	H
I, F (I, B)	L
**DON'T CARE	–

CONTROL	
OMC ARCH.	
REGISTERED (D-TYPE)	D
FIXED INPUT	I
FIXED OUTPUT	O
BIDIRECTIONAL I/O	B
OUTPUT POLARITY	
NON-INVERTING	H
INVERTING	L
CONFIG. CELL*	
PIN 1 = CLK; PIN 11 = OE	L
PIN 1, PIN 11 = INPUT	H

OR ARRAY (FIXED)	
DATA CANNOT BE ENTERED INTO THE OR ARRAY FIELD DUE TO THE FIXED NATURE OF THE DEVICE ARCHITECTURE.	
DIRECTION CONTROL	D
ACTIVE OUTPUT	A
NOT USED	

* THE CONFIGURATION CELL IS AUTOMATICALLY PROGRAMMED BASED ON THE OMC ARCHITECTURE.
** FOR SP, AR: "—" IS NOT ALLOWED.

SNAP RESOURCE SUMMARY DESIGNATIONS



Zero standby power CMOS versatile PAL devices

PLC18V8Z25/PLC18V8ZIA

DESCRIPTION

The PLC18V8Z is a universal PAL® device featuring high performance and virtually zero-standby power for power sensitive applications. They are reliable, user-configurable substitutes for discrete TTL/CMOS logic. While compatible with TTL and HCT logic, the PLC18V8Z can also replace HC logic over the V_{CC} range of 4.5 to 5.5V.

The PLC18V8Z is a two-level logic element comprised of 10 inputs, 74 AND gates (product terms) and 8 output Macro cells.

Each output features an "Output Macro Cell" which can be individually configured as a dedicated input, a combinatorial output, or a registered output with internal feedback. As a result, the PLC18V8Z is capable of emulating all common 20-pin PAL devices to reduce documentation, inventory, and manufacturing costs.

A power-up reset function and a Register Preload function have been incorporated in the PLC18V8Z architecture to facilitate state machine design and testing.

With a standby current of less than 100µA and active power consumption of 1.5mA/MHz, the PLC18V8Z is ideally suited for power sensitive applications in battery operated/backed portable instruments and computers.

The PLC18V8Z is also processed to industrial requirements for operation over an extended temperature range of -40°C to +85°C and supply voltage of 4.5V to 5.5V.

Ordering information can be found below.

FEATURES

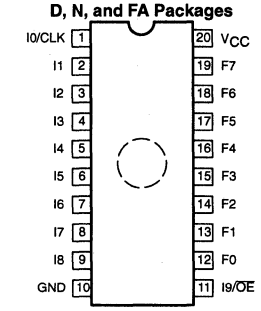
- 20-pin Universal Programmable Array Logic
- Virtually Zero-Standby-power
  - 20µA (typical)
- Available in 300mil-wide DIP with quartz window, plastic DIP (OTP), PLCC (OTP), and SOL (OTP)
- Functional replacement for Series 20 PAL devices
  - I_{OL} = 24mA
- High-performance CMOS EPROM cell technology
  - Erasable
  - Reconfigurable
  - 100% testable
- 25ns Max propagation delay (comm)
- Up to 18 inputs and 8 input/output macro cells
- Programmable output polarity
- Power-up reset on all registers
- Register Preload capability
- Synchronous Preset/Asynchronous Reset
- Security fuse to prevent duplication of proprietary designs
- Design support provided using SLICE software development package and other CAD tools for PLDs

APPLICATIONS

- Battery powered instruments
- Laptop and pocket computers
- Industrial control
- Medical Instruments
- Portable communications equipment

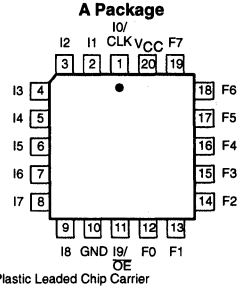
PIN CONFIGURATIONS

D, N, and FA Packages



N = Plastic Dual In-Line Package (DIP) (300mil-wide)
FA = Ceramic DIP with Quartz Window (300mil-wide)
D = Plasitc Small Outline Large Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

PIN LABEL DESCRIPTIONS

I	Dedicated input
B	Bidirectional input/output
O	Dedicated output
D	Registered output (D-type flip-flop)
F	Macrocell Input/Output
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

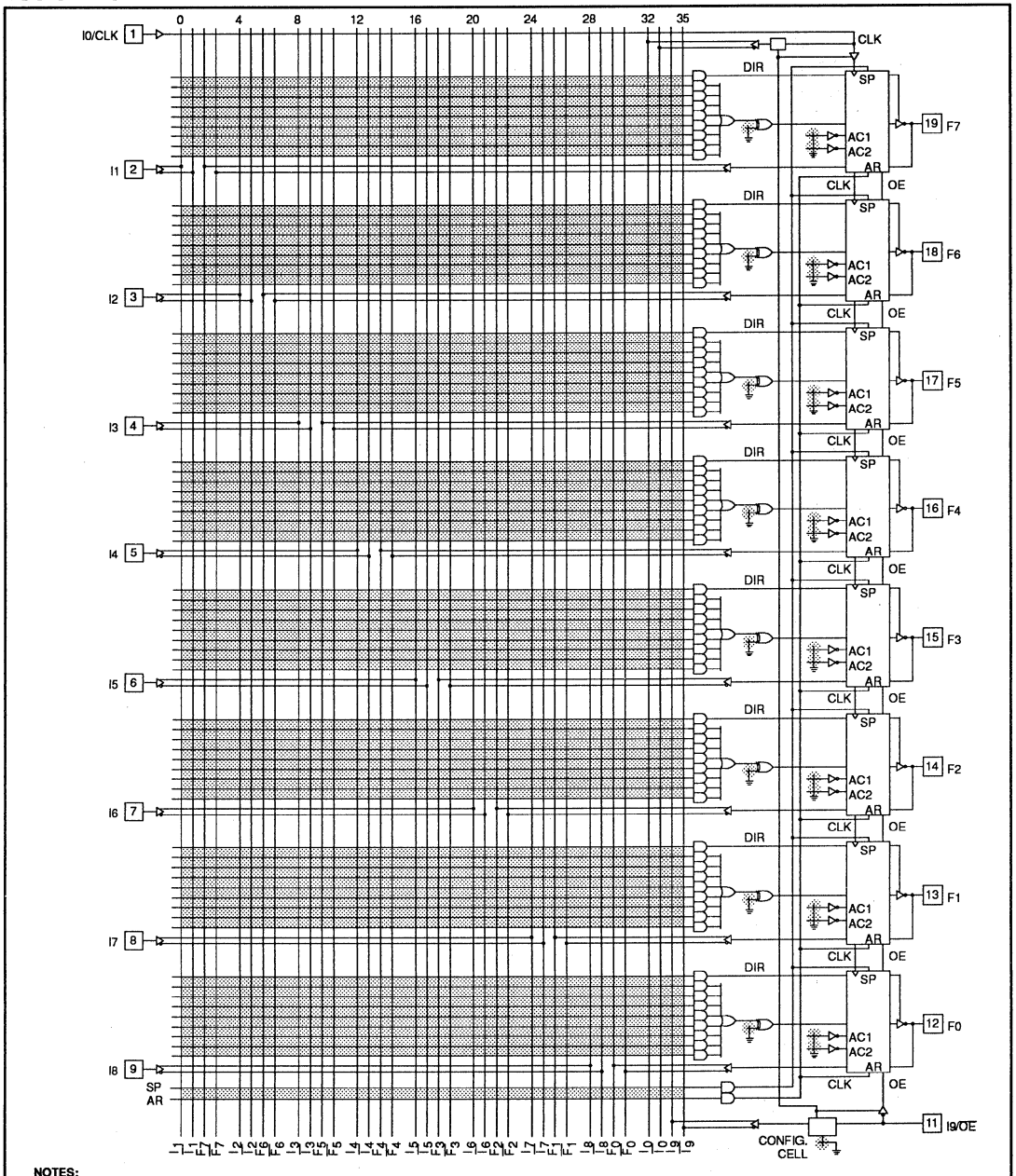
ORDERING INFORMATION

DESCRIPTION	OPERATING CONDITIONS	ORDER CODE	DRAWING NUMBER
20-Pin (300mil-wide) Plastic Dual In-Line Package (t_{PD} = 25ns)	Commercial Temperature Range ± 5% Power Supplies	PLC18V8Z25N	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window (t_{PD} = 25ns)		PLC18V8Z25FA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier (t_{PD} = 25ns)		PLC18V8Z25A	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package (t_{PD} = 25ns)		PLC18V8Z25D	0172D
20-Pin (300mil-wide) Plastic Dual In-Line Package (t_{PD} = 25ns)	Industrial Temperature Range ± 10% Power Supplies	PLC18V8ZIAN	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window (t_{PD} = 25ns)		PLC18V8ZIAFA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier Package (t_{PD} = 25ns)		PLC18V8ZIAA	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package (t_{PD} = 25ns)		PLC18V8ZIAD	0172D

Zero standby power CMOS versatile PAL devices

PLC18V8Z25/PLC18V8ZIA

LOGIC DIAGRAM



NOTES:

In the unprogrammed or virgin state:

All cells are in a conductive state.
All AND gate locations are pulled to a logic "0" (Low).
Output polarity is inverting.

Pins 1 and 11 are configured as inputs 0 and 9, respectively, via the configuration cell. The clock and OE functions are disabled.
All output macro cells (OMC) are configured as bidirectional I/O, with the outputs disabled via the direction term.
▒ Denotes a programmable cell location.

Zero standby power CMOS versatile PAL devices

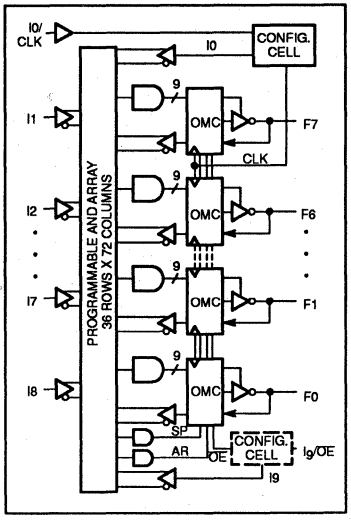
PLC18V8Z25/PLC18V8ZIA

PAL DEVICE TO PLC18V8Z OUTPUT PIN CONFIGURATION CROSS REFERENCE

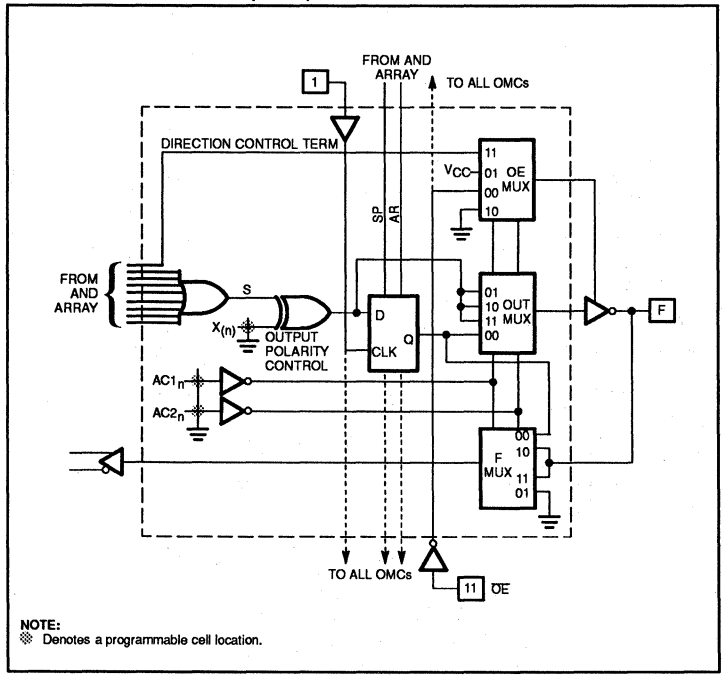
PIN NO.	PLC 18V8Z	16L8 16H8 16P8 16P8	16R4 16RP4	16R6 16RP6	16R8 16RP8	16L2 16H2 16P2	14L4 14H4 14P4	12L6 12H6 12P6	10L8 10H8 10P8
1	I_0/CLK	I	CLK	CLK	CLK	I	I	I	I
19	F7	B	B	B	D	I	I	I	O
18	F6	B	B	D	D	I	I	O	O
17	F5	B	D	D	D	I	O	O	O
16	F4	B	D	D	D	O	O	O	O
15	F3	B	D	D	D	O	O	O	O
14	F2	B	D	D	D	I	O	O	O
13	F1	B	B	D	D	I	I	O	O
12	F0	B	B	B	D	I	I	I	O
11	I_9/$\overline{OE}$	I	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	I	I	I	I

The Philips Semiconductors' state-of-the-art Floating-Gate CMOS EPROM process yields bipolar equivalent performance at less than one-quarter the power consumption. The erasable nature of the EPROM process enables Philips Semiconductors to functionally test the devices prior to shipment to the customer. Additionally, this allows Philips Semiconductors to extensively stress test, as well as ensure the threshold voltage of each individual EPROM cell. 100% programming yield is subsequently guaranteed.

FUNCTIONAL DIAGRAM



OUTPUT MACRO CELL (OMC)



NOTE:
※ Denotes a programmable cell location.

THE OUTPUT MACRO CELL (OMC)

The PLC18V8Z series devices have 8 individually programmable Output Macro Cells. The 72 AND inputs (or product terms) from the programmable AND array are connected to the 8 OMCs in groups of 9. Eight of the AND terms are dedicated to logic functions; the ninth is for asynchronous direction control, which enables/disables the respective bidirectional I/O pin. Two product terms are dedicated for the Synchronous Preset and Asynchronous Reset functions.

Each OMC can be independently programmed via 16 architecture control bits, $AC1_n$ and $AC2_n$ (one pair per macro cell). Similarly, each OMC has a programmable output polarity control bit (Xn). By configuring the pair of architecture control bits according to the configuration cell table, 4 different configurations may be implemented. Note that the configuration cell is automatically programmed based on the OMC configuration.

DESIGN SECURITY

The PLC18V8Z series devices have a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

CONFIGURATION CELL

A single configuration cell controls the functions of Pins 1 and 11. Refer to Functional Diagram. When the configuration cell is programmed, Pin 1 is a dedicated clock and Pin 11 is dedicated for output enable. When the configuration cell is unprogrammed, Pins 1 and 11 are both dedicated inputs. Note that the output enable for all registered OMCs is common—from Pin 11 only. Output enable control of the bidirectional I/O OMCs is provided from the AND array via the direction product term.

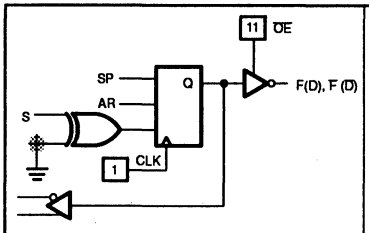
If any one OMC is configured as registered, the configuration cell will be automatically configured (via the design software) to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively. If none of the OMCs are registered, the configuration cell will be programmed such that Pins 1 and 11 are dedicated inputs. The programming codes are as follows:

Pin 1 = CLK, Pin 11 = $\overline{OE}$	L
Pin 1 and Pin 11 = Input	H

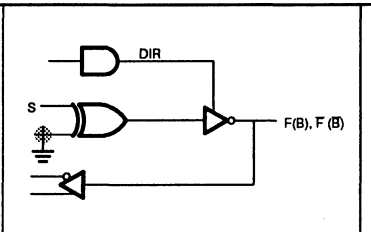
FUNCTION	CONTROL CELL CONFIGURATIONS			COMMENTS
	$AC1_1$	$AC2_N$	CONFIG. CELL	
Registered mode	Programmed	Programmed	Programmed	Dedicated clock from Pin 1. $\overline{OE}$ Control for all registerd OMCs from Pin 11 only.
Bidirectional I/O mode[1]	Unprogrammed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. 3-State control from AND array only.
Fixed input mode	Unprogrammed	Programmed	Unprogrammed	Pins 1 and 11 are dedicated inputs.
Fixed output mode	Programmed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. The feedback path (via F_{MUX}) is disabled.

NOTE:
1. This is the virgin state as shipped from the factory.

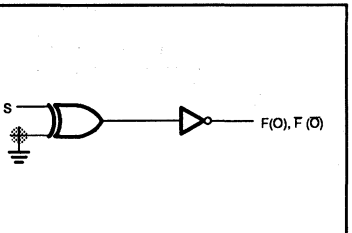
ARCHITECTURE CONTROL—AC1 and AC2



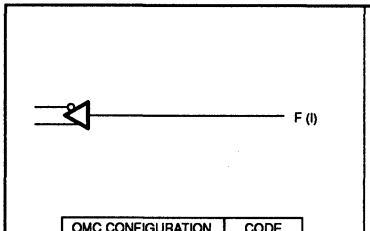
OMC CONFIGURATION	CODE
REGISTERED (D–TYPE)	D



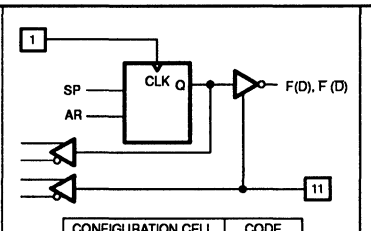
OMC CONFIGURATION	CODE
BIDIRECTIONAL I/O[1] (COMBINATORIAL)	B



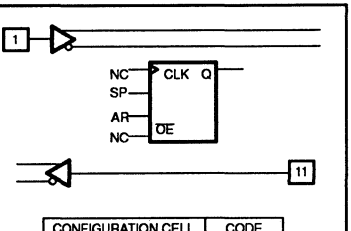
OMC CONFIGURATION	CODE
FIXED OUTPUT	O



OMC CONFIGURATION	CODE
FIXED INPUT	I



CONFIGURATION CELL	CODE
PIN 1 = CLK PIN 11 = $\overline{OE}$	L



CONFIGURATION CELL	CODE
PIN 1 = INPUT PIN 11 = INPUT	H[6]

NOTES:
A factory shipped unprogrammed device is configured such that:
1. This is the initial unprogrammed state. All cells are in a conductive state.
2. All AND gates are pulled to a logic "0" (Low).
3. Output polarity is inverting.
4. Pins 1 and 11 are configured as inputs 0 and 9. The clock and $\overline{OE}$ functions are disabled.
5. All Output Macro Cells (OMCs) are configured as bidirectional I/O, with the outputs disabled via the direction term.
6. This configuration cannot be used if any OMCs are configured as registered (Code = D).

Zero standby power CMOS versatile PAL devices

PLC18V8Z25/PLC18V8ZIA

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +7	V_{DC}
V_{CC}	Operating supply voltage	4.5 to 5.5 (Industrial) 4.75 to 5.25 (Commercial)	V_{DC}
V_{IN}	Input voltage	–0.5 to V_{CC} + 0.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} + 0.5	V_{DC}
$\Delta t/\Delta V$	Input/clock transition rise or fall[2]	250	ns/V maximum
I_{IN}	Input currents	–10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	–40 to +85 (Industrial) 0 to +75 (Commercial)	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

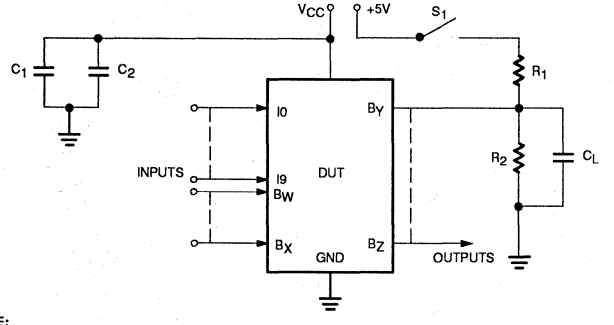
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

2. All digital circuits can oscillate or trigger prematurely when input rise and fall times are very long. When the input signal to a device is at or near the switching threshold, noise on the line will be amplified and can cause oscillation which, if the frequency is low enough, can cause subsequent stages to switch and give erroneous results. For this reason, Schmitt-triggers are recommended if rise/fall times are likely to exceed 250ns at V_{CC} = 4.5V.

THERMAL RATINGS

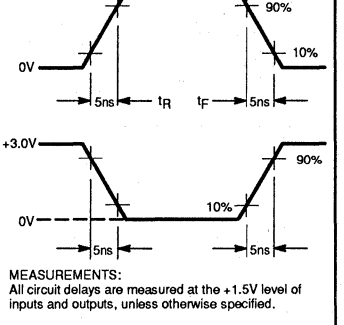
TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

DC ELECTRICAL CHARACTERISTICS

Commercial = $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$;
Industrial = $-40°C \le T_{amb} \le +85°C$, $4.5V \le V_{CC} \le 5.5V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 20µA V_{CC} = MIN, I_{OL} = 24mA			0.100 0.500	V V
V_{OH}	High	V_{CC} = MIN, I_{OH} = –3.2mA V_{CC} = MIN, I_{OH} = –20µA	2.4 V_{CC} – 0.1V			V V
Input current						
I_{IL}	Low[7]	V_{IN} = GND			–10	µA
I_{IH}	High	V_{IN} = V_{CC}			10	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC} V_{OUT} = GND			10 –10	µA µA
I_{OS}	Short-circuit[3]	V_{OUT} = GND			–130	mA
I_{CC}	V_{CC} supply current (Standby)	V_{CC} = MAX, V_{IN} = 0 or V_{CC}[8]		20	100	µA
I_{CC}/f	V_{CC} supply current (Active)[4]	V_{CC} = MAX (CMOS inputs)[5, 6]			1.5	mA/MHz
Capacitance						
C_I	Input	V_{CC} = 5V V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

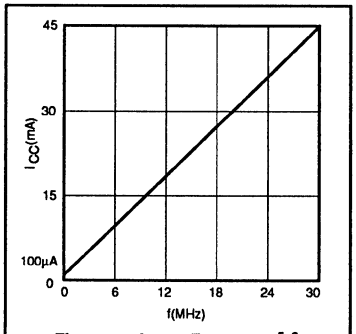


Figure 1. I_{CC} vs Frequency[5, 6] (Worst Case)

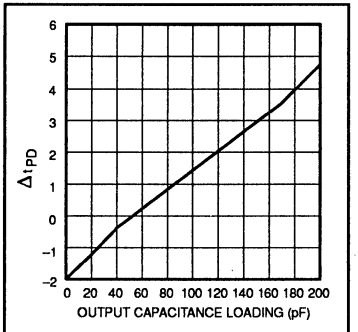


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with TTL input levels: V_{IL} = 0.45V, V_{IH} = 2.4V. Measured with all outputs switching.
5. ΔI_{CC}/TTL input = 2mA.
6. ΔI_{CC} vs frequency (registered configuration) = 2mA/MHz.
7. I_{IL} for Pin 1 (I_O/CLK) is ± 10µA with V_{IN} = 0.4V.
8. V_{IN} includes CLK and $\overline{OE}$ if applicable.

AC ELECTRICAL CHARACTERISTICS[4]

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V; R_2 = 390Ω

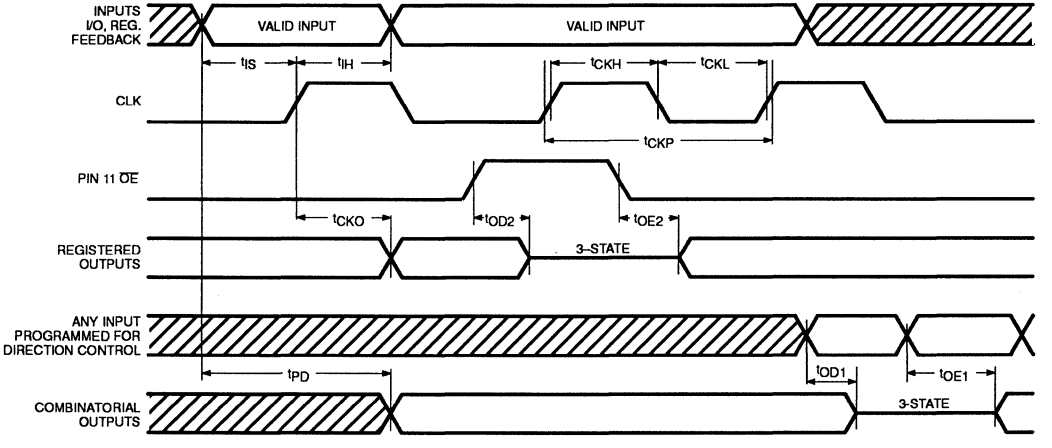
SYMBOL	PARAMETER	FROM	TO	TEST CONDITION[1]		PLC18V8Z25 (Commercial)		PLC18V8ZIA (Industrial)		UNIT
				R_1 (Ω)	C_L (pF)	MIN	MAX	MIN	MAX	
Pulse width										
t_{CKP}	Clock period (Minimum $t_{IS} + t_{CKO}$)	CLK +	CLK +	200	50	33		33		ns
t_{CKH}	Clock width High	CLK +	CLK –	200	50	15		15		ns
t_{CKL}	Clock width Low	CLK –	CLK +	200	50	15		15		ns
t_{ARW}	Async reset pulse width	I ±, F±	I ∓, F ∓			25		25		ns
Hold time										
t_{IH}	Input or feedback data hold time	CLK +	Input ±	200	50	0		0		ns
Setup time										
t_{IS}	Input or feedback data setup time	I ±, F±	CLK +	200	50	18		18		ns
Propagation delay										
t_{PD}	Delay from input to active output	I ±, F±	F±	200	50		25		25	ns
t_{CKO}	Clock High to output valid access Time	CLK +	F±	200	50		15		15	ns
t_{OE1}[3]	Product term enable to outputs off	I ±, F±	F±	Active-High R = 1.5k Active-Low R = 550	50		25		25	ns
t_{OD1}[2]	Product term disable to outputs off	I ±, F±	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		25		25	ns
t_{OD2}[2]	Pin 11 output disable High to outputs off	OE –	F±	From V_{OH} R = ∞ From V_{OL} R = 200	5		20		20	ns
t_{OE2}[3]	Pin 11 output enable to active output	OE +	F±	Active-High R = 1.5k Active-Low R = 550	50		20		20	ns
t_{ARD}	Async reset delay	I ±, F±	F +				30		30	ns
t_{ARR}	Async reset recovery time	I ±, F±	CLK +			20		20		ns
t_{SPR}	Sync preset recovery time	I ±, F±	CLK +			20		20		ns
t_{PPR}	Power-up reset	V_{CC} +	F +				25		25	ns
Frequency of operation										
f_{MAX}	Maximum frequency	$1/(t_{IS} + t_{CKO})$		200	50		30		30	MHz

NOTES:
1. Refer also to AC Test Conditions. (Test Load Circuit)
2. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. Resistor values of 1.5k and 550Ω provide 3-State levels of 1.0V and 2.0V, respectively. Output timing measurements are to 1.5V level.
4. Leave all the cells on unused product terms intact (unprogrammed) for all patterns.

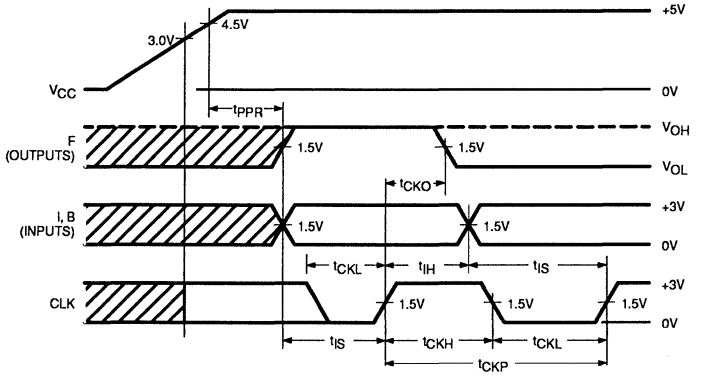
POWER-UP RESET

In order to facilitate state machine design and testing, a power-up reset function has been incorporated in the PLC18V8Z. All internal registers will reset to Active-Low (logical "0") after a specified period of time (t_{PPR}). Therefore, any OMC that has been configured as a registered output will always produce an Active-High on the associated output pin because of the inverted output buffer. The internal feedback (Q) of a registered OMC will also be set Low. The programmed polarity of OMC will not affect the Active-High output condition during a system power-up condition.

TIMING DIAGRAMS



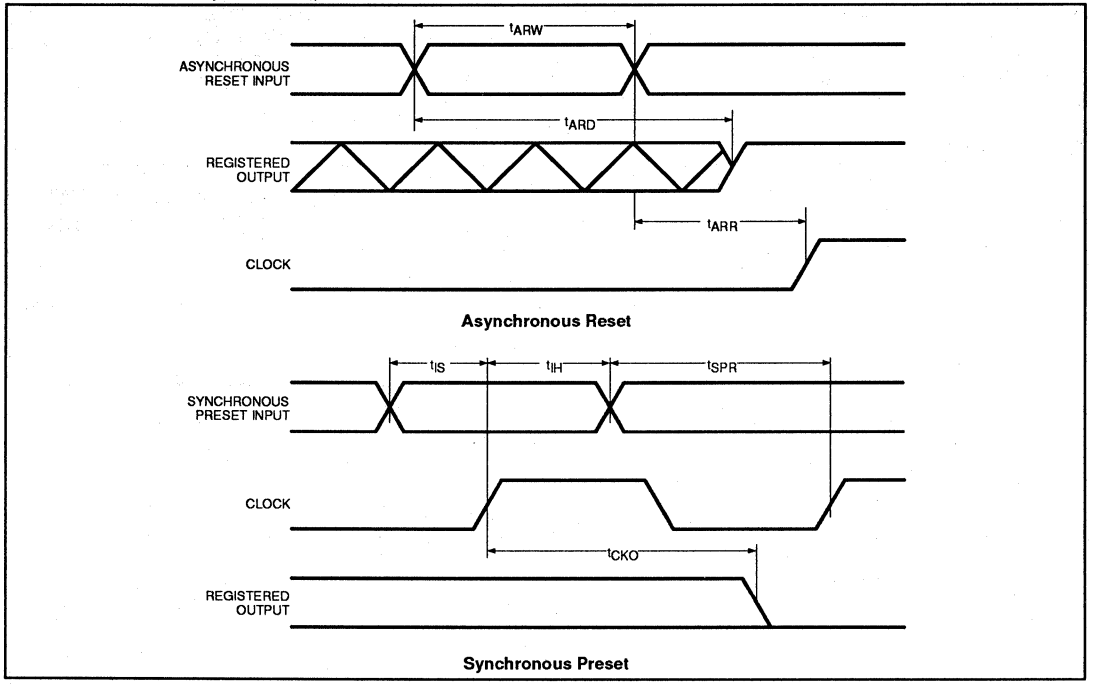
Switching Waveforms



NOTE:
Diagram presupposes that the outputs (F) are enabled. The reset occurs regardless of the output condition (enabled or disabled).

Power-Up Reset

TIMING DIAGRAMS (Continued)



Asynchronous Reset

Synchronous Preset

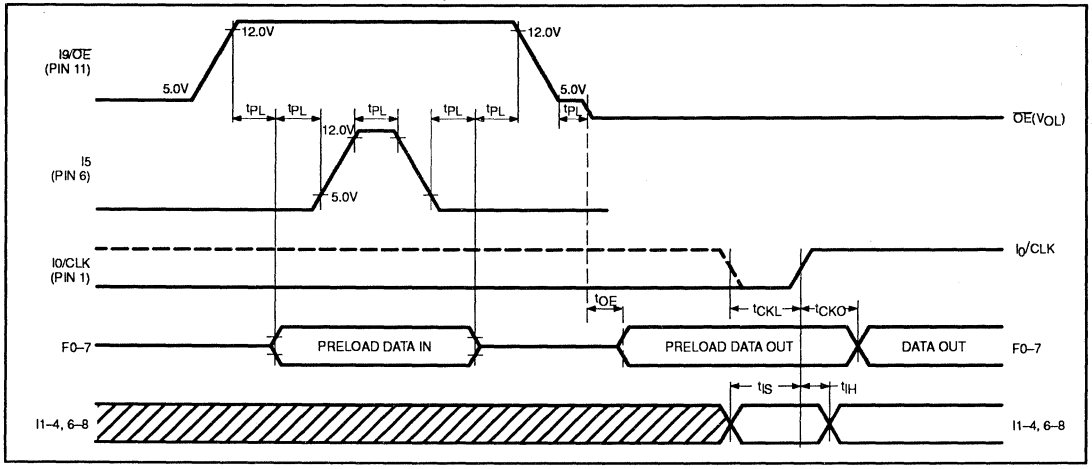
REGISTER PRELOAD FUNCTION (DIAGNOSTIC MODE ONLY)

In order to facilitate the testing of state machine/controller designs, a diagnostic mode register preload feature has been incorporated into the PLC18V8Z series device. This feature enables the user to load the registers with predetermined states while a super voltage is applied to Pins 11 and 6 (I9/$\overline{OE}$ and I5). (See diagram for timing and sequence.)

To read the data out, Pins 11 and 6 must be returned to normal TTL levels. The outputs, F0 – F7, must be enabled in order to read data out. The Q outputs of the registers will reflect data in as input via F0 – F7 during preload. Subsequently, the register Q output via the feedback path will reflect the data in as input via F0 – F7.

Refer to the voltage waveform for timing and voltage references. t_{PL} = 10µsec.

REGISTER PRELOAD (DIAGNOSTIC MODE)



LOGIC PROGRAMMING

The PLC18V8Z series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLC18V8Z architecture.

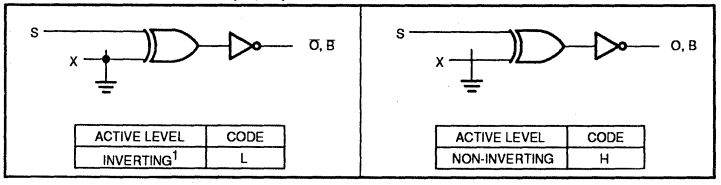
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC18V8Z logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table. Similarly, various OMC configurations are implemented by programming the Architecture Control bits AC1 and AC2. Note that the configuration cell is automatically programmed based on the OMC configuration.

In this table, the logic state of variables I, P and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

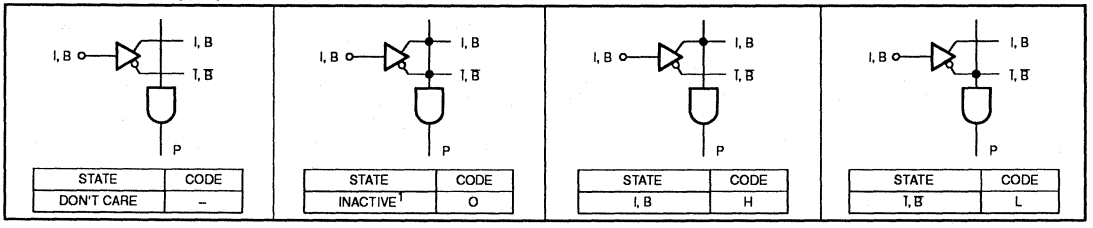
OUTPUT POLARITY – (O, B)



ACTIVE LEVEL	CODE
INVERTING[1]	L

ACTIVE LEVEL	CODE
NON-INVERTING	H

"AND" ARRAY – (I, B)



STATE	CODE
DON'T CARE	–

STATE	CODE
INACTIVE[1]	O

STATE	CODE
I, B	H

STATE	CODE
I, B	L

NOTE:
1. A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC18V8Z Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lighting could erase a typical PLC18V8Z in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC18V8Z is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PLC18V8Z is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING/SOFTWARE SUPPORT

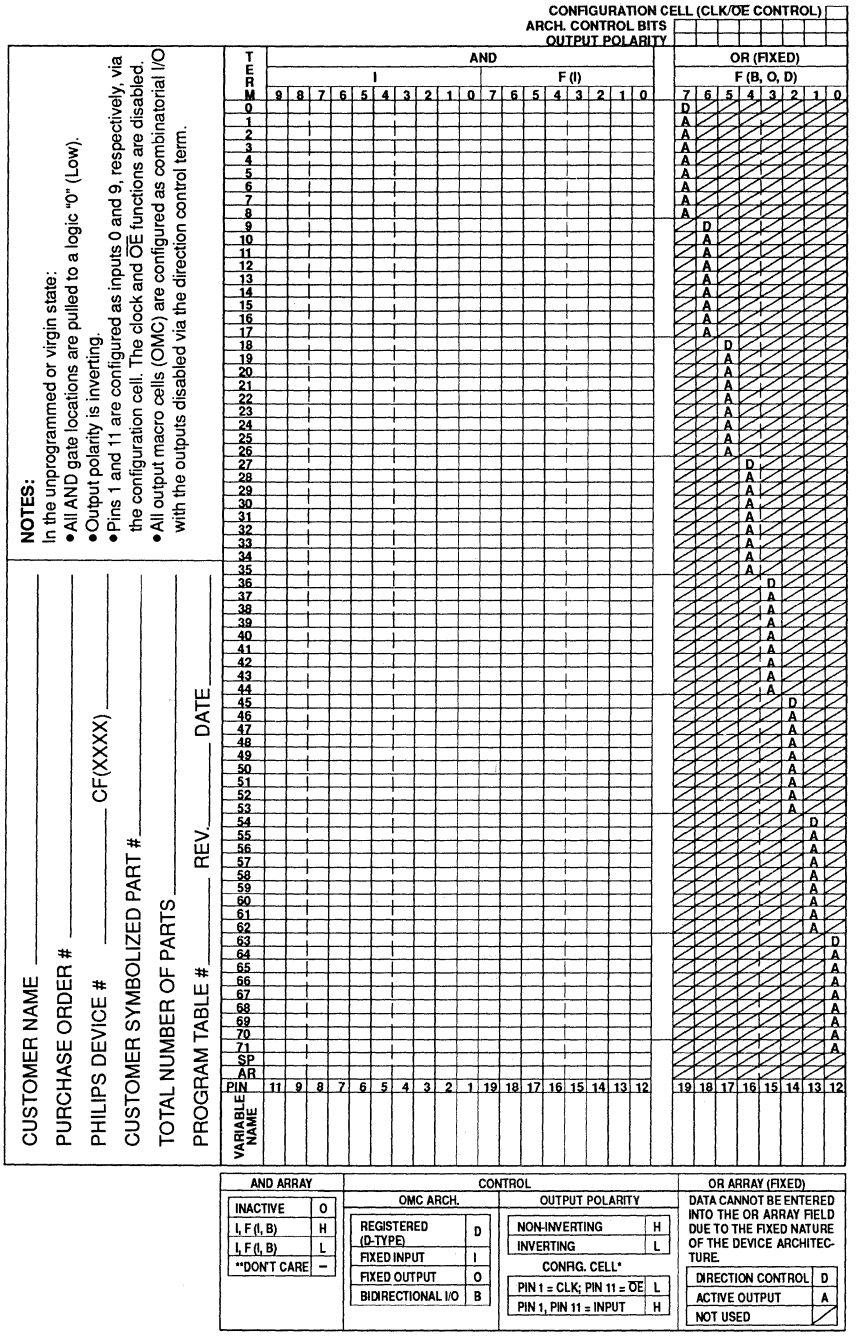
Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Zero standby power CMOS versatile PAL devices

PLC18V8Z25/PLC18V8ZIA

PROGRAM TABLE



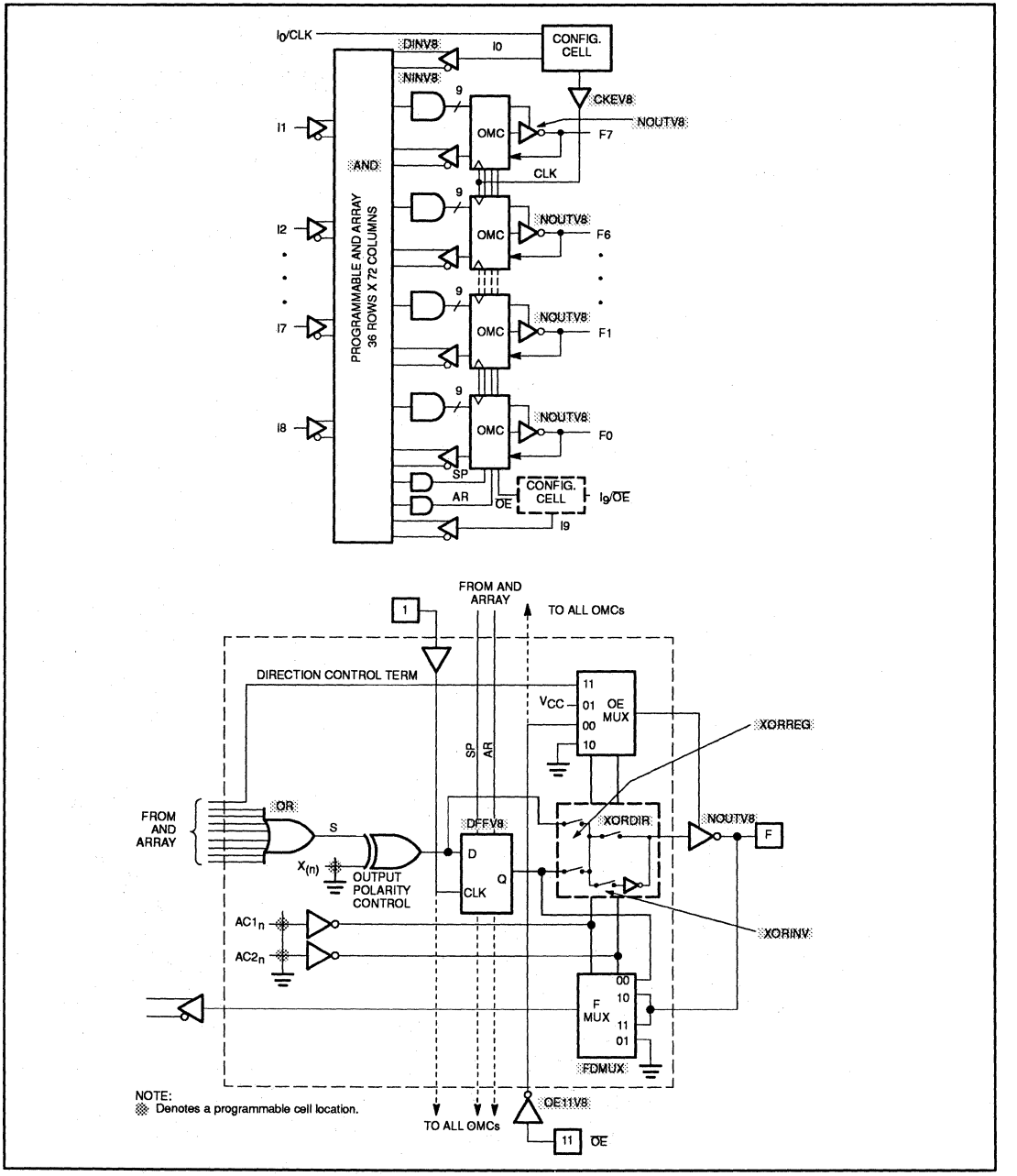
AND ARRAY	
INACTIVE	0
I, F (I, B)	H
I, F (I, B)	L
**DON'T CARE	–

CONTROL			
OMC ARCH.		OUTPUT POLARITY	
REGISTERED (D-TYPE)	D	NON-INVERTING	H
FIXED INPUT	I	INVERTING	L
FIXED OUTPUT	O	CONFIG. CELL*	
BIDIRECTIONAL I/O	B	PIN 1 = CLK; PIN 11 = $\overline{OE}$	L
		PIN 1, PIN 11 = INPUT	H

OR ARRAY (FIXED)	
DATA CANNOT BE ENTERED INTO THE OR ARRAY FIELD DUE TO THE FIXED NATURE OF THE DEVICE ARCHITECTURE.	
DIRECTION CONTROL	D
ACTIVE OUTPUT	A
NOT USED	

* THE CONFIGURATION CELL IS AUTOMATICALLY PROGRAMMED BASED ON THE OMC ARCHITECTURE.
** FOR SP, AR: "–" IS NOT ALLOWED.

SNAP RESOURCE SUMMARY DESIGNATIONS



PAL devices
16L8, 16R8, 16R6, 16R4

PLUS16R8D/-7 SERIES

FEATURES

- Ultra high-speed
  - t_{PD} = 7.5ns and f_{MAX} = 74MHz for the PLUS16R8-7 Series
  - t_{PD} = 10ns and f_{MAX} = 60 MHz for the PLUS16R8D Series
- 100% functionally and pin-for-pin compatible with industry standard 20-pin PAL® ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via SNAP and other CAD tools for Series 20 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs

DESCRIPTION

The Philips Semiconductors PLUS16XX family consists of ultra high-speed 7.5ns and 10ns versions of Series 20 PAL devices.

The PLUS16XX family is 100% functional and pin-compatible with the 16L8, 16R8, 16R6, and 16R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 programmable AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLUS16R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to Active-Low after a specific period of time.

The Philips Semiconductors State-of-the-Art oxide isolation Bipolar fabrication process is employed to achieve high-performance operation.

The PLUS16XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The SNAP software package from Philips Semiconductors supports easy design entry for the PLUS16XX series as well as other PLD devices from Philips Semiconductors. The PLUS16XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLUS16L8	10	8 (6 I/O)	0
PLUS16R8	8	0	8
PLUS16R6	8	2 I/O	6
PLUS16R4	8	4 I/O	4

ORDERING INFORMATION

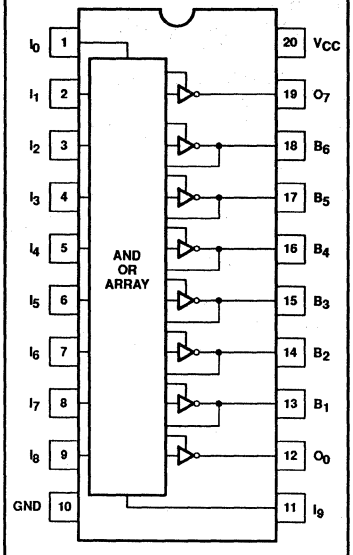
DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual-In-Line 300mil-wide	PLUS16R8DN PLUS16R6DN PLUS16R4DN PLUS16L8DN PLUS16R8–7N PLUS16R6–7N PLUS16R4–7N PLUS16L8–7N	0408B
20-Pin Plastic Leaded Chip Carrier (PLCC)	PLUS16R8DA PLUS16R6DA PLUS16R4DA PLUS16L8DA PLUS16R8–7A PLUS16R6–7A PLUS16R4–7A PLUS16L8–7A	0400E

NOTE:
The PLUS16XX series of devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Philips Semiconductors Military Data Book.

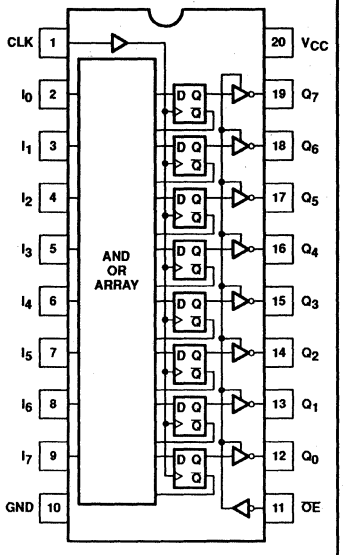
®PAL is a registered trademark of Advanced Micro Devices, Inc.

PIN CONFIGURATIONS

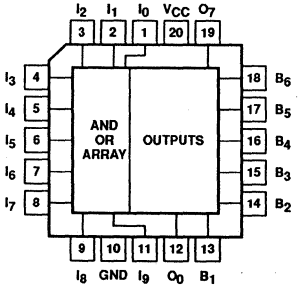
PLUS16L8



PLUS16R8

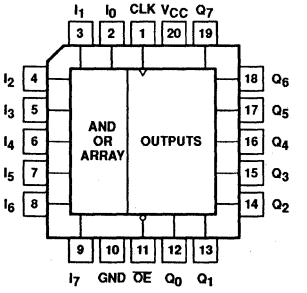


PLUS16L8



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

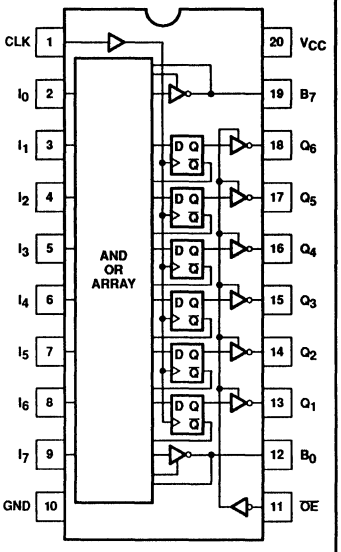
PLUS16R8



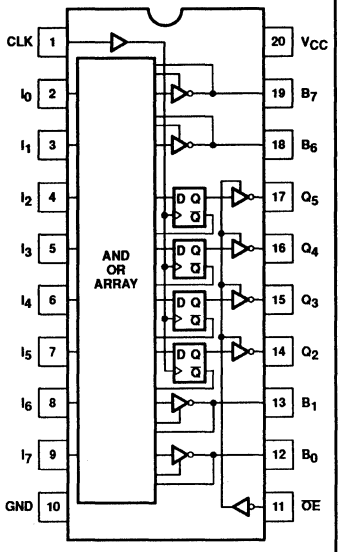
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

PIN CONFIGURATIONS

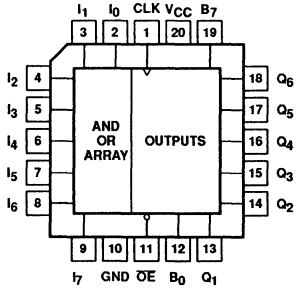
PLUS16R6



PLUS16R4

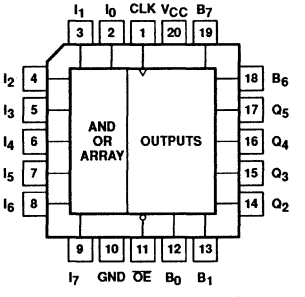


PLUS16R6



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

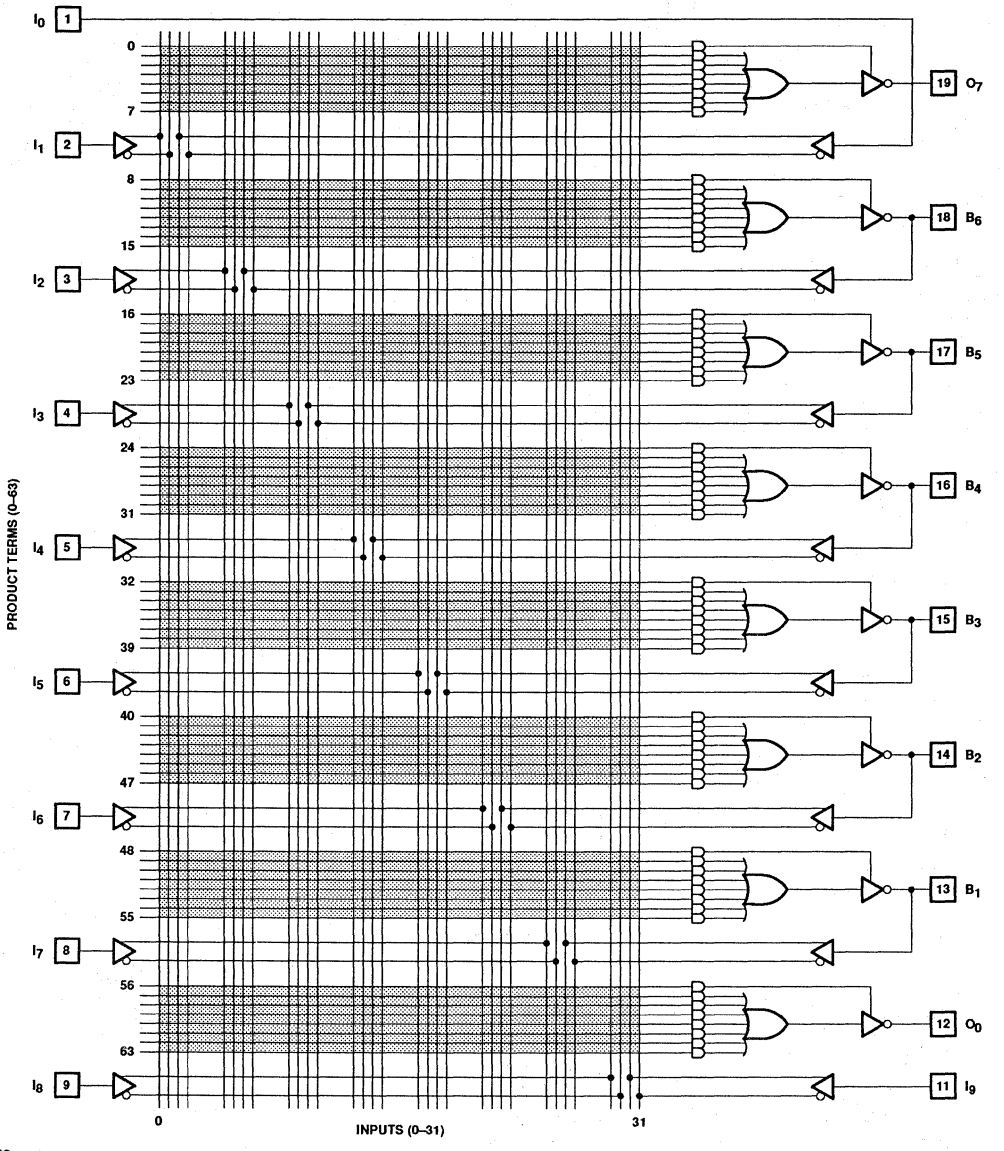
PLUS16R4



SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

LOGIC DIAGRAM

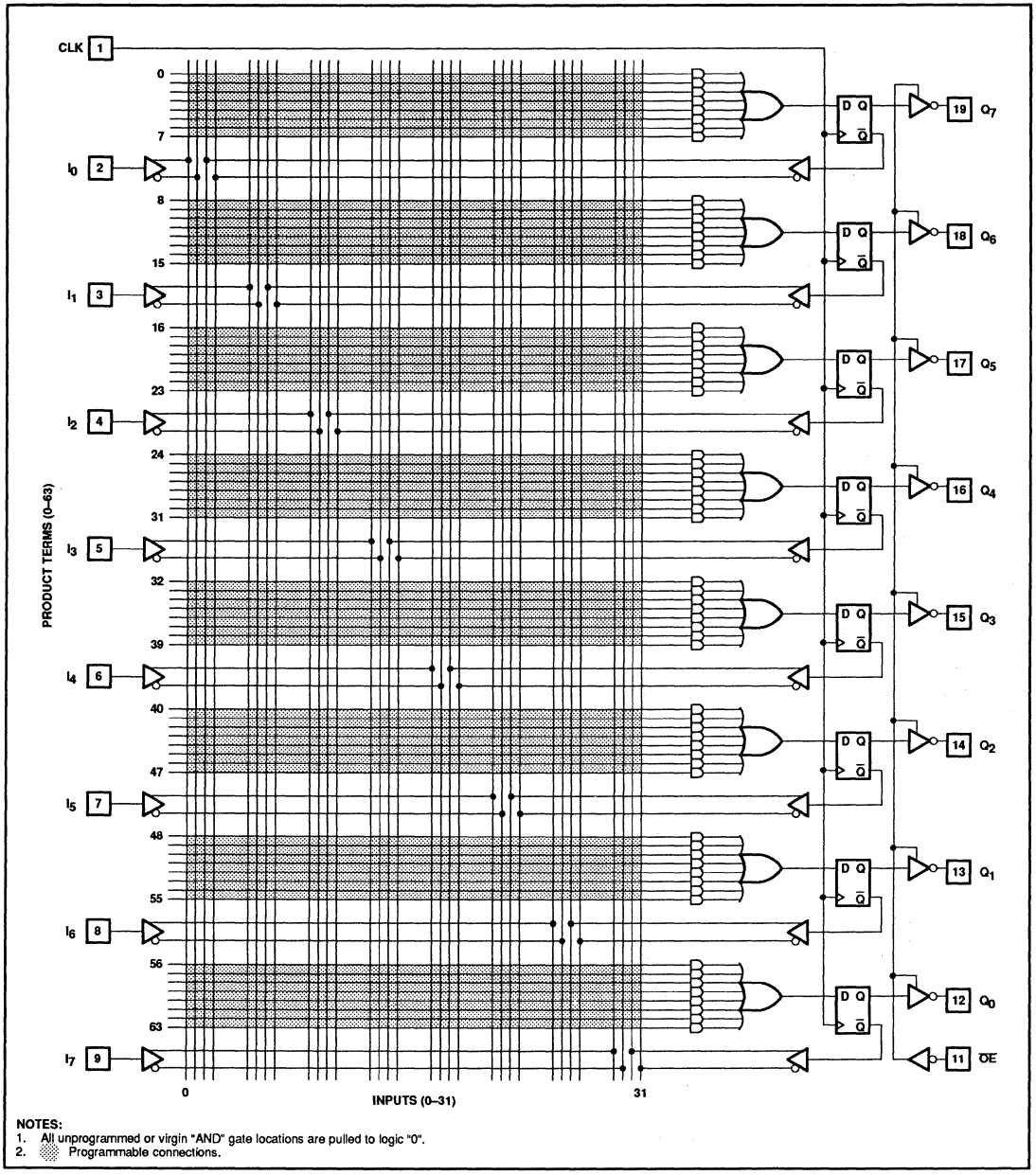
PLUS16L8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

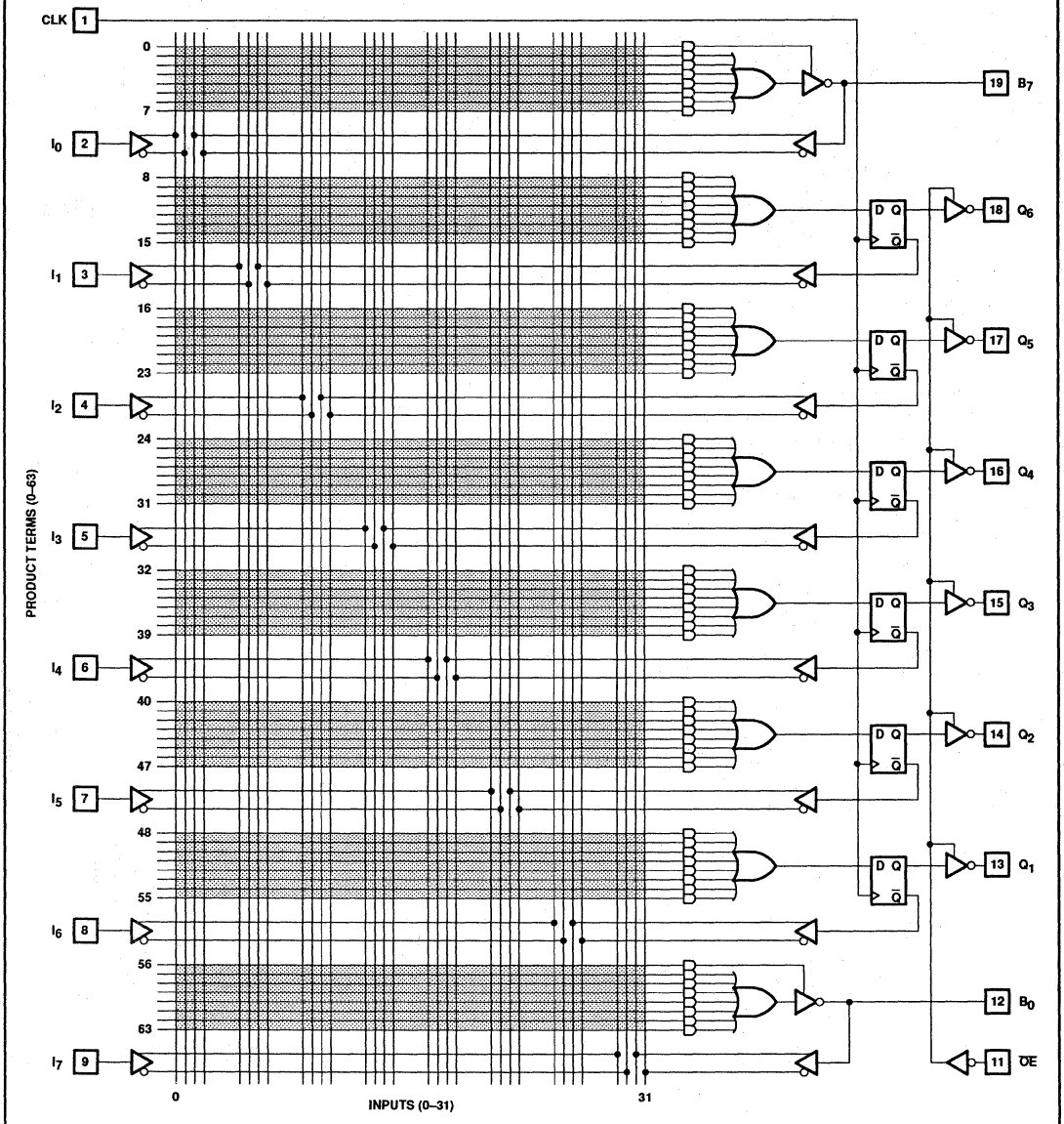
PLUS16R8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

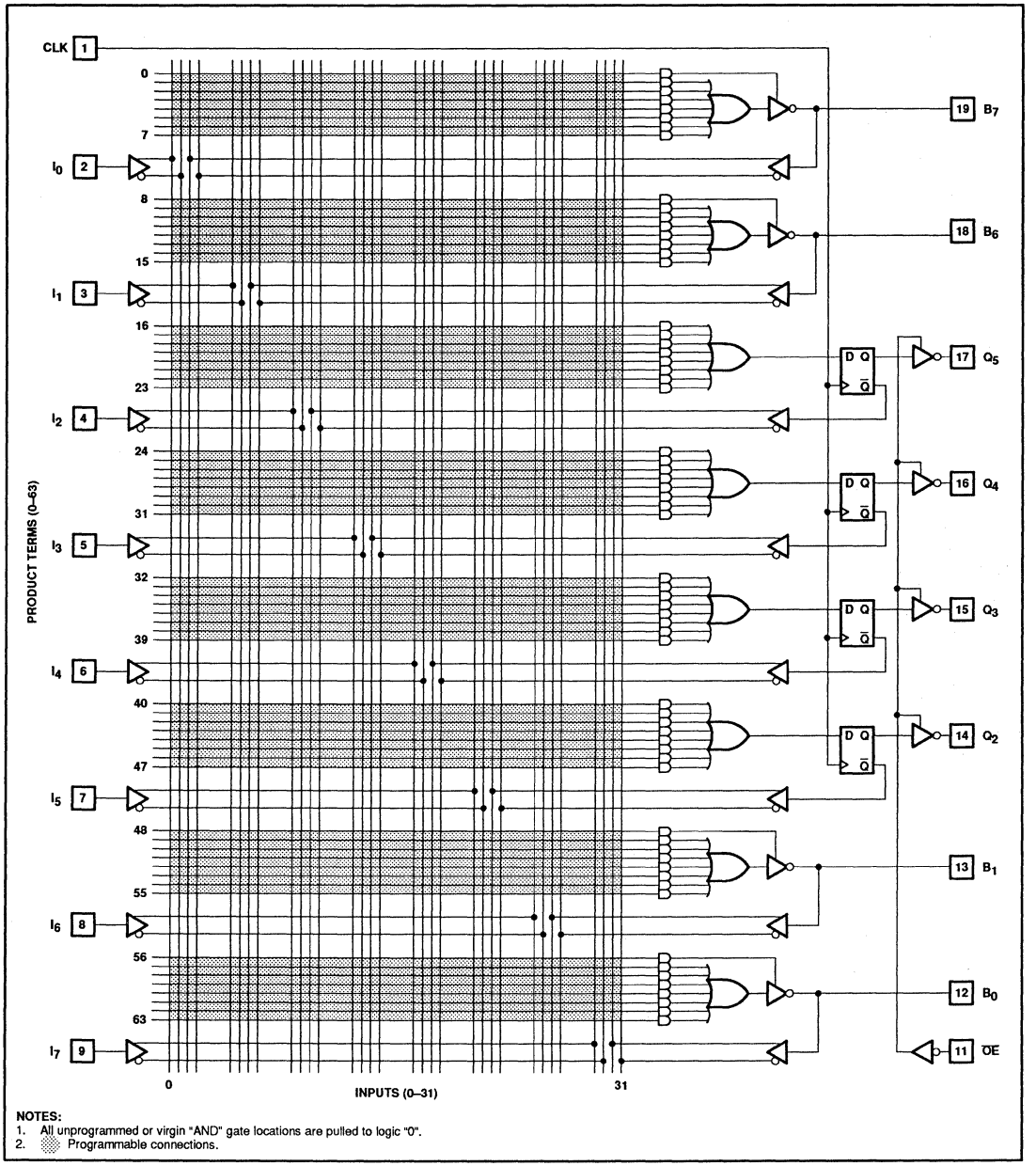
PLUS16R6



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

PLUS16R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

FUNCTIONAL DESCRIPTIONS

The PLUS16XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLUS16XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLUS16L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLUS16R8, PLUS16R6, PLUS16R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLUS16XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLUS16XX products feature variable Input/Output ratios. In addition to 8 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLUS16L8 provides 10 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLUS16R8 has 8 output registers, the 16R6 has 6, and the 16R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLUS16R8, R6, R4 enhance state machine design and initialization capability.

Software Support

Like other Programmable Logic Devices from Philips Semiconductors, the PLUS16XX series are supported by SLICE, the PC-based software development tool from Philips Semiconductors. The PLUS16XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

SLICE is available free of charge to qualified users.

Logic Programming

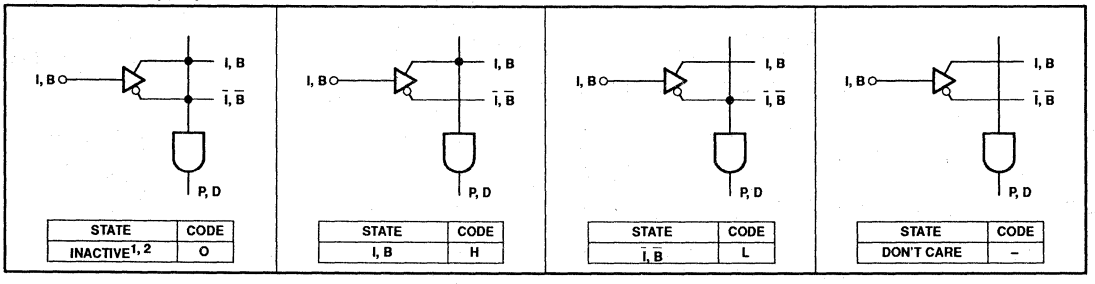
The PLUS16XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLUS16XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

Programming/Software Support

Ref to Section 9 *(Development Software)* and Section 10. *(Third-Party Programmer/ Software Support)* of the PLD data handbook for additional information.

AND ARRAY – (I, B)



STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B	H

STATE	CODE
$\overline{I, B}$	L

STATE	CODE
DON'T CARE	–

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–1.2	+8.0	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5V	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free–air temperature	0	+75	°C

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V			–250	µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = V_{CC} = V_{CCMAX}			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V			–100	µA
I_{OS}	Short circuit[4, 5]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		160	180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

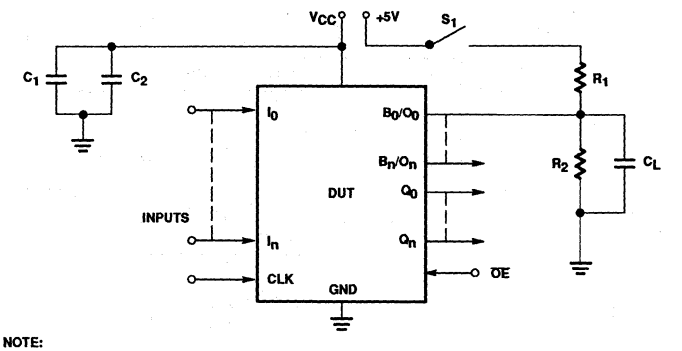
SYMBOL	PARAMETER	FROM	TO	LIMITS					UNIT
				-7			D		
				MIN[1]	TYP	MAX	MIN[1]	MAX	
Pulse Width									
t_{CKH}	Clock High	CK+	CK–	5			7		ns
t_{CKL}	Clock Low	CK–	CK+	5			7		ns
t_{CKP}	Period	CK+	CK+	10			14		ns
Setup & Hold time									
t_{IS}	Input	Input or feedback	CK+	7			9		ns
t_{IH}	Input	CK+	Input or feedback	0			0		ns
Propagation delay									
t_{CKO}	Clock	CK±	Q±	3		6.5	3	7.5	ns
t_{CKF}	Clock[3]	CK±	$\overline{Q}$			3		6.5	ns
t_{PD}	Output (16L8, R6, R4)[2]	I, B	Output	3		7.5	3	10	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable	3		8	3	10	ns
t_{OE2}	Output enable[4,5]	I	Output enable	3		10	3	10	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable	3		8	3	10	ns
t_{OD2}	Output disable[4,5]	I	Output disable	3		10	3	10	ns
t_{SKW}	Output	Q	Q			1		1	ns
t_{PPR}	Power-Up Reset	V_{CC}+	Q+			10		10	ns
Frequency (16R8, R6, R4)									
f_{MAX}	No feedback $1/(t_{CKL} + t_{CKH})$[6]				100		71.4		MHz
	Internal feedback $1/(t_{IS} + t_{CKF})$[6]				90		64.5		MHz
	External feedback $1/(t_{IS} + t_{CKO})$[6]				74		60.6		MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

NOTES:

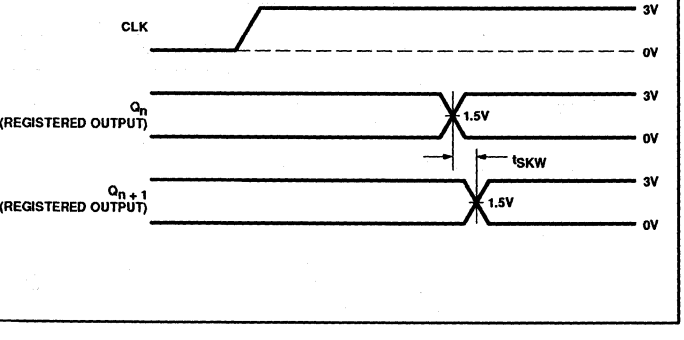
1. CL = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: CL = 50pF (with jig and scope capacitance), $V_{IH} = 3V$, $V_{IL} = 0V$, $V_{OH} = V_{OL} = 1.5V$.
3. t_{CKF} was calculated from measured Internal f_{MAX}.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

TEST LOAD CIRCUIT

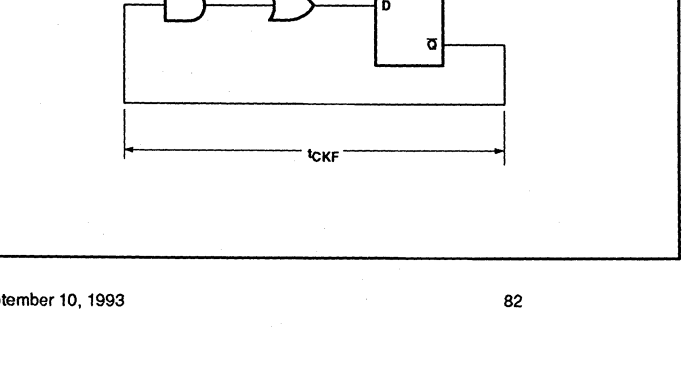


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

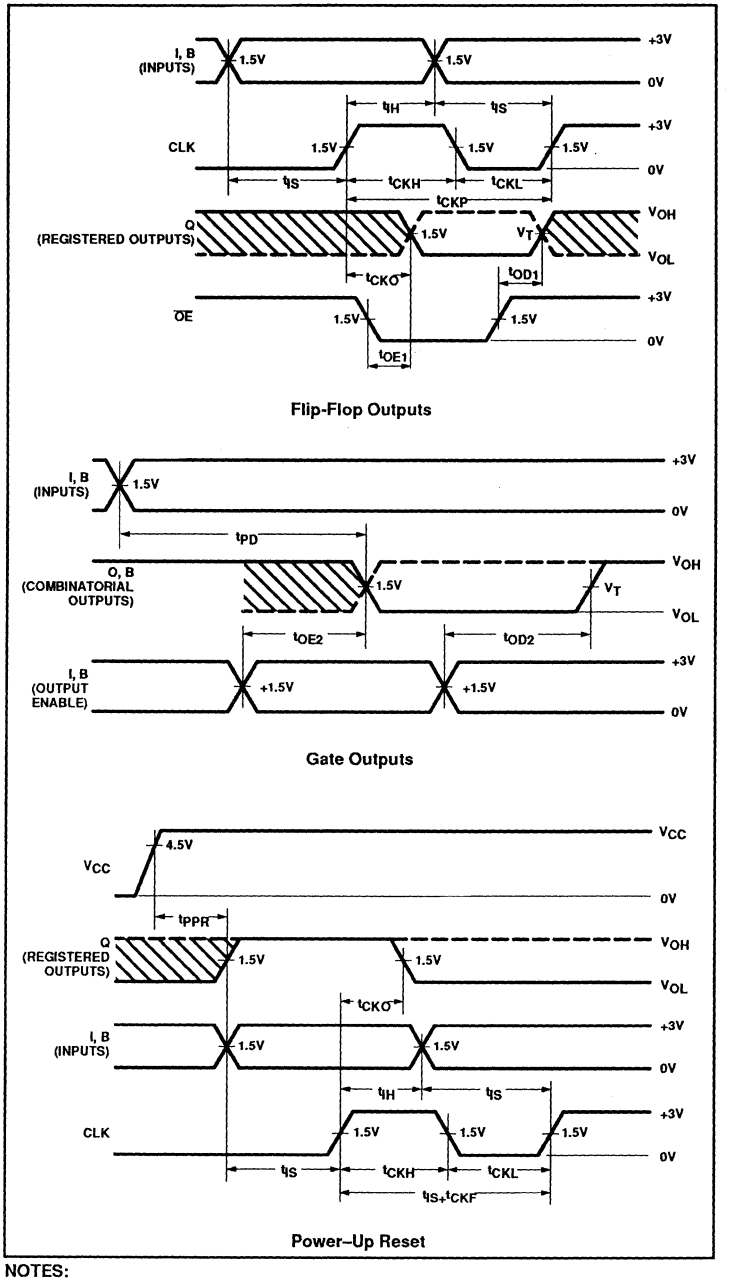
OUTPUT REGISTER SKEW



CLOCK TO FEEDBACK PATH



TIMING DIAGRAMS[1, 2]



TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.

FREQUENCY DEFINITIONS

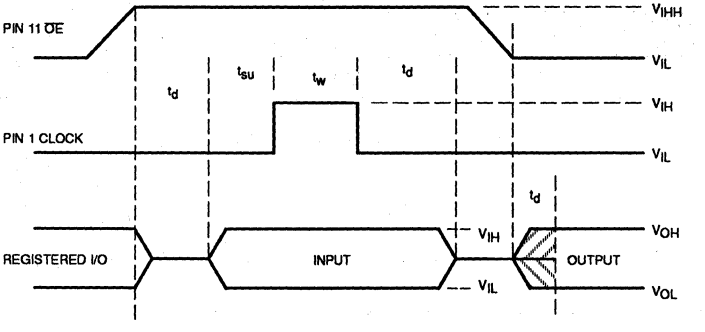
f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.5ns.

OUTPUT REGISTER PRELOAD

The output registers can be preloaded to any desired state during device testing. This permits any state to be tested without having to step through the entire state-machine sequence. Each register is preloaded individually by following the steps given below.

Step 1. With V_{CC} at 5V and Pin 1 at V_{IL}, raise Pin 11 to V_{IHH}.
Step 2. Apply either V_{IL} or V_{IH} to the output corresponding to the register to be preloaded.
Step 3. Pulse Pin 1, clocking in preload data.
Step 4. Remove output voltage, then lower Pin 11 to V_{IL}. Preload can be verified by observing the voltage level at the output pin.

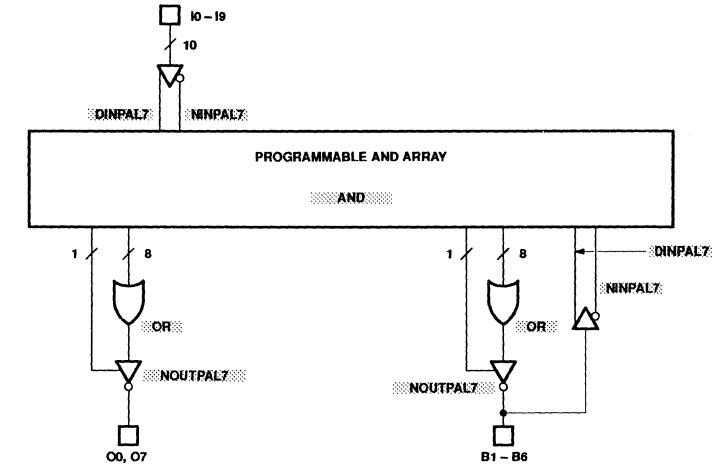


NOTE: $t_d = t_{su} = t_w = 100ns$ to 1000ns.
$V_{IHH} = 10.25V$ to 10.75V.
Pin number references for DIP package.

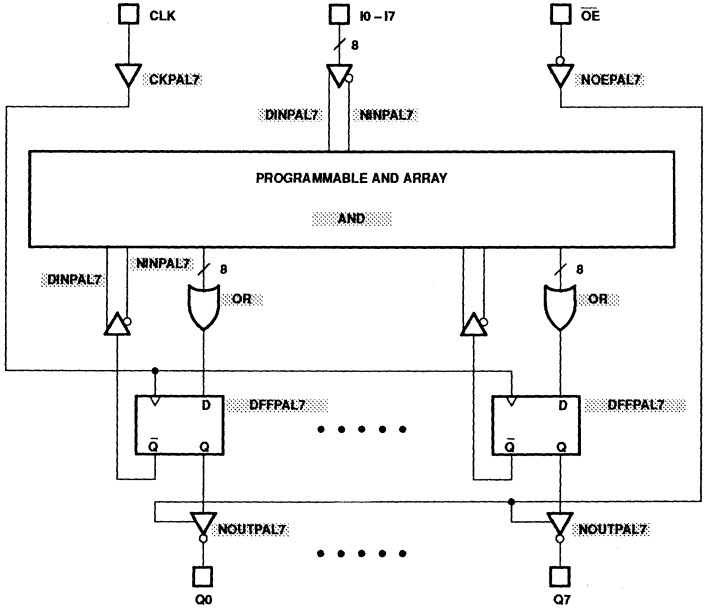
PROGRAMMING/SOFTWARE

Refer to Section 9 (*Development Software*) and Section 10 (*Third-Party Programmer/Software Support*) of this data handbook for additional information.

SNAP RESOURCE SUMMARY DESIGNATIONS

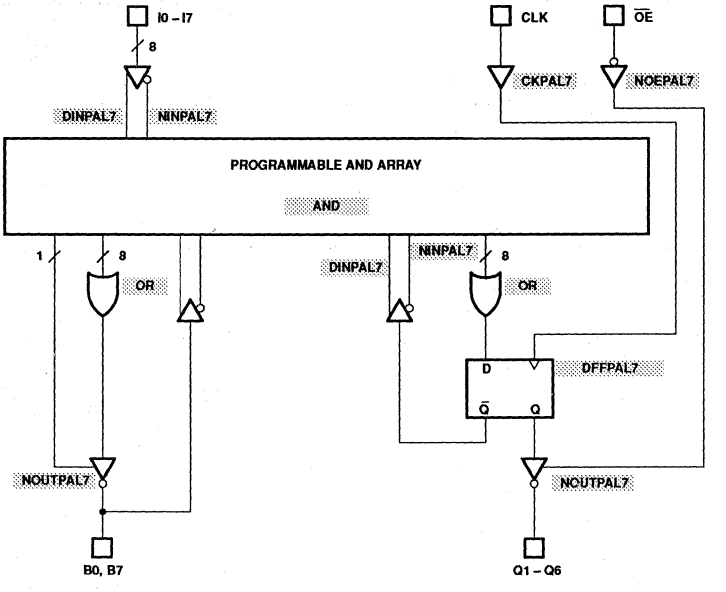


PLUS16L8

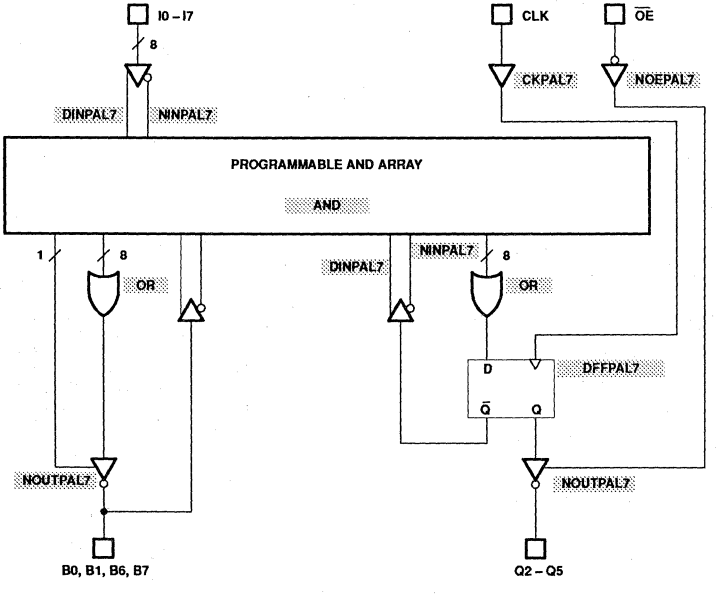


PLUS16R8

SNAP RESOURCE SUMMARY DESIGNATIONS (Continued)



PLUS16R6



PLUS16R4

PAL devices 20L8, 20R8, 20R6, 20R4

PLUS20R8D/-7 SERIES

FEATURES

- Ultra high-speed
  - t_{PD} = 7.5ns and f_{MAX} = 74MHz for the PLUS20R8-7 Series
  - t_{PD} = 10ns and f_{MAX} = 60 MHz for the PLUS20R8D Series
- 100% functionally and pin-for-pin compatible with industry standard 24-pin PAL® ICs
- Power-up reset function to enhance state machine design and testability
- Design support provided via SLICE and other CAD tools for Series 24 PAL devices
- Field-programmable on industry standard programmers
- Security fuse
- Individual 3-State control of all outputs

DESCRIPTION

The Philips Semiconductors PLUS20XX family consists of ultra high-speed 7.5ns and 10ns versions of Series 24 PAL devices.

The PLUS20XX family is 100% functional and pin-compatible with the 20L8, 20R8, 20R6, and 20R4 Series devices.

The sum of products (AND-OR) architecture is comprised of 64 AND gates and 8 fixed OR gates. Multiple bidirectional pins provide variable input/output pin ratios. Individual 3-State control of all outputs and registers with feedback (R8, R6, R4) is also provided. Proprietary designs can be protected by programming the security fuse.

The PLUS20R8, R6, and R4 have D-type flip-flops which are loaded on the Low-to-High transition of the clock input.

In order to facilitate state machine design and testing, a power-up reset function has been incorporated into these devices to reset all internal registers to active-Low after a specific period of time.

The Philips Semiconductors State-of-the-Art oxide isolation Bipolar fabrication process is employed to achieve high-performance operation.

The PLUS20XX family of devices are field programmable, enabling the user to quickly generate custom patterns using standard programming equipment. See the programmer chart for qualified programmers.

The SNAP software package from Philips Semiconductors supports easy design entry for the PLUS20XX series as well as other PLD devices from Philips Semiconductors. The PLUS20XX series are also supported by other standard CAD tools for PAL-type devices.

Order codes are listed in the Ordering Information table.

DEVICE NUMBER	DEDICATED INPUTS	COMBINATORIAL OUTPUTS	REGISTERED OUTPUTS
PLUS20L8	14	8 (6 I/O)	0
PLUS20R8	12	0	8
PLUS20R6	12	2 I/O	6
PLUS20R4	12	4 I/O	4

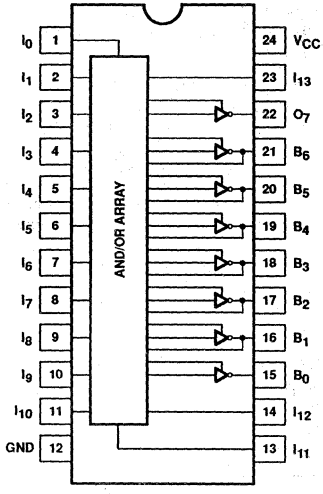
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin (300mils-wide) Plastic Dual-In-Line Package (DIP)	PLUS20R8DN PLUS20R6DN PLUS20R4DN PLUS20L8DN PLUS20R8–7N PLUS20R6–7N PLUS20R4–7N PLUS20L8–7N	0410D
28-Pin (300mils-wide) Plastic Leaded Chip Carrier (PLCC)	PLUS20R8DA PLUS20R6DA PLUS20R4DA PLUS20L8DA PLUS20R8–7A PLUS20R6–7A PLUS20R4–7A PLUS20L8–7A	0401F

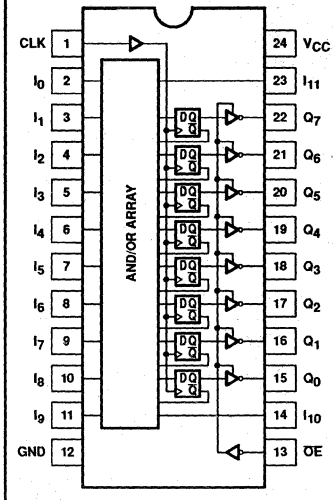
®PAL is a registered trademark of Advanced Micro Devices, Inc.

PIN CONFIGURATIONS

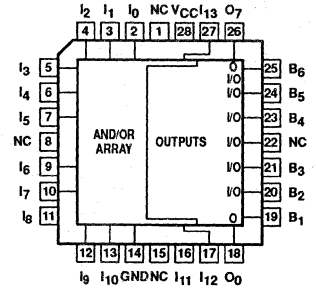
PLUS20L8



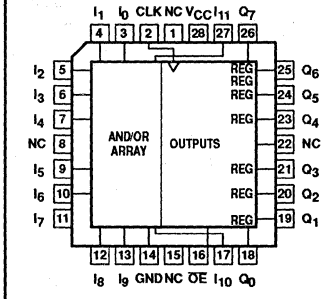
PLUS20R8



PLUS20L8



PLUS20R8

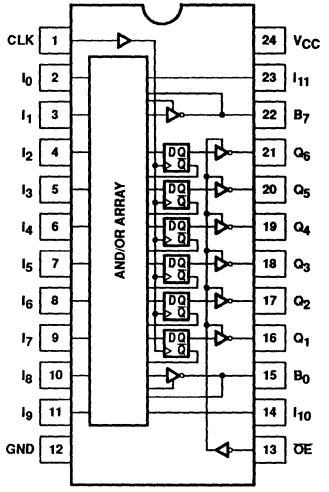


SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

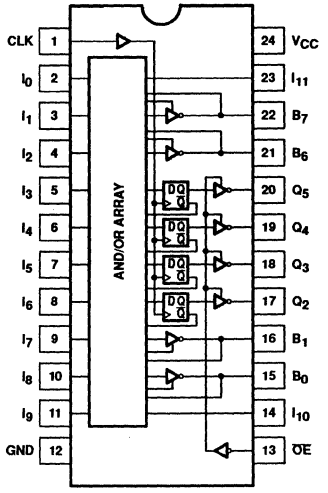
SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

PIN CONFIGURATIONS

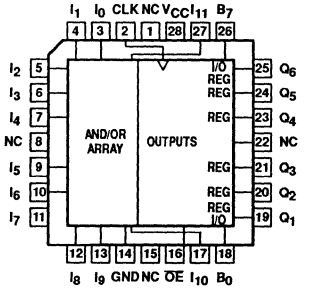
PLUS20R6



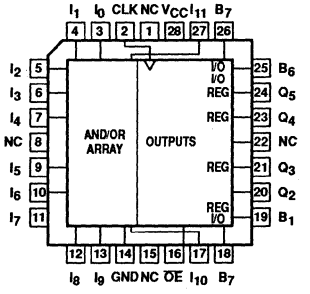
PLUS20R4



PLUS20R6



PLUS20R4

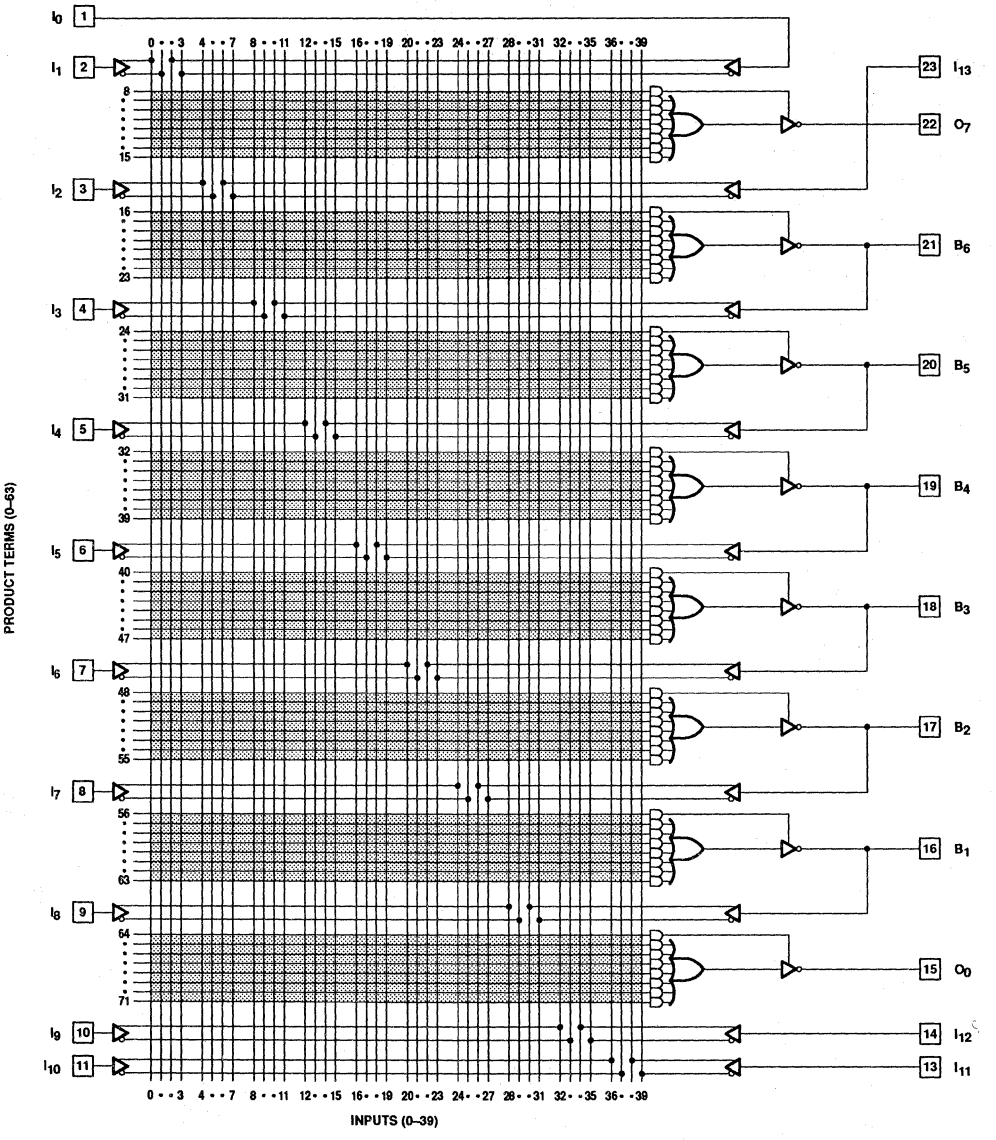


SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

SYMBOL	DESCRIPTION
I	Dedicated Input
O	Dedicated combinatorial Output
Q	Registered output
B	Bidirectional (input/output)
CLK	Clock input
OE	Output Enable
V_{CC}	Supply Voltage
GND	Ground
NC	No Connection

LOGIC DIAGRAM

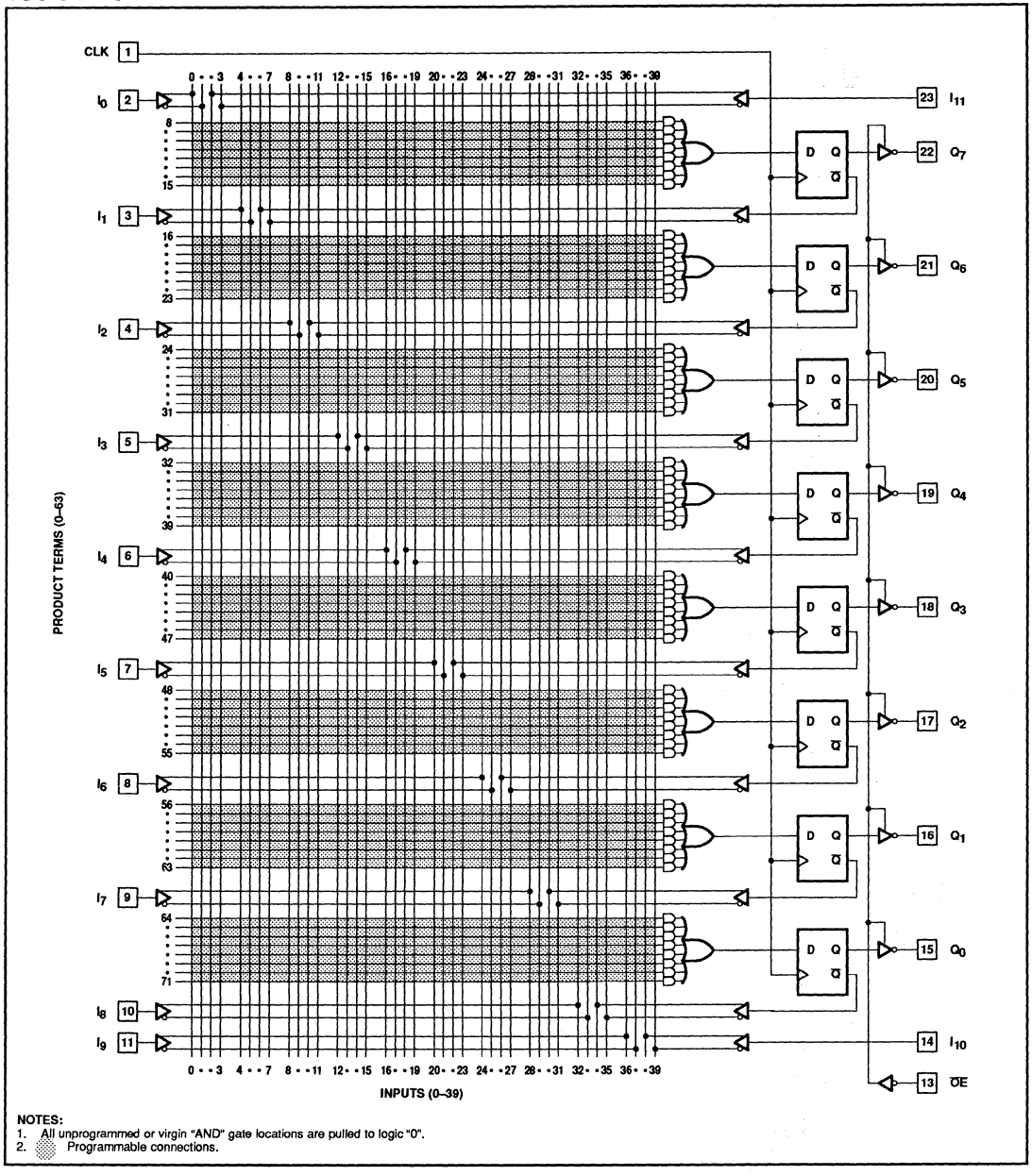
PLUS20L8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

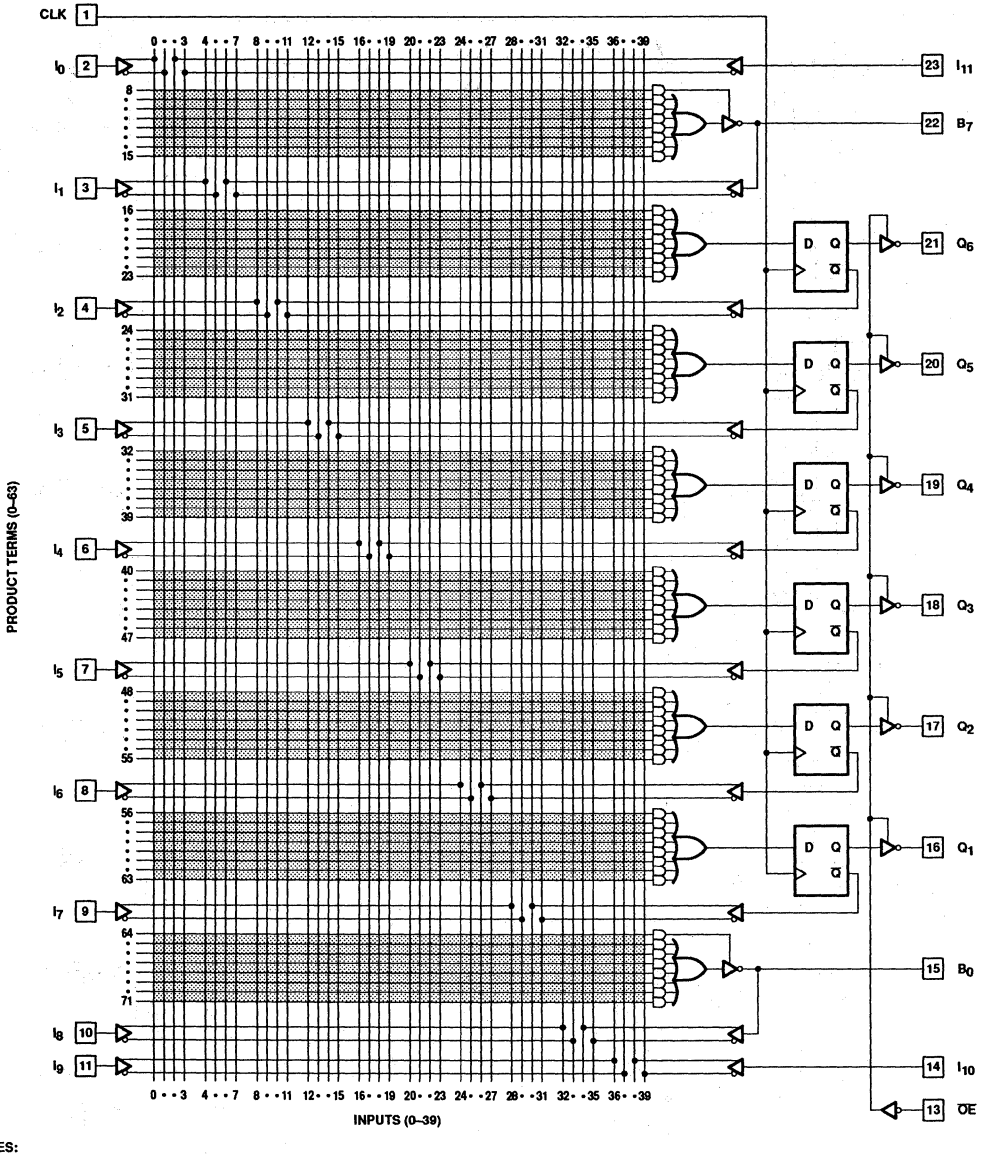
PLUS20R8



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

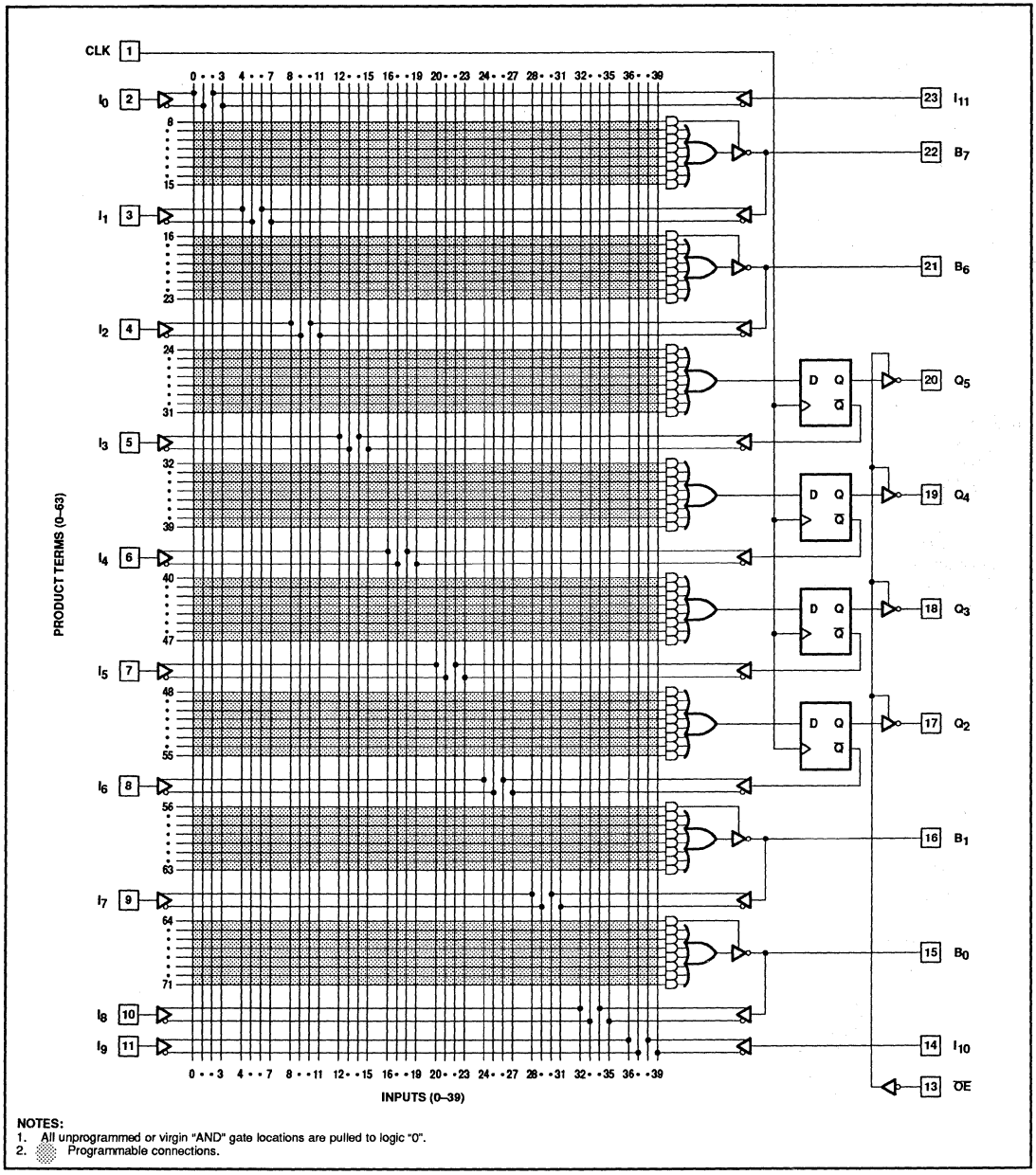
PLUS20R6



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

LOGIC DIAGRAM

PLUS20R4



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0".
2. ▒ Programmable connections.

FUNCTIONAL DESCRIPTIONS

The PLUS20XX series utilizes the familiar sum-of-products implementation consisting of a programmable AND array and a fixed OR array. These devices are capable of replacing an equivalent of four or more SSI/MSI integrated circuits to reduce package count and board area occupancy, consequently improving reliability and design cycle over Standard Cell or gate array options. By programming the security fuse, proprietary designs can be protected from duplication.

The PLUS20XX series consists of four PAL-type devices. Depending on the particular device type, there are a variable number of combinatorial and registered outputs available to the designer. The PLUS20L8 is a combinatorial part with 8 user configurable outputs (6 bidirectional), while the other three devices, PLUS20R8, PLUS20R6, PLUS20R4, have respectively 8, 6, and 4 output registers.

3-State Outputs

The PLUS20XX series devices also feature 3-State output buffers on each output pin which can be programmed for individual control of all outputs. The registered outputs (Qn) are controlled by an external input (/OE), and the combinatorial outputs (On, Bn) use a product term to control the enable function.

Programmable Bidirectional Pins

The PLUS20XX products feature variable Input/Output ratios. In addition to 12 dedicated inputs, each combinatorial output pin of the registered devices can be individually programmed as an input or output. The PLUS20L8 provides 14 dedicated inputs and 6 Bidirectional I/O lines that can be individually configured as inputs or outputs.

Output Registers

The PLUS20R8 has 8 output registers, the 20R6 has 6, and the 20R4 has 4. Each output register is a D-type flip-flop which is loaded on the Low-to-High transition of the clock input. These output registers are capable of feeding the outputs of the registers back into the array to facilitate design of synchronous state machines.

Power-up Reset

By resetting all flip-flops to a logic Low, as the power is turned on, the PLUS20R8, R6, R4 enhance state machine design and initialization capability.

Software Support

Like other Programmable Logic Devices from Philips Semiconductors, the PLUS20XX series are supported by SLICE, the PC-based software development tool from Philips Semiconductors. The PLUS20XX family of devices are also supported by standard CAD tools for PAL devices, including ABEL and CUPL.

SLICE is available free of charge to qualified users.

Logic Programming

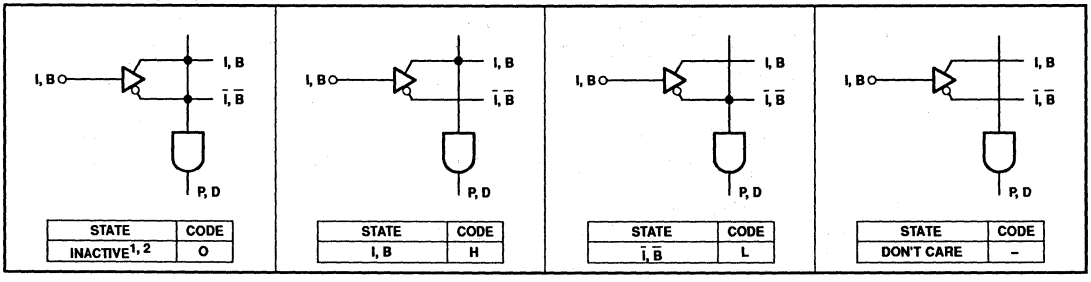
The PLUS20XX series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLUS20XX architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of the PLD data handbook for additional information.

AND ARRAY – (I, B)



VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–1.2	+8.0	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5V	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free–air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = 24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low[3]	V_{IN} = 0.40V			–250	µA
I_{IH}	High[3]	V_{IN} = 2.7V			25	µA
I_I	Maximum input current	V_{IN} = V_{CC} = V_{CCMAX}			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage	V_{OUT} = 2.7V			100	µA
I_{OZL}	Output leakage	V_{OUT} = 0.4V			–100	µA
I_{OS}	Short circuit [4, 5]	V_{OUT} = 0V	–30		–90	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		150	210	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5V				
		V_{OUT} = 2.0V		8		pF
C_B	I/O (B)	V_{OUT} = 2V, f = 1MHz		8		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZH}.
4. Test one at a time.
5. Duration of short circuit should not exceed 1 second.
6. These parameters are not 100% tested but periodically sampled.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 200\Omega$, $R_2 = 390\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS					UNIT
				-7			D		
				MIN[1]	TYP	MAX	MIN[1]	MAX	
Pulse Width									
t_{CKH}	Clock High	CK+	CK–	5			7		ns
t_{CKL}	Clock Low	CK–	CK+	5			7		ns
t_{CKP}	Period	CK+	CK+	10			14		ns
Setup & Hold time									
t_{IS}	Input	Input or feedback	CK+	7			9		ns
t_{IH}	Input	CK+	Input or feedback	0			0		ns
Propagation delay									
t_{CKO}	Clock	CK±	Q±	3		6.5	3	7.5	ns
t_{CKF}	Clock[3]	CK±	$\overline{Q}$			3		6.5	ns
t_{PD}	Output (20L8, R6, R4)[2]	I, B	Output	3		7.5	3	10	ns
t_{OE1}	Output enable[4]	$\overline{OE}$	Output enable	3		8	3	10	ns
t_{OE2}	Output enable[4,5]	I	Output enable	3		10	3	10	ns
t_{OD1}	Output disable[4]	$\overline{OE}$	Output disable	3		8	3	10	ns
t_{OD2}	Output disable[4,5]	I	Output disable	3		10	3	10	ns
t_{SKW}	Output	Q	Q			1		1	ns
t_{PPR}	Power-Up Reset	V_{CC}+	Q+			10		10	ns
Frequency (20R8, R6, R4)									
f_{MAX}	No feedback $1/(t_{CKL} + t_{CKH})$[6]				100		71.4		MHz
	Internal feedback $1/(t_{IS} + t_{CKF})$[6]				90		64.5		MHz
	External feedback $1/(t_{IS} + t_{CKO})$[6]				74		60.6		MHz

* For definitions of the terms, please refer to the Timing/Frequency Definitions tables.

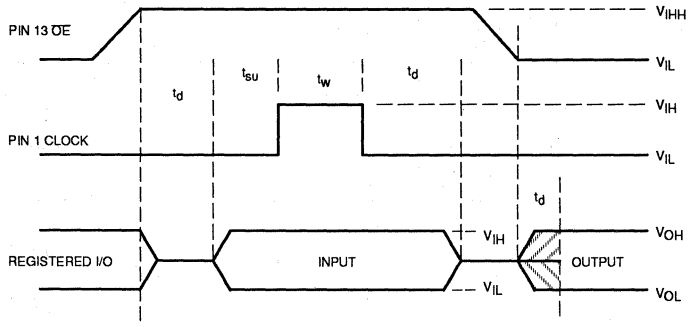
NOTES:

1. CL = 0pF while measuring minimum output delays.
2. t_{PD} test conditions: CL = 50pF (with jig and scope capacitance), $V_{IH} = 3V$, $V_{IL} = 0V$, $V_{OH} = V_{OL} = 1.5V$.
3. t_{CKF} was calculated from measured Internal f_{MAX}.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
5. Same function as t_{OE1} and t_{OD1}, with the difference of using product term control.
6. Not 100% tested, but calculated at initial characterization and at any time a modification in design takes place which may affect the frequency.

OUTPUT REGISTER PRELOAD

The output registers can be preloaded to any desired state during device testing. This permits any state to be tested without having to step through the entire state-machine sequence. Each register is preloaded individually by following the steps given below.

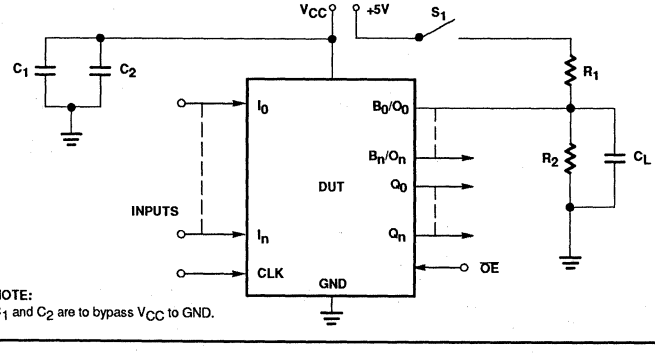
Step 1. With V_{CC} at 5V and Pin 1 at V_{IL}, raise Pin 13 to V_{IHH}.
Step 2. Apply either V_{IL} or V_{IH} to the output corresponding to the register to be preloaded.
Step 3. Pulse Pin 1, clocking in preload data.
Step 4. Remove output voltage, then lower Pin 13 to V_{IL}. Preload can be verified by observing the voltage level at the output pin.



NOTE: $t_d = t_{su} = t_w$ = 100ns to 1000ns.
V_{IHH} = 10.25V to 10.75V.

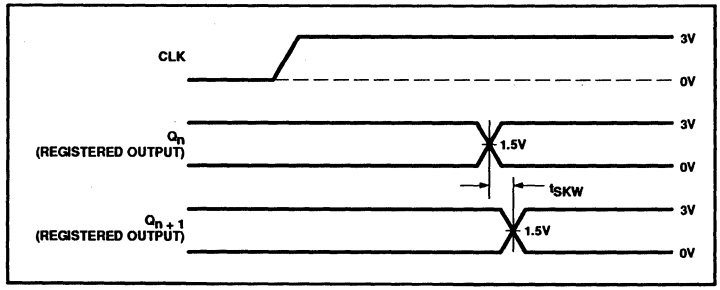
Pin and number reference for DIP package

TEST LOAD CIRCUIT

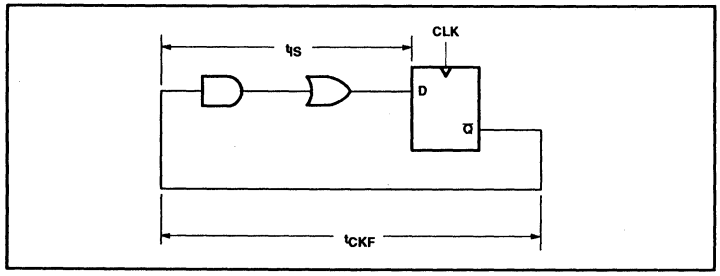


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

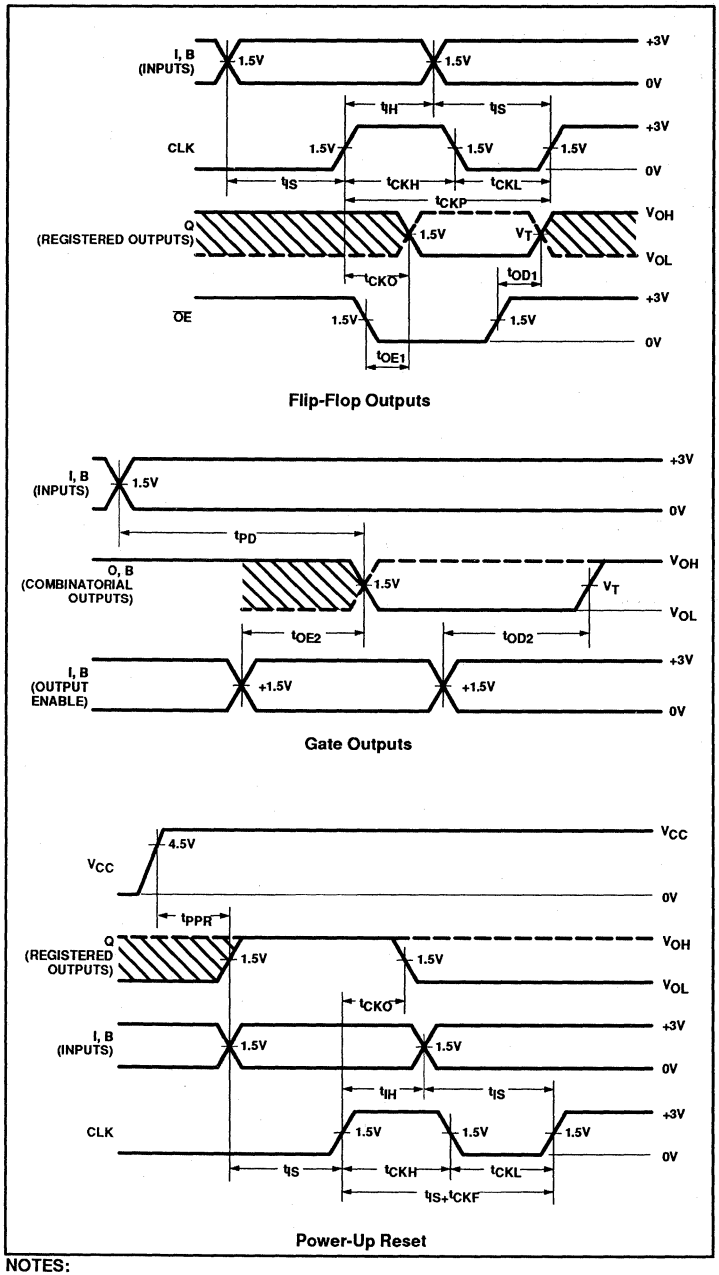
OUTPUT REGISTER SKEW



CLOCK TO FEEDBACK PATH



TIMING DIAGRAMS[1, 2]



TIMING DEFINITIONS

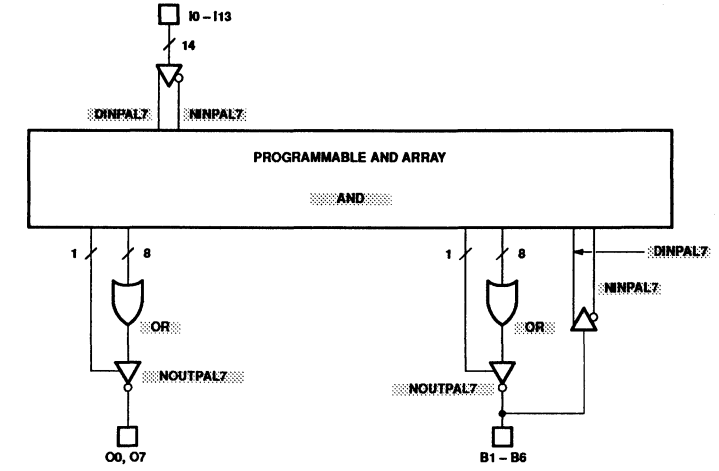
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{IS}	Required delay between beginning of valid input and positive transition of clock.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKF}	Delay between positive transition of clock and when internal $\overline{Q}$ output of flip-flop becomes valid.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the Off-State.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the Off-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.

FREQUENCY DEFINITIONS

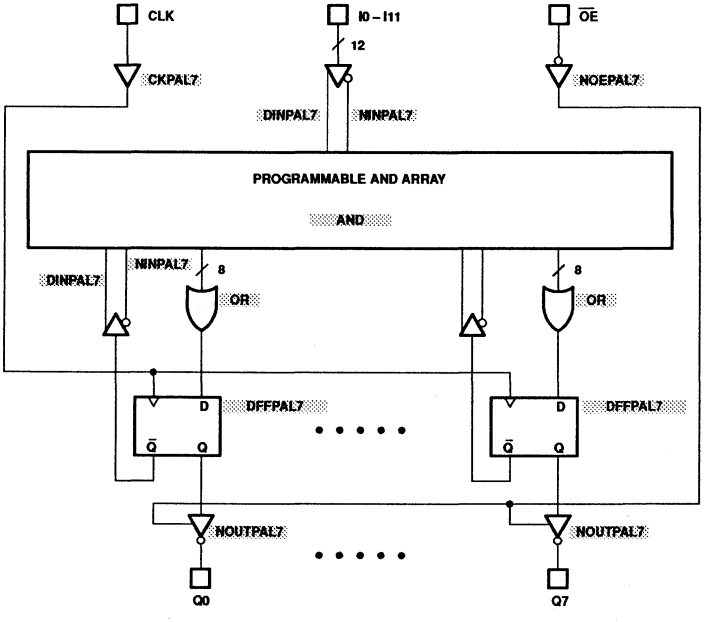
f_{MAX}	**No feedback:** Determined by the minimum clock period, $1/(t_{CKL} + t_{CKH})$. **Internal feedback:** Determined by the internal delay from flip-flop outputs through the internal feedback and array to the flip-flop inputs, $1/(t_{IS} + t_{CKF})$. **External feedback:** Determined by clock-to-output delay and input setup time, $1/(t_{IS} + t_{CKO})$.

NOTES:
1. Input pulse amplitude is 0V to 3V.
2. Input rise and fall times are 2.5ns.

SNAP RESOURCE SUMMARY DESIGNATIONS

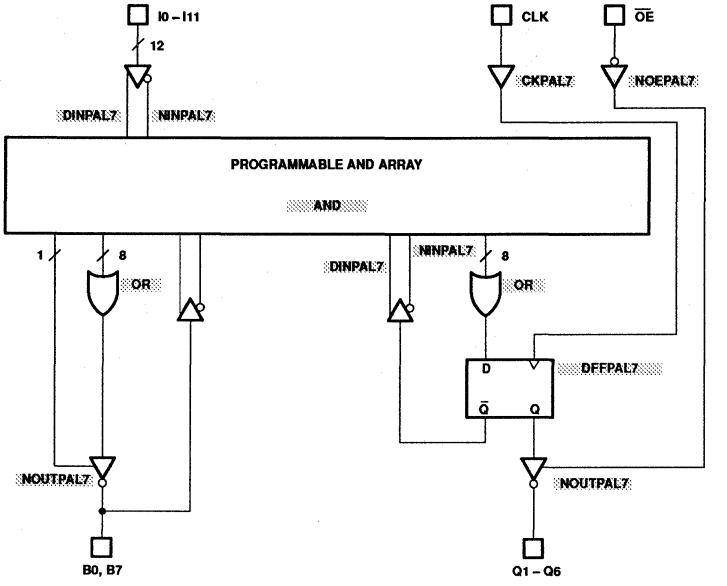


PLUS20L8

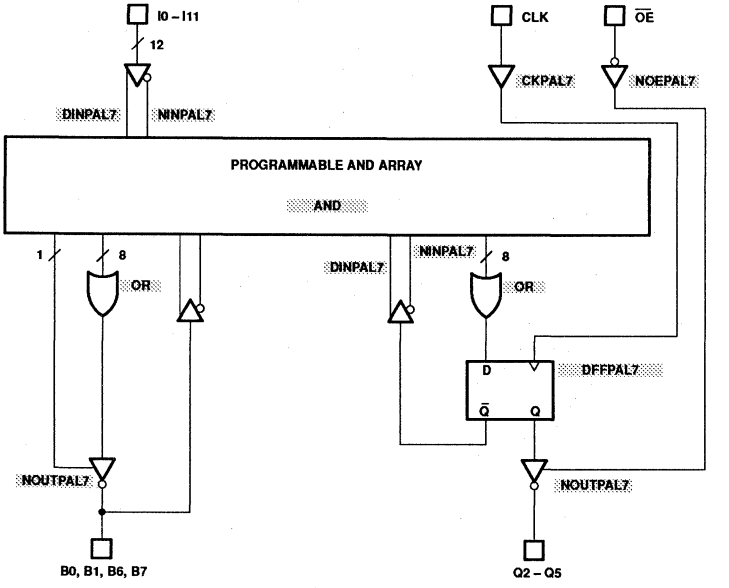


PLUS20R8

SNAP RESOURCE SUMMARY DESIGNATIONS (Continued)



PLUS20R6



PLUS20R4

CMOS programmable electrically erasable logic device

PL22V10-10

FEATURES

- Advanced CMOS EEPROM technology
- Ultra high performance
  - 10ns, (t_{PD}) commercial version
  - f_{MAX} as fast as 83.3MHz
- Available in Dual In-Line, Small Outline Large, and Plastic Leaded Chip Carrier packages
- Low power consumption
  - 110mA + 0.5mA/MHz max
- EE reprogrammability
  - Low-risk reprogrammable inventory
  - Superior programming and functional yield
  - 100% testable
  - Erases and programs in seconds
  - 100 guaranteed erase cycles
- Development and programming support
  - Third-party software and programmers
  - SLICE development software
- Architectural flexibility
  - 132 product term × 44 input AND array
  - Up to 22 inputs and 10 outputs
  - Variable product term distribution (8 to 16 per output) for greater logic flexibility
  - Independently programmable 4-configuration I/O macrocells
  - Synchronous preset, asynchronous clear
  - Independently programmable output enables
- Application versatility
  - Pin-for-pin and JEDEC-file compatible with the bipolar AmPAL22V10, CMOS PALC22V10 and PEEL22CV10A

DESCRIPTION

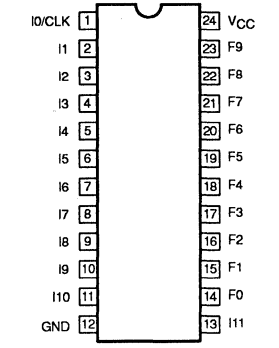
The Philips Semiconductors PL22V10-10 is a CMOS programmable electrically erasable logic device that provides a high-performance, low-power, reprogrammable, and architecturally enhanced alternative to early generation programmable logic devices (PLDs). Designed in advanced CMOS EEPROM technology, the PL22V10 rivals speed parameters of comparable bipolar PLDs while providing a dramatic improvement in active power consumption. The EE reprogrammability of the PL22V10 allows cost effective plastic packaging, low risk inventory, reduced development and retrofit costs, and enhanced testability to ensure 100% field programmability and function. The PL22V10's flexible architecture offers complete function and JEDEC-file compatibility with the bipolar AmPAL22V10 and the CMOS PALC22V10. Applications for the PL22V10 include: replacement of random SSI/MSI logic circuitry and user customized sequential and combinatorial functions such as counters, shift registers, state machines, address decoders, multiplexers, etc. Development and programming support for the PL22V10 is provided by Philips Semiconductors and third-party manufacturers.

PIN LABEL DESCRIPTIONS

I1 – I11	Dedicated Input
NC	Not Connected
F0 – F9	Macro Cell Input/Output
CLK/I0	Clock Input/Dedicated Input
V_{CC}	Supply Voltage
GND	Ground

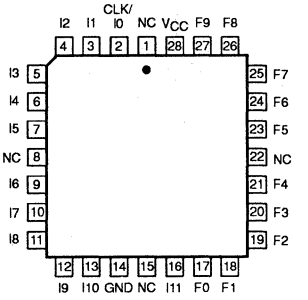
PIN CONFIGURATIONS

D and N Packages



N = Plastic Dual In-Line Package (300mil-wide)
D = Plastic Small Outline Large (300mil-wide) Package

A Package



A = Plastic Leaded Chip Carrier

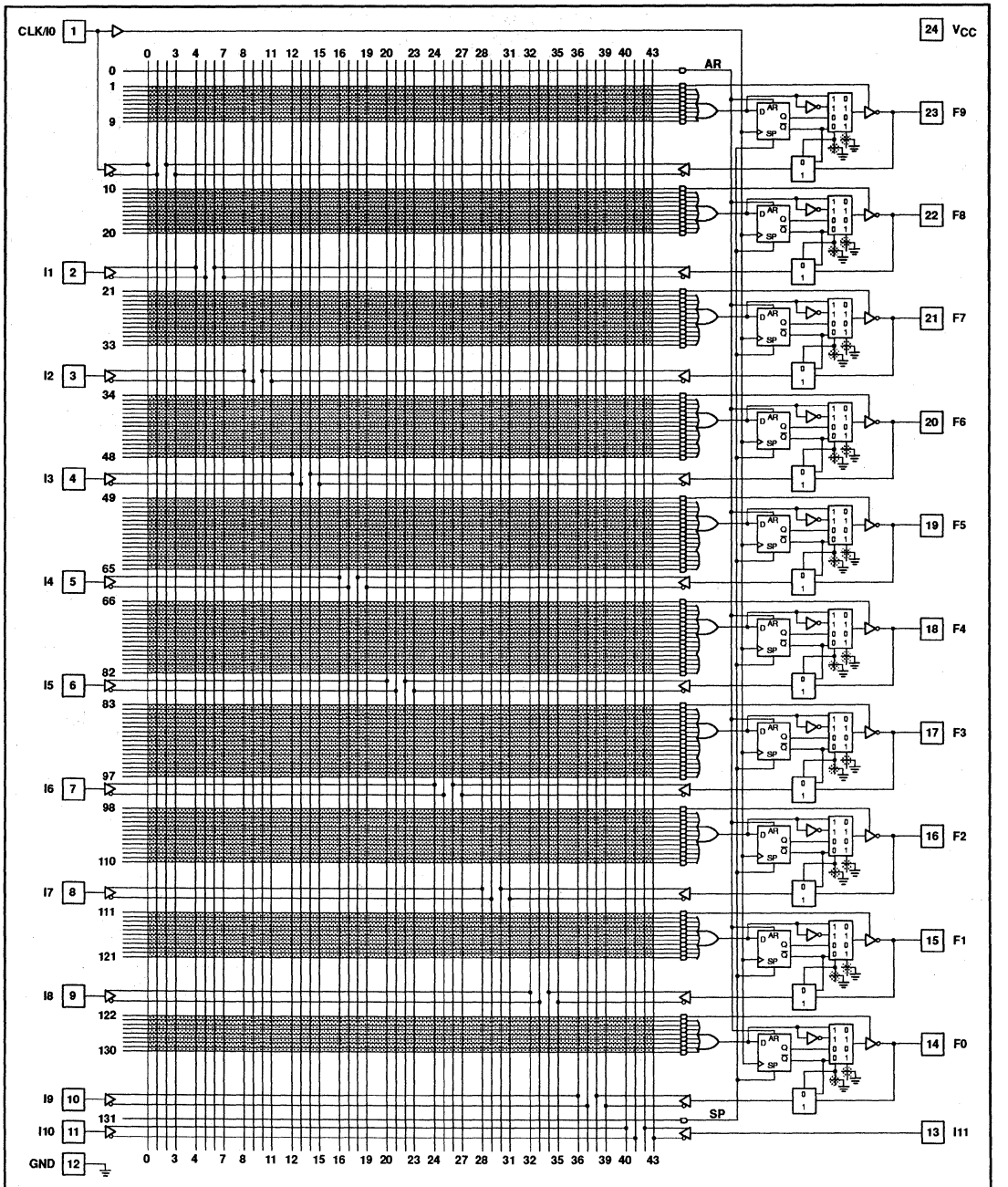
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin (300mil-wide) Plastic DIP (Dual-In-Line Package)	PL22V10–10N	0410D
28-Pin (300mil-wide) PLCC (Plastic Leaded Chip Carrier Package)	PL22V10–10A	0401F
24-Pin (300mil-wide) Plastic SOL (Small Outline Large) Package	PL22V10–10D	0173D

CMOS programmable electrically erasable logic device

PL22V10-10

LOGIC DIAGRAM



NOTE:
▒ Programmable connection.

CMOS programmable electrically erasable logic device

PL22V10-10

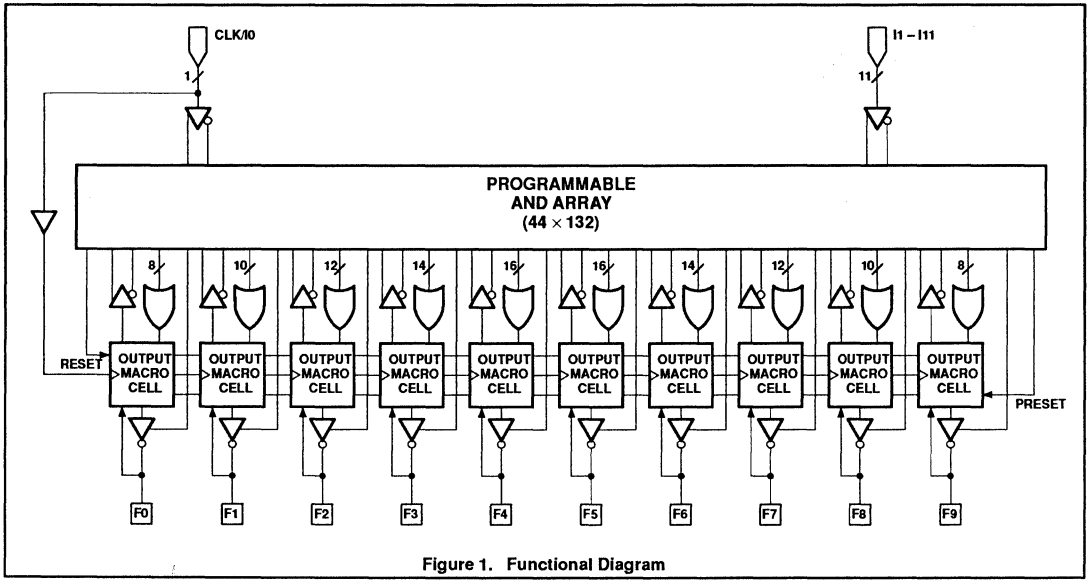


Figure 1. Functional Diagram

FUNCTION DESCRIPTION

The PL22V10 implements logic functions as sum-of-products expressions in a programmable-AND/fixed-OR logic array. User-defined functions are created by programming the connections of input signals into the array. User-configurable output structures in the form of I/O macrocells further increase logic flexibility.

ARCHITECTURE OVERVIEW

The PL22V10 architecture is illustrated in the Figure 1. Twelve dedicated inputs and 10 I/Os provide up to 22 inputs and 10 outputs for creation of logic functions. At the core of the device is a programmable electrically-erasable AND array which drives a fixed OR array. With this structure, the PL22V10 can implement up to 10 sum-of-products logic expressions.

Associated with each of the 10 OR functions is an I/O macro cell which can be independently programmed to one of 4 different configurations. The programmable macro cells allow each I/O to create sequential or combinatorial logic functions with either Active-High or Active-Low polarity.

AND/OR LOGIC ARRAY

The programmable AND array of the PL22V10 (shown in the Logic Diagram) is formed by input lines intersecting product terms. The input lines and product terms are used as follows:

44 input lines:

- 24 input lines carry the True and Complement of the signals applied to the 12 input pins
- 20 additional lines carry the True and Complement values of feedback or input signals from the 10 I/Os

132 product terms:

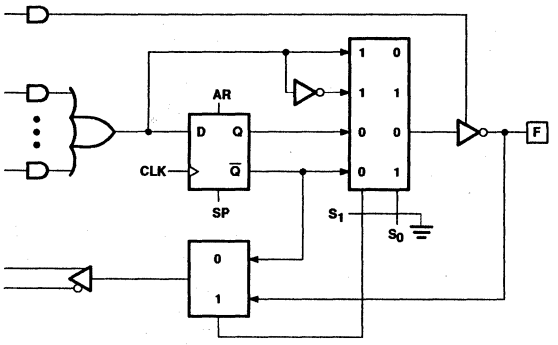
- 120 product terms (arranged in 2 groups of 8, 10, 12, 14, and 16) used to form logical sums
- 10 output enable terms (one for each I/O)
- 1 global synchronous preset term
- 1 global asynchronous clear term

At each input-line/product-term intersection there is an EEPROM memory cell which determines whether or not there is a logical connection at that intersection. Each product term is essentially a 44-input AND gate. A product term which is connected to both the True and Complement of an input signal will always be FALSE, and thus will not effect the OR function that it drives. When all the connections on a product term are opened, a Don't Care state exists and that term will always be TRUE.

When programming the PL22V10, the device programmer first performs a bulk erase to instantly remove the previous pattern. The erase cycle opens every logical connection in the array. The device is then configured to perform the user-defined function by programming selected connections in the AND array. (Note that EEPROM device programmers automatically program the connections on unused product terms so that they will have no effect on the output function.)

VARIABLE PRODUCT TERM DISTRIBUTION

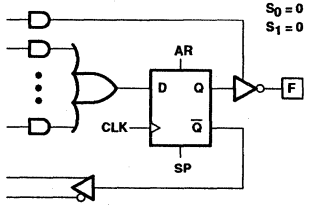
The PL22V10 provides 120 product terms to drive the 10 OR functions. These product terms are distributed among the outputs in groups of 8, 10, 12, 14, and 16 to form logical sums (see Logic Diagram). This distribution allows optimum use of device resources.



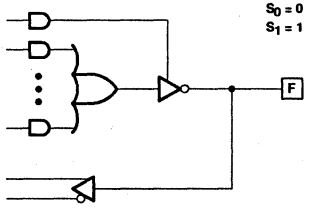
S_1	S_0	OUTPUT CONFIGURATION
0	0	Registered/Active-LOW
0	1	Registered/Active-HIGH
1	0	Combinatorial/Active-LOW
1	1	Combinatorial/Active-HIGH

0 = Unprogrammed fuse
1 = Programmed fuse

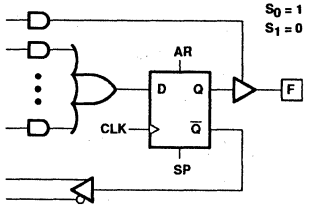
Figure 2. Output Macro Cell Logic Diagram



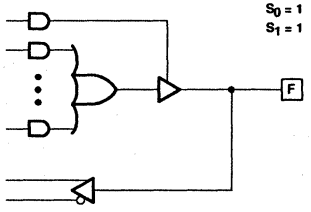
a. Registered/Active-LOW



c. Combinatorial/Active-LOW



b. Registered/Active-HIGH



d. Combinatorial/Active-HIGH

Figure 3.

CMOS programmable electrically erasable logic device

PL22V10-10

PROGRAMMABLE I/O MACROCELL

The output macrocell provides complete control over the architecture of each output. The ability to configure each output independently permits users to tailor the configuration of the PL22V10 to the precise requirements of their designs.

MACROCELL ARCHITECTURE

Each I/O macrocell, as shown in Figure 2, consists of a D-type flip-flop and two signal-select multiplexers. The configuration of each macrocell of the PL22V10 is determined by the two EEPROM bits controlling these multiplexers. These bits determine output polarity, and output type (registered or non-registered). Equivalent circuits for the macrocell configurations are illustrated in Figure 3.

OUTPUT TYPE

The signal from the OR array can be fed directly to the output pin (combinatorial function) or latched in the D-type flip-flop (registered function). The D-type flip-flop latches data on the rising edge of the clock and is controlled by the global preset and clear terms. When the synchronous preset term is satisfied, the Q output of the register will be set HIGH at the next rising edge of the clock input. Satisfying the asynchronous clear term will set Q LOW, regardless of the clock state. If both terms are satisfied simultaneously, the clear will override the preset.

PROGRAM/ERASE CYCLES

The PL22V10 is 100% testable, erases/programs in seconds, and has 100 guaranteed erase cycles.

OUTPUT POLARITY

Each macrocell can be configured to implement Active-High or Active-Low logic. Programmable polarity eliminates the need for external inverters.

OUTPUT ENABLE

The output of each I/O macrocell can be enabled or disabled under the control of its associated programmable output enable product term. When the logical conditions programmed on the output enable term are satisfied, the output signal is propagated to the I/O pin. Otherwise, the output buffer is driven into the high-impedance state.

Under the control of the output enable term, the I/O pin can function as a dedicated input, a dedicated output, or a bi-directional I/O. Opening every connection on the output enable term will permanently enable the output buffer and yield a dedicated output. Conversely, if every connection is intact, the enable term will always be logically FALSE and the I/O will function as a dedicated input.

REGISTER FEEDBACK SELECT

When the I/O macrocell is configured to implement a registered function (S1=0) (Figures 3.a or 3.b), the feedback signal to the AND array is taken from the $\overline{Q}$ output.

BI-DIRECTIONAL I/O SELECT

When configuring an I/O macrocell to implement a combinatorial function (S1=1) (Figures 3.c or 3.d), the feedback signal is taken from the I/O pin. In this case, the pin can be used as a dedicated input, a dedicated output, or a bi-directional I/O.

POWER-ON RESET

To ease system initialization, all flip-flops will power-up to a reset condition and the Q output will be low. The actual output of the PL22V10 will depend on the programmed output polarity. The V_{CC} rise must be monotonic and the reset delay time is 5µs maximum.

DESIGN SECURITY

The PL22V10 provides a special EEPROM security bit that prevents unauthorized reading or copying of designs programmed into the device. The security bit is set by the PLD programmer, either at the conclusion of the programming cycle or as a separate step, after the device has been programmed. Once the security bit is set it is impossible to verify (read) or program the PL22V10 until the entire device has first been erased with the bulk-erase function.

PROGRAM AND ERASE

The PL22V10 can be programmed on standard logic programmers. If a device needs to be reprogrammed, simply place back into the programmer, at which point it will be automatically erased, then repatterned.

Approved programmers are listed in the Philips Semiconductors Programmer Reference Guide.

SOFTWARE SUPPORT

The PL22V10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™, CUPL™, and PALASM® 90 design software packages also support the PL22V10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PL22V10 logic designs can also be generated using the program table entry format. This program table entry format is supported by SNAP only.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	CONDITIONS	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	Relative to GND	–0.5	+7.0	V
V_{IN}, V_{OUT}	Voltage applied to any pin[3]	Relative to GND[2]	–1.2	V_{CC} + 0.5	V_{DC}
I_{OUT}	Output current	Per pin (I_{OL}, I_{OH})	±25		mA
T_{stg}	Storage temperature range		–65	+125	°C
T_{LT}	Lead temperature	Soldering 10 seconds	+300		°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
2. Minimum DC input is –0.5V, however inputs may undershoot to –2.0V for periods less than 20ns.
3. V_{IN} and V_{OUT} are not specified for program/verify operation.

OPERATING RANGES

SYMBOL	PARAMETER	CONDITIONS	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	Commercial[1]	+4.75	+5.25	V_{DC}
		Industrial	+4.5	+5.5	V_{DC}
T_{amb}	Ambient temperature	Commercial[1]	0	+70	°C
		Industrial	–40	+85	°C
t_R	Clock Rise Time	See note 2		250	ns
t_F	Clock Fall Time	See note 2		250	ns
t_{RVCC}	V_{CC} Rise Time	See note 2		250	ms

NOTES:
1. Voltage applied to input or output must not exceed V_{CC} +0.3V.
2. Test points for Clock and V_{CC} in t_R, t_F, t_{CL}, t_{CH}, and t_{RESET} are referenced at 10% and 90% levels.

DC ELECTRICAL CHARACTERISTICS

Commercial = $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	CONDITIONS	LIMITS			UNIT
			MIN	TYP[4]	MAX	
Input voltage						
V_{IL}	Low		–0.3		0.8	V
V_{IH}	High		2.0		V_{CC} + 0.3	V
Output voltage						
V_{OL}	Low – TTL	V_{CC} = MIN, I_{OL} = 16mA			0.5	V
V_{OLC}	Low – CMOS	V_{CC} = MIN, I_{OL} = 10μA			0.1	V
V_{OH}	High – TTL	V_{CC} = MIN, I_{OH} = –4.0mA	2.4			V
V_{OHC}	High – CMOS	V_{CC} = MIN, I_{OH} = –10μA	V_{CC} – 0.1			V
Input current						
I_{IL}/I_{IH}	Input leakage current	V_{CC} = MAX, GND ≤ V_{IN} ≤ V_{CC}		1	±10	μA
Output current						
I_{OZ}	Output leakage	I/O = Hi-Z, GND ≤ V_O ≤ V_{CC}		2	±10	μA
I_{SC}[5]	Short circuit	V_{CC} = 5V, V_{OUT} = 0.5V[1]	–30		–130	mA
I_{CC}	V_{CC} active current, CMOS (commercial)	V_{IN} = V_{CC} or GND[2, 3]		80 + 0.5mA/MHz	110 + 0.5mA/MHz	mA
	V_{CC} active current, CMOS (industrial)	V_{IN} = V_{CC} or GND[2, 3]		90 + 0.5mA/MHz	120 + 0.5mA/MHz	mA
	V_{CC} active current, TTL (commercial)	V_{IN} = V_{IL} or V_{IH}[2, 3]		90 + 0.5mA/MHz	120 + 0.5mA/MHz	mA
	V_{CC} active current, TTL (industrial)	V_{IN} = V_{IL} or V_{IH}[2, 3]		100 + 0.5mA/MHz	130 + 0.5mA/MHz	mA

NOTES:

1. No more than one output should be tested at a time. Duration of the short-circuit test should not exceed one second.
2. I/O pins open (no load).
3. I_{CC} for a typical application: This parameter is tested with the device programmed as a 10-bit Counter.
4. Typical values are at V_{CC} = 5V. Typical values are guaranteed by design.
5. Room temperature only.

AC ELECTRICAL CHARACTERISTICS

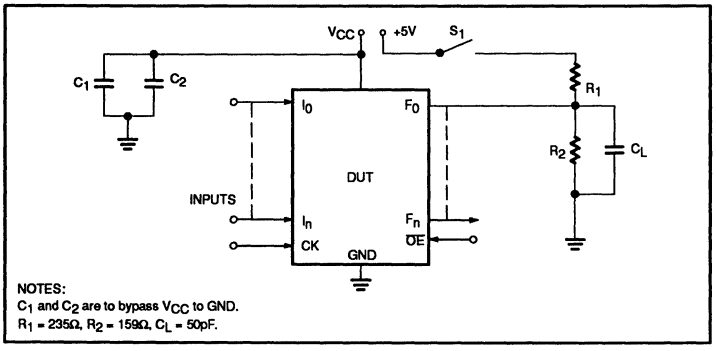
Commercial = $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$[1,2]

SYMBOL	PARAMETER	TEST CONDITIONS	PL22V10-10 MIN	PL22V10-10 MAX	UNIT
t_{PD}	Input[3] to non-registered output	50pF		10	ns
t_{EA}	Input[3] to Output Enable[4]	50pF		10	ns
t_{ER}	Input[3] to Output Disable[4]	5pF		10	ns
t_{CO}	Clock to output	50pF		8	ns
t_{CO2}	Clock to combinatorial output delay via internal registered feedback	50pF		14	ns
t_S	Input[3] or feedback setup to clock	50pF	7		ns
t_{SF}	Internal feedback[6]	50pF	5		ns
t_H	Input[3] hold after clock	50pF	0		ns
t_{WL}, t_{WH}	Clock width – clock low time, clock high time[5]	50pF	6		ns
t_{CP}	MIN clock period External ($t_S + t_{CO}$)	50pF	15		ns
f_{MAX1}	MAX operating frequency; Internal feedback[6] $\left(\frac{1}{t_{SF} + t_{CO}}\right)$	50pF	76.9		MHz
f_{MAX2}	MAX operating frequency; External ($1/t_{CP}$)	50pF	66.6		MHz
f_{MAX3}	MAX clock frequency; No feedback[6] $\left(\frac{1}{t_{WL} + t_{WH}}\right)$	50pF	83.3		MHz
t_{ARW}	Asynchronous Reset pulse width	50pF	10		ns
t_{AR}	Input[3] to Asynchronous Reset	50pF		12	ns
t_{ARR}	Asynchronous Reset recovery time	50pF	8		ns
t_{SPR}	Synchronous Preset recovery time	50pF	10		ns
t_{RESET}	Power-on reset time for registers in clear state[5]	50pF		5	µs
Capacitance[6]					
C_{IN}	Input Capacitance[7]	$T_{amb} = 25°C$, $V_{CC} = 5.0V$ @ f = 1MHz		6	pF
C_{OUT}	Output Capacitance[7]			12	pF

NOTES:

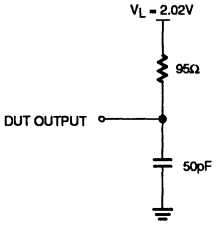
1. Test conditions assume: signal transition times of 2.5ns or less from the 10% and 90% points, timing reference levels of 1.5V (unless otherwise specified).
2. Device test loads are specified at the end of this section.
3. "Input" refers to an Input pin signal.
4. t_{OE} is measured from input transition to $V_{REF} \pm 0.1V$, t_{OD} is measured from input transition to $V_{OH} - 0.1V$ or $V_{OL} + 0.1V$.
5. Test points for Clock and V_{CC} in t_R, t_F, t_{CL}, t_{CH}, and t_{RESET} are referenced at 10% and 90% levels.
6. Parameters are not 100% tested. Specifications are based on initial characterization and are tested after any design or process modification which may affect operational frequency.
7. Capacitances are tested on a sample basis.

TEST LOAD CIRCUIT

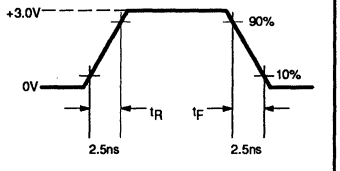


NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
R_1 = 235Ω, R_2 = 159Ω, C_L = 50pF.

THEVENIN EQUIVALENT



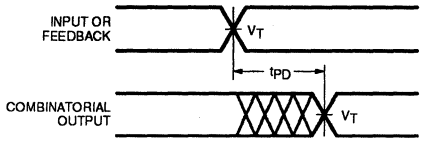
VOLTAGE WAVEFORM



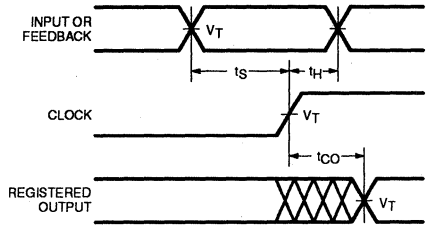
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

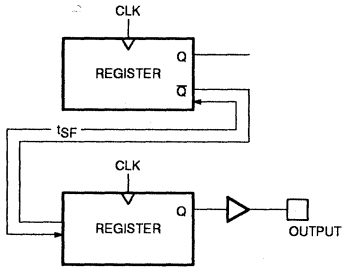
SWITCHING WAVEFORMS



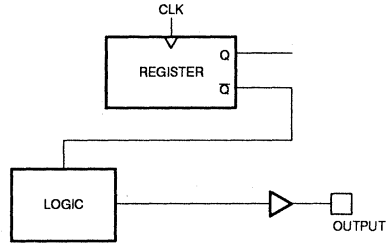
Combinatorial Output



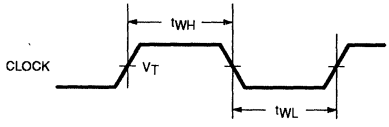
Registered Output



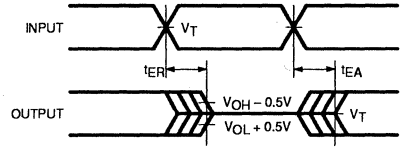
f_{MAX1}; Internal Feedback $\left(\frac{1}{t_{SF} + t_{CO}}\right)$



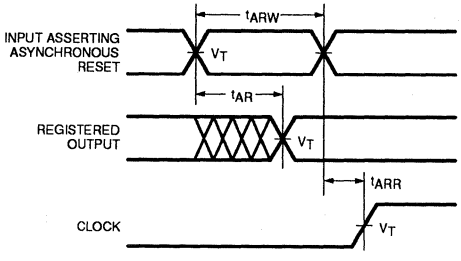
Clock to Combinatorial Output (t_{CO2})



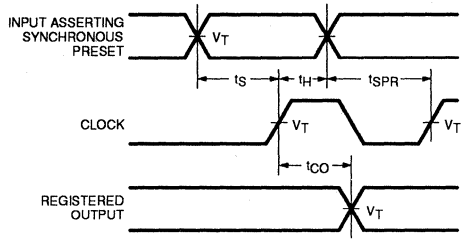
Clock Width



Input to Output Disable/Enable



Asynchronous Reset



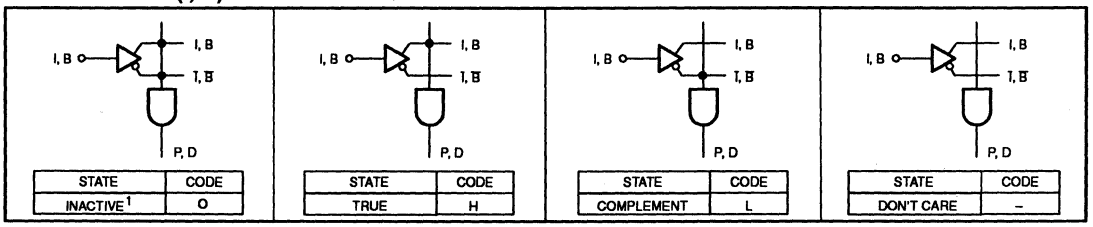
Synchronous Preset

NOTES:
1. V_T = 1.5V.
2. Input pulse amplitude 0V to 3.0V.
3. Input rise and fall times 2.5ns max.

CMOS programmable electrically erasable logic device

PL22V10-10

"AND" ARRAY – (I, B)



STATE	CODE
INACTIVE[1]	O

STATE	CODE
TRUE	H

STATE	CODE
COMPLEMENT	L

STATE	CODE
DON'T CARE	–

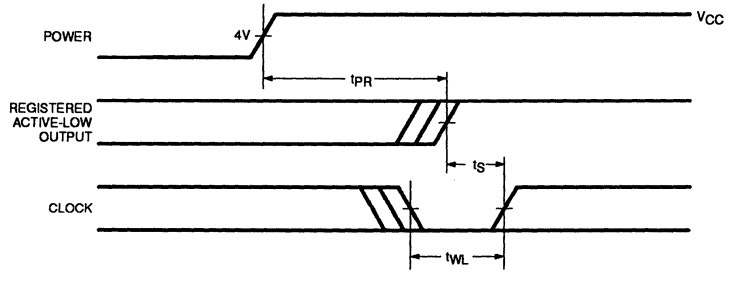
NOTE:
1. This is the initial state.

POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways V_{CC} can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The V_{CC} rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
t_{PR}	Power-up Reset Time		1	µs
t_S, t_{SF}	Input or Feedback Setup Time	See AC Electrical Characteristics		
t_{WL}	Clock Width LOW			

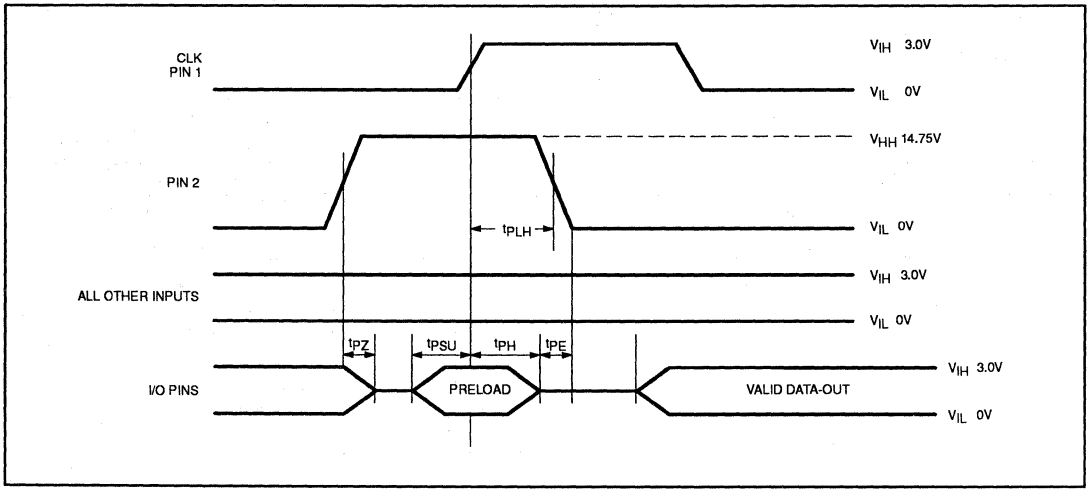


Power-Up Reset Waveform

PRELOAD TEST CONDITION

SYMBOL	PARAMETER	CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
t_{PE}	Valid data out			100		ns
t_{PZ}	Output 3-State delay time after assertion of Preload (Pin 2 = V_{HH})			100		ns
t_{PH}	Hold time of all preload inputs with respect to Clock rising edge			15		ns
t_{PSU}	Setup time of all preload inputs with respect to Clock rising edge			100		ns
t_{PLH}	Hold time for Preconditioning input			50		ns
V_{HH}	Preload enable voltage		14.50	14.75	15.0	V

PRELOAD WAVEFORM



CMOS programmable electrically erasable logic device

PL22V10-10

PROGRAM TABLE

B – COMBINATORIAL
D – REGISTERED
CONTROL WORD

POLARITY
H → ACTIVE HIGH
L → ACTIVE LOW
POLARITY

TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	F(I) 9	8	7	6	5	4	3	2	1	0	F(O) 9	8	7	6	5	4	3	2	1	0
0																							D								
1																							A								
2																							A								
3																							A								
4																							A								
5																							A								
6																							A								
7																							A								
8																							A								
9																								D							
10																								A							
11																								A							
12																								A							
13																								A							
14																								A							
15																								A							
16																								A							
17																								A							
18																								A							
19																								A							
20																									D						
21																									A						
22																									A						
23																									A						
24																									A						
25																									A						
26																									A						
27																									A						
28																									A						
29																									A						
30																									A						
31																									A						
32																									A						
33																										D					
34																										A					
35																										A					
36																										A					
37																										A					
38																										A					
39																										A					
40																										A					
41																										A					
42																										A					
43																										A					
44																										A					
45																										A					
46																										A					
47																										A					
48																											D				
49																											A				
50																											A				
51																											A				
52																											A				
53																											A				
54																											A				
55																											A				
56																											A				
57																											A				
58																											A				
59																											A				
60																											A				
61																											A				
62																											A				
63																											A				
64																											A				
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

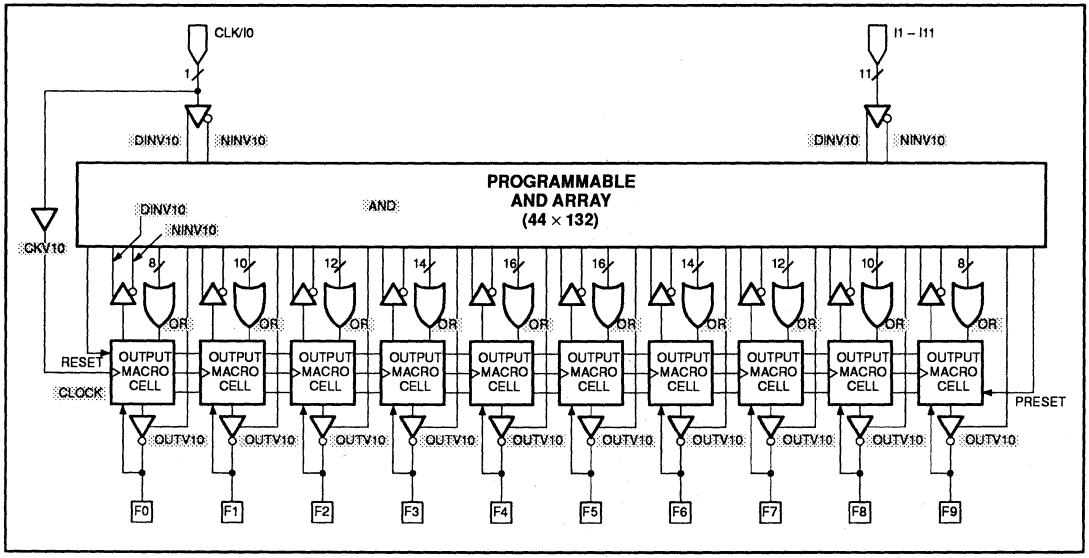
CMOS programmable electrically erasable logic device

PL22V10-10

PROGRAM TABLE (Continued)

TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	F(I) 9	8	7	6	5	4	3	2	1	0	F(O) 9	8	7	6	5	4	3	2	1	0
65																												D				
66																												A				
67																												A				
68																												A				
69																												A				
70																												A				
71																												A				
72																												A				
73																												A				
74																												A				
75																												A				
76																												A				
77																												A				
78																												A				
79																												A				
80																												A				
81																												A				
82																													D			
83																													A			
84																													A			
85																													A			
86																													A			
87																													A			
88																													A			
89																													A			
90																													A			
91																													A			
92																													A			
93																													A			
94																													A			
95																													A			
96																													A			
97																														D		
98																														A		
99																														A		
100																														A		
101																														A		
102																														A		
103																														A		
104																														A		
105																														A		
106																														A		
107																														A		
108																														A		
109																														A		
110																															D	
111																															A	
112																															A	
113																															A	
114																															A	
115																															A	
116																															A	
117																															A	
118																															A	
119																															A	
120																															A	
121																																D
122																																A
123																																A
124																																A
125																																A
126																																A
127																																A
128																																A
129																																A
AR																																
SP																																
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

SNAP RESOURCE SUMMARY DESIGNATIONS



CMOS programmable electrically erasable logic device

PL22V10-12/-15, PL22V10I15

FEATURES

- Advanced CMOS EEPROM technology
- Ultra high performance
  - 12ns, and 15ns (t_{PD}) commercial versions
  - 15ns (t_{PD}) industrial version
  - f_{MAX} as fast as 71.4MHz
- Available in Dual In-Line, Small Outline Large, and Plastic Leaded Chip Carrier packages
- Low power consumption
  - 90mA + 0.5mA/MHz max
- EE reprogrammability
  - Low-risk reprogrammable inventory
  - Superior programming and functional yield
  - 100% testable
  - Erases and programs in seconds
  - 100 guaranteed erase cycles
- Development and programming support
  - Third-party software and programmers
  - SLICE development software
- Architectural flexibility
  - 132 product term × 44 input AND array
  - Up to 22 inputs and 10 outputs
  - Variable product term distribution (8 to 16 per output) for greater logic flexibility
  - Independently programmable 4-configuration I/O macrocells
  - Synchronous preset, asynchronous clear
  - Independently programmable output enables
- Application versatility
  - Pin-for-pin and JEDEC-file compatible with the bipolar AmPAL22V10, CMOS PALC22V10 and PEEL22CV10A

DESCRIPTION

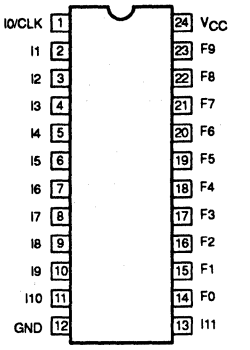
The Philips Semiconductors PL22V10-12 and PL22V10-15 are CMOS programmable electrically erasable logic devices that provide a high-performance, low-power, reprogrammable, and architecturally enhanced alternative to early generation programmable logic devices (PLDs). Designed in advanced CMOS EEPROM technology, the PL22V10 rivals speed parameters of comparable bipolar PLDs while providing a dramatic improvement in active power consumption. The EE reprogrammability of the PL22V10 allows cost effective plastic packaging, low risk inventory, reduced development and retrofit costs, and enhanced testability to ensure 100% field programmability and function. The PL22V10's flexible architecture offers complete function and JEDEC-file compatibility with the bipolar AmPAL22V10 and the CMOS PALC22V10. Applications for the PL22V10 include: replacement of random SSI/MSI logic circuitry and user customized sequential and combinatorial functions such as counters, shift registers, state machines, address decoders, multiplexers, etc. Development and programming support for the PL22V10 is provided by Philips Semiconductors and third-party manufacturers.

PIN LABEL DESCRIPTIONS

I1 – I11	Dedicated Input
NC	Not Connected
F0 – F9	Macro Cell Input/Output
CLK/I0	Clock Input/Dedicated Input
V_{CC}	Supply Voltage
GND	Ground

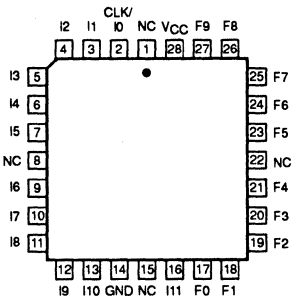
PIN CONFIGURATIONS

D and N Packages



N = Plastic Dual In-Line Package (300mil-wide)
D = Plastic Small Outline Large (300mil-wide) Package

A Package

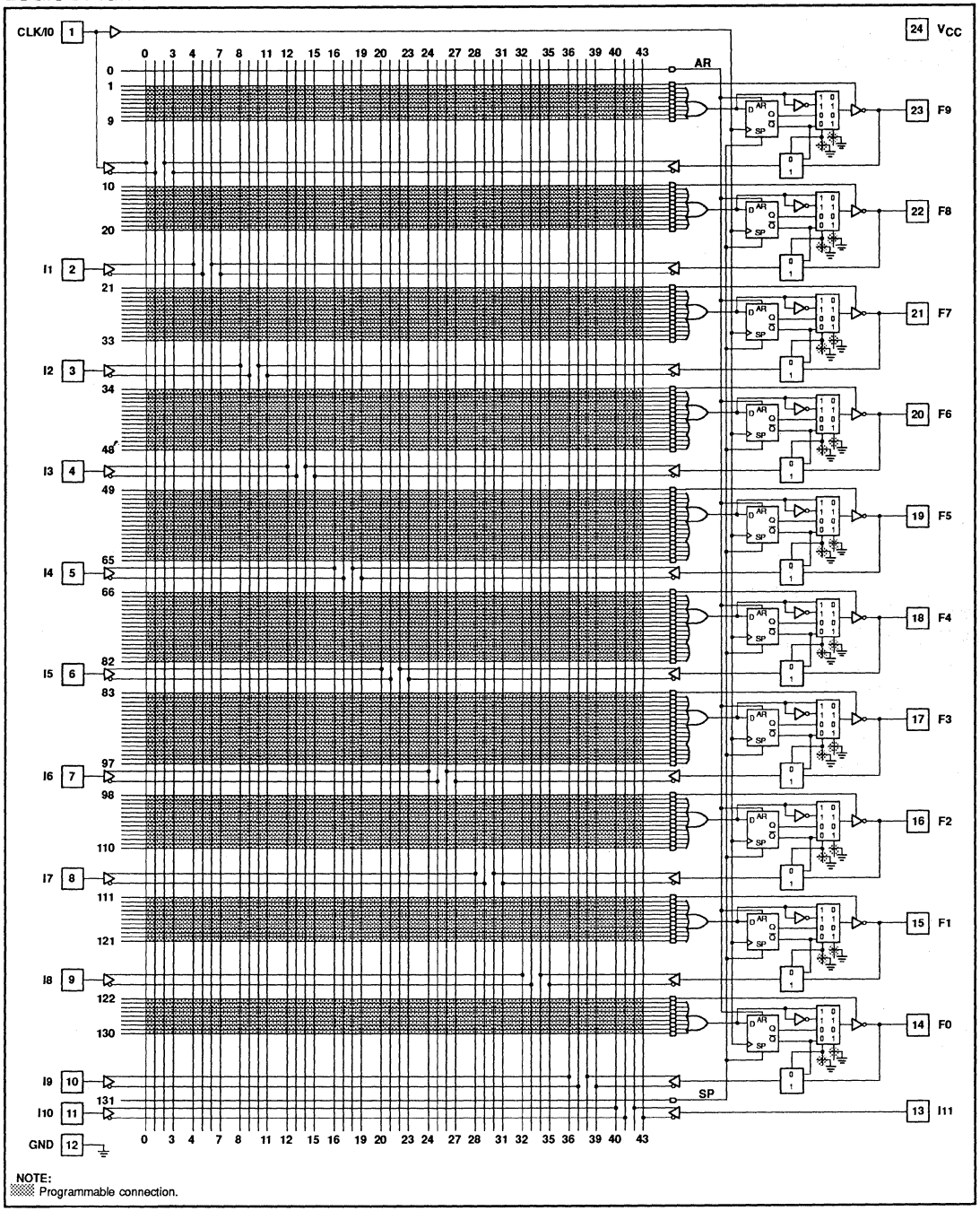


A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin (300mil-wide) Plastic DIP (Dual-In-Line Package)	PL22V10–12N PL22V10–15N PL22V10I15N (Industrial)	0410D
28-Pin (300mil-wide) PLCC (Plastic Leaded Chip Carrier Package)	PL22V10–12A PL22V10–15A PL22V10I15A (Industrial)	0401F
24-Pin (300mil-wide) Plastic SOL (Small Outline Large) Package	PL22V10–10D PL22V10–12D PL22V10–15D PL22V10I15D (Industrial)	0173D

LOGIC DIAGRAM



NOTE:
▒▒ Programmable connection.

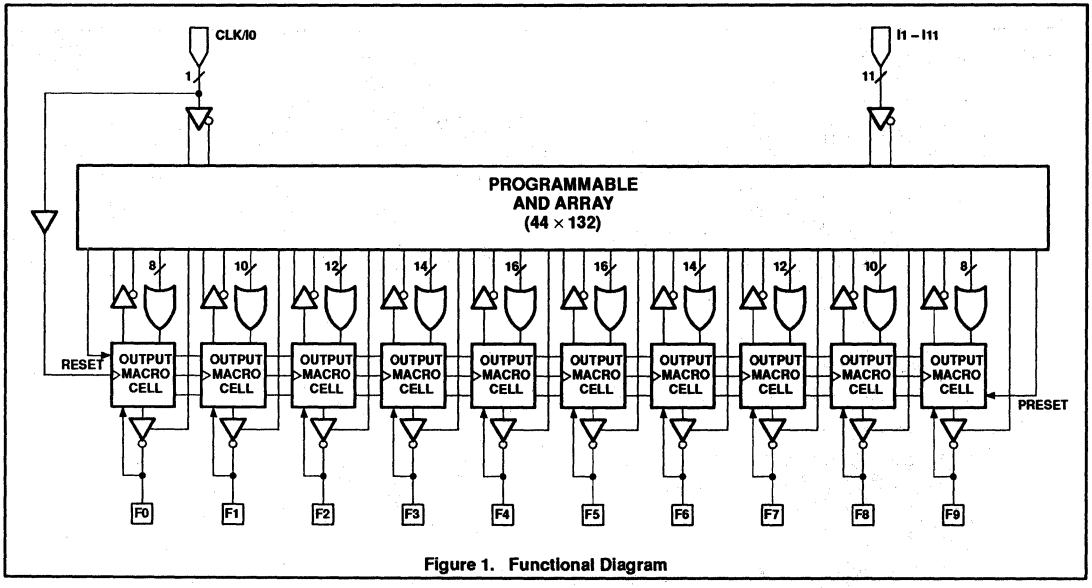


Figure 1. Functional Diagram

FUNCTION DESCRIPTION

The PL22V10 implements logic functions as sum-of-products expressions in a programmable-AND/fixed-OR logic array. User-defined functions are created by programming the connections of input signals into the array. User-configurable output structures in the form of I/O macrocells further increase logic flexibility.

ARCHITECTURE OVERVIEW

The PL22V10 architecture is illustrated in the Figure 1. Twelve dedicated inputs and 10 I/Os provide up to 22 inputs and 10 outputs for creation of logic functions. At the core of the device is a programmable electrically-erasable AND array which drives a fixed OR array. With this structure, the PL22V10 can implement up to 10 sum-of-products logic expressions.

Associated with each of the 10 OR functions is an I/O macro cell which can be independently programmed to one of 4 different configurations. The programmable macro cells allow each I/O to create sequential or combinatorial logic functions with either Active-High or Active-Low polarity.

AND/OR LOGIC ARRAY

The programmable AND array of the PL22V10 (shown in the Logic Diagram) is formed by input lines intersecting product terms. The input lines and product terms are used as follows:

44 input lines:

- 24 input lines carry the True and Complement of the signals applied to the 12 input pins
- 20 additional lines carry the True and Complement values of feedback or input signals from the 10 I/Os

132 product terms:

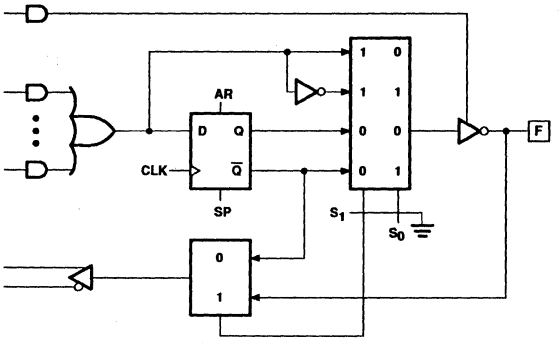
- 120 product terms (arranged in 2 groups of 8, 10, 12, 14, and 16) used to form logical sums
- 10 output enable terms (one for each I/O)
- 1 global synchronous preset term
- 1 global asynchronous clear term

At each input-line/product-term intersection there is an EEPROM memory cell which determines whether or not there is a logical connection at that intersection. Each product term is essentially a 44-input AND gate. A product term which is connected to both the True and Complement of an input signal will always be FALSE, and thus will not effect the OR function that it drives. When all the connections on a product term are opened, a Don't Care state exists and that term will always be TRUE.

When programming the PL22V10, the device programmer first performs a bulk erase to instantly remove the previous pattern. The erase cycle opens every logical connection in the array. The device is then configured to perform the user-defined function by programming selected connections in the AND array. (Note that EEPROM device programmers automatically program the connections on unused product terms so that they will have no effect on the output function.)

VARIABLE PRODUCT TERM DISTRIBUTION

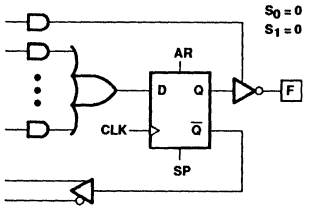
The PL22V10 provides 120 product terms to drive the 10 OR functions. These product terms are distributed among the outputs in groups of 8, 10, 12, 14, and 16 to form logical sums (see Logic Diagram). This distribution allows optimum use of device resources.



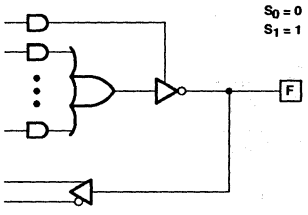
S_1	S_0	OUTPUT CONFIGURATION
0	0	Registered/Active-LOW
0	1	Registered/Active-HIGH
1	0	Combinatorial/Active-LOW
1	1	Combinatorial/Active-HIGH

0 = Unprogrammed fuse
1 = Programmed fuse

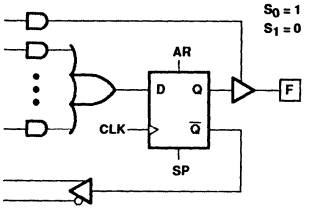
Figure 2. Output Macro Cell Logic Diagram



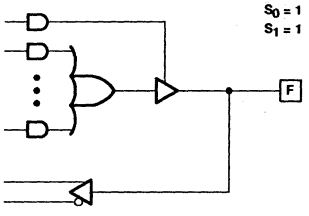
a. Registered/Active-LOW



c. Combinatorial/Active-LOW



b. Registered/Active-HIGH



d. Combinatorial/Active-HIGH

Figure 3.

CMOS programmable electrically erasable logic device

PL22V10-12/-15, PL22V10I15

PROGRAMMABLE I/O MACROCELL

The output macrocell provides complete control over the architecture of each output. The ability to configure each output independently permits users to tailor the configuration of the PL22V10 to the precise requirements of their designs.

MACROCELL ARCHITECTURE

Each I/O macrocell, as shown in Figure 2, consists of a D-type flip-flop and two signal-select multiplexers. The configuration of each macrocell of the PL22V10 is determined by the two EEPROM bits controlling these multiplexers. These bits determine output polarity, and output type (registered or non-registered). Equivalent circuits for the macrocell configurations are illustrated in Figure 3.

OUTPUT TYPE

The signal from the OR array can be fed directly to the output pin (combinatorial function) or latched in the D-type flip-flop (registered function). The D-type flip-flop latches data on the rising edge of the clock and is controlled by the global preset and clear terms. When the synchronous preset term is satisfied, the Q output of the register will be set HIGH at the next rising edge of the clock input. Satisfying the asynchronous clear term will set Q LOW, regardless of the clock state. If both terms are satisfied simultaneously, the clear will override the preset.

PROGRAM/ERASE CYCLES

The PL22V10 is 100% testable, erases/programs in seconds, and has 100 guaranteed erase cycles.

OUTPUT POLARITY

Each macrocell can be configured to implement Active-High or Active-Low logic. Programmable polarity eliminates the need for external inverters.

OUTPUT ENABLE

The output of each I/O macrocell can be enabled or disabled under the control of its associated programmable output enable product term. When the logical conditions programmed on the output enable term are satisfied, the output signal is propagated to the I/O pin. Otherwise, the output buffer is driven into the high-impedance state.

Under the control of the output enable term, the I/O pin can function as a dedicated input, a dedicated output, or a bi-directional I/O. Opening every connection on the output enable term will permanently enable the output buffer and yield a dedicated output. Conversely, if every connection is intact, the enable term will always be logically FALSE and the I/O will function as a dedicated input.

REGISTER FEEDBACK SELECT

When the I/O macrocell is configured to implement a registered function (S1=0) (Figures 3.a or 3.b), the feedback signal to the AND array is taken from the $\overline{Q}$ output.

BI-DIRECTIONAL I/O SELECT

When configuring an I/O macrocell to implement a combinatorial function (S1=1) (Figures 3.c or 3.d), the feedback signal is taken from the I/O pin. In this case, the pin can be used as a dedicated input, a dedicated output, or a bi-directional I/O.

POWER-ON RESET

To ease system initialization, all flip-flops will power-up to a reset condition and the Q output will be low. The actual output of the PL22V10 will depend on the programmed output polarity. The V_{CC} rise must be monotonic and the reset delay time is 5µs maximum.

DESIGN SECURITY

The PL22V10 provides a special EEPROM security bit that prevents unauthorized reading or copying of designs programmed into the device. The security bit is set by the PLD programmer, either at the conclusion of the programming cycle or as a separate step, after the device has been programmed. Once the security bit is set it is impossible to verify (read) or program the PL22V10 until the entire device has first been erased with the bulk-erase function.

PROGRAM AND ERASE

The PL22V10 can be programmed on standard logic programmers. If a device needs to be reprogrammed, simply place back into the programmer, at which point it will be automatically erased, then repatterned.

Approved programmers are listed in the Philips Semiconductors Programmer Reference Guide.

SOFTWARE SUPPORT

The PL22V10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™, CUPL™, and PALASM® 90 design software packages also support the PL22V10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PL22V10 logic designs can also be generated using the program table entry format. This program table entry format is supported by SNAP only.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	CONDITIONS	RATINGS		UNIT
			MIN	MAX	
V_{CC}	Supply voltage	Relative to GND	–0.5	+7.0	V
V_{IN}, V_{OUT}	Voltage applied to any pin[3]	Relative to GND[2]	–1.2	V_{CC} + 0.5	V_{DC}
I_{OUT}	Output current	Per pin (I_{OL}, I_{OH})	±25		mA
T_{stg}	Storage temperature range		–65	+125	°C
T_{LT}	Lead temperature	Soldering 10 seconds	+300		°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
2. Minimum DC input is –0.5V, however inputs may undershoot to –2.0V for periods less than 20ns.
3. V_{IN} and V_{OUT} are not specified for program/verify operation.

OPERATING RANGES

SYMBOL	PARAMETER	CONDITIONS	RATINGS		UNIT
			MIN	MAX	
V_{CC}	Supply voltage	Commercial[1]	+4.75	+5.25	V_{DC}
		Industrial	+4.5	+5.5	V_{DC}
T_{amb}	Ambient temperature	Commercial[1]	0	+70	°C
		Industrial	–40	+85	°C
t_R	Clock Rise Time	See note 2		250	ns
t_F	Clock Fall Time	See note 2		250	ns
t_{RVCC}	V_{CC} Rise Time	See note 2		250	ms

NOTES:
1. Voltage applied to input or output must not exceed V_{CC} +0.3V.
2. Test points for Clock and V_{CC} in t_R, t_F, t_{CL}, t_{CH}, and t_{RESET} are referenced at 10% and 90% levels.

DC ELECTRICAL CHARACTERISTICS

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V;
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V

SYMBOL	PARAMETER	CONDITIONS	LIMITS			UNIT
			MIN	TYP[4]	MAX	
Input voltage						
V_{IL}	Low		–0.3		0.8	V
V_{IH}	High		2.0		V_{CC} + 0.3	V
Output voltage						
V_{OL}	Low – TTL	V_{CC} = MIN, I_{OL} = 16mA			0.5	V
V_{OLC}	Low – CMOS	V_{CC} = MIN, I_{OL} = 10μA			0.1	V
V_{OH}	High – TTL	V_{CC} = MIN, I_{OH} = –4.0mA	2.4			V
V_{OHC}	High – CMOS	V_{CC} = MIN, I_{OH} = –10μA	V_{CC} – 0.1			V
Input current						
I_{IL}/I_{IH}	Input leakage current	V_{CC} = MAX, GND ≤ V_{IN} ≤ V_{CC}		1	±10	μA
Output current						
I_{OZ}	Output leakage	I/O = Hi-Z, GND ≤ V_O ≤ V_{CC}		2	±10	μA
I_{SC}[5]	Short circuit	V_{CC} = 5V, V_{OUT} = 0.5V[1]	–30		–130	mA
I_{CC}	V_{CC} active current, CMOS (commercial)	V_{IN} = V_{CC} or GND[2, 3]		70 + 0.5mA/MHz	90 + 0.5mA/MHz	mA
	V_{CC} active current, CMOS (industrial)	V_{IN} = V_{CC} or GND[2, 3]		80 + 0.5mA/MHz	100 + 0.5mA/MHz	mA
	V_{CC} active current, TTL (commercial)	V_{IN} = V_{IL} or V_{IH}[2, 3]		80 + 0.5mA/MHz	100 + 0.5mA/MHz	mA
	V_{CC} active current, TTL (industrial)	V_{IN} = V_{IL} or V_{IH}[2, 3]		90 + 0.5mA/MHz	110 + 0.5mA/MHz	mA

NOTES:
1. No more than one output should be tested at a time. Duration of the short-circuit test should not exceed one second.
2. I/O pins open (no load).
3. I_{CC} for a typical application: This parameter is tested with the device programmed as a 10-bit Counter.
4. Typical values are at V_{CC} = 5V. Typical values are guaranteed by design.
5. Room temperature only.

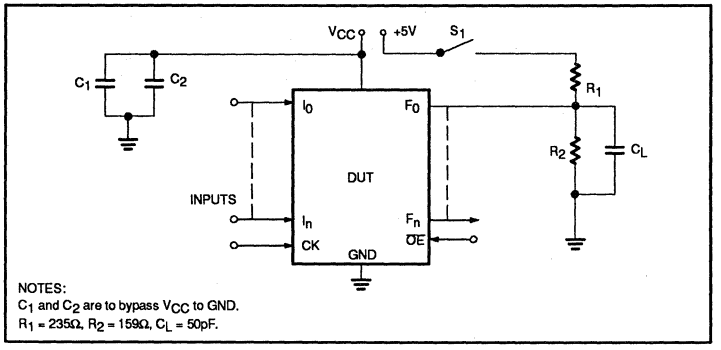
AC ELECTRICAL CHARACTERISTICS

Commercial = 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V[1, 2];
Industrial = –40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V

SYMBOL	PARAMETER	TEST CONDITIONS	PL22V10-12		PL22V10-15 PL22V10I15		UNIT
			MIN	MAX	MIN	MAX	
t_{PD}	Input[3] to non-registered output	50pF		12		15	ns
t_{EA}	Input[3] to Output Enable[4]	50pF		12		15	ns
t_{ER}	Input[3] to Output Disable[4]	5pF		12		15	ns
t_{CO}	Clock to output	50pF		9		10	ns
t_{CO2}	Clock to combinatorial output delay via internal registered feedback	50pF		16		19	ns
t_S	Input[3] or feedback setup to clock	50pF	8		10		ns
t_{SF}	Internal feedback[6]	50pF	6		9		ns
t_H	Input[3] hold after clock	50pF	0		0		ns
t_{WL}, t_{WH}	Clock width – clock low time, clock high time[5]	50pF	7		8		ns
t_{CP}	MIN clock period External ($t_S + t_{CO}$)	50pF	17		20		ns
f_{MAX1}	MAX operating frequency; Internal feedback[6] $\left(\frac{1}{t_{SF} + t_{CO}}\right)$	50pF	66.7		52.6		MHz
f_{MAX2}	MAX operating frequency; External ($1/t_{CP}$)	50pF	58.8		50.0		MHz
f_{MAX3}	MAX clock frequency; No feedback[6] $\left(\frac{1}{t_{WL} + t_{WH}}\right)$	50pF	71.4		62.5		MHz
t_{ARW}	Asynchronous Reset pulse width	50pF	12		15		ns
t_{AR}	Input[3] to Asynchronous Reset	50pF		15		18	ns
t_{ARR}	Asynchronous Reset recovery time	50pF	10		10		ns
t_{SPR}	Synchronous Preset recovery time	50pF	10		10		ns
t_{RESET}	Power-on reset time for registers in clear state[5]	50pF		5		5	µs
Capacitance[6]							
C_{IN}	Input Capacitance[7]	T_{amb} = 25°C, V_{CC} = 5.0V		6		6	pF
C_{OUT}	Output Capacitance[7]	@ f = 1MHz		12		12	pF

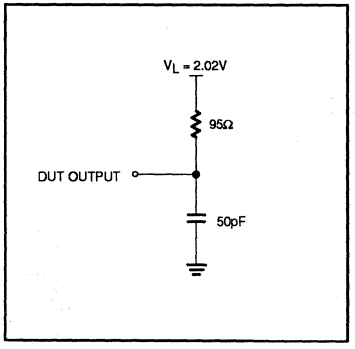
NOTES:
1. Test conditions assume: signal transition times of 2.5ns or less from the 10% and 90% points, timing reference levels of 1.5V (unless otherwise specified).
2. Device test loads are specified at the end of this section.
3. "Input" refers to an Input pin signal.
4. t_{OE} is measured from input transition to $V_{REF} \pm 0.1V$, t_{OD} is measured from input transition to $V_{OH} - 0.1V$ or $V_{OL} + 0.1V$.
5. Test points for Clock and V_{CC} in t_R, t_F, t_{CL}, t_{CH}, and t_{RESET} are referenced at 10% and 90% levels.
6. Parameters are not 100% tested. Specifications are based on initial characterization and are tested after any design or process modification which may affect operational frequency.
7. Capacitances are tested on a sample basis.

TEST LOAD CIRCUIT

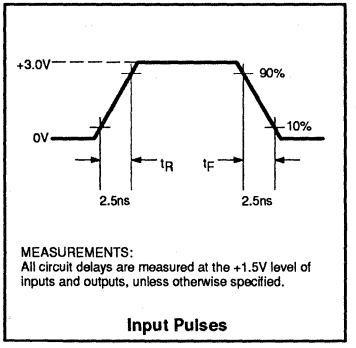


NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
R_1 = 235Ω, R_2 = 159Ω, C_L = 50pF.

THEVENIN EQUIVALENT



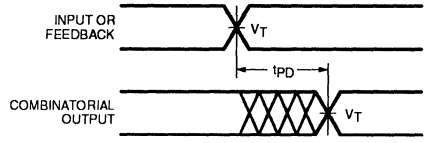
VOLTAGE WAVEFORM



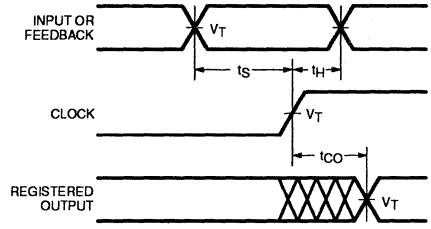
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

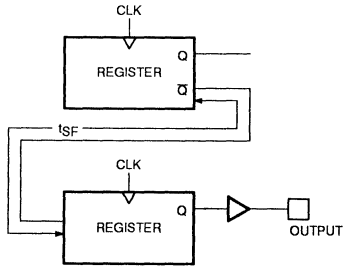
SWITCHING WAVEFORMS



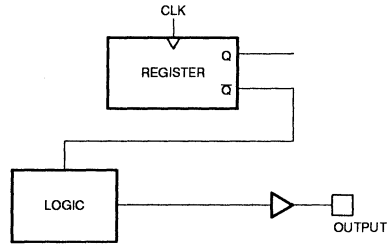
Combinatorial Output



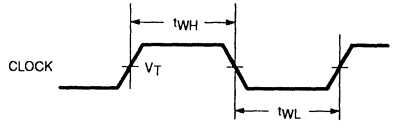
Registered Output



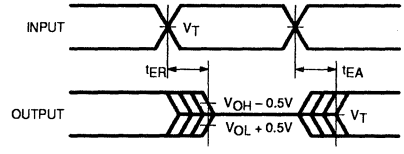
f_{MAX1}; Internal Feedback $\left(\frac{1}{t_{SF} + t_{CO}}\right)$



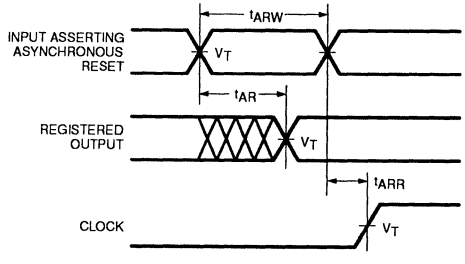
Clock to Combinatorial Output (t_{CO2})



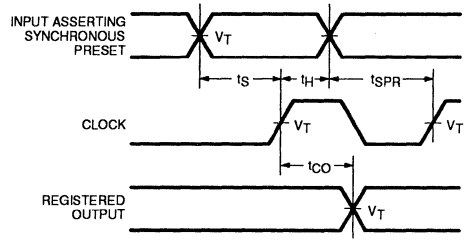
Clock Width



Input to Output Disable/Enable



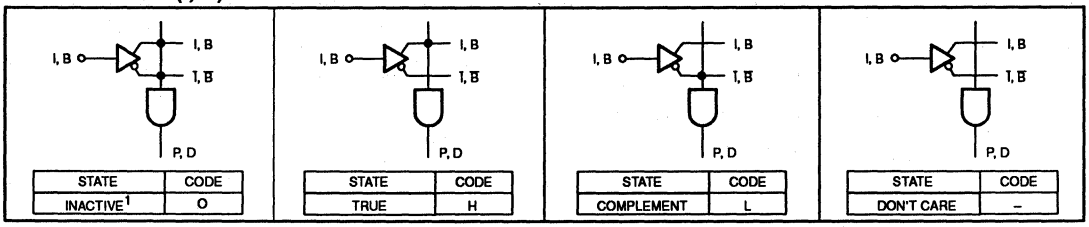
Asynchronous Reset



Synchronous Preset

NOTES:
1. V_T = 1.5V.
2. Input pulse amplitude 0V to 3.0V.
3. Input rise and fall times 2.5ns max.

"AND" ARRAY – (I, B)



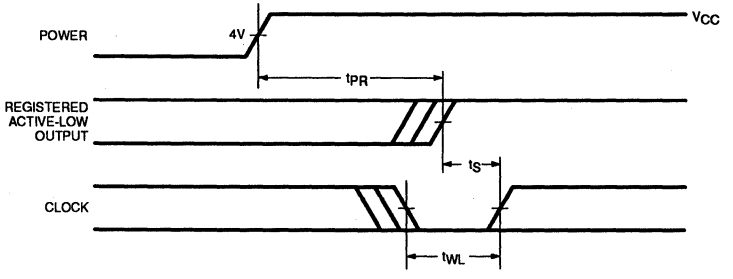
NOTE:
1. This is the initial state.

POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways V_{CC} can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The V_{CC} rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
t_{PR}	Power-up Reset Time		1	µs
t_S, t_{SF}	Input or Feedback Setup Time	See AC Electrical Characteristics		
t_{WL}	Clock Width LOW			

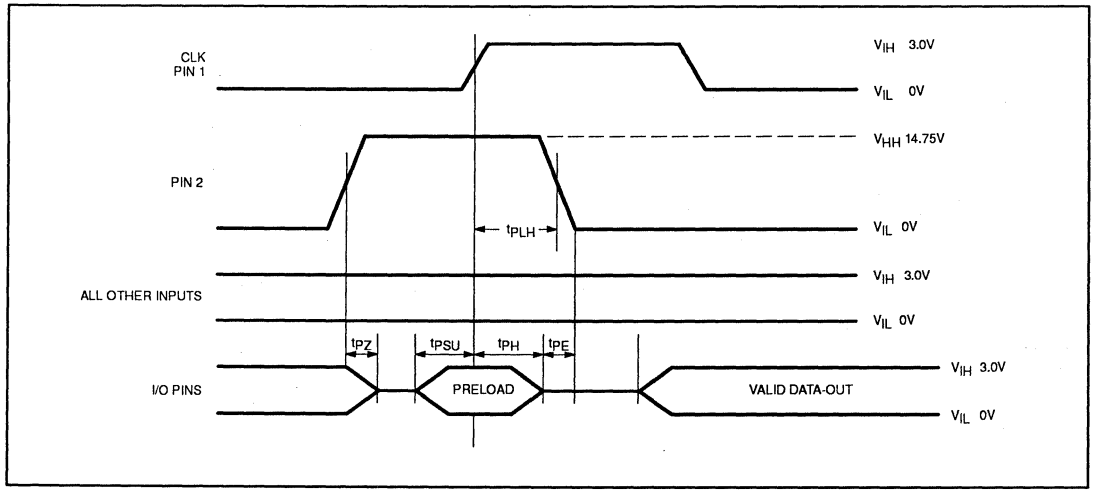


Power-Up Reset Waveform

PRELOAD TEST CONDITION

SYMBOL	PARAMETER	CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
t_{PE}	Valid data out			100		ns
t_{PZ}	Output 3-State delay time after assertion of Preload (Pin 2 = V_{HH})			100		ns
t_{PH}	Hold time of all preload inputs with respect to Clock rising edge			15		ns
t_{PSU}	Setup time of all preload inputs with respect to Clock rising edge			100		ns
t_{PLH}	Hold time for Preconditioning input			50		ns
V_{HH}	Preload enable voltage		14.50	14.75	15.0	V

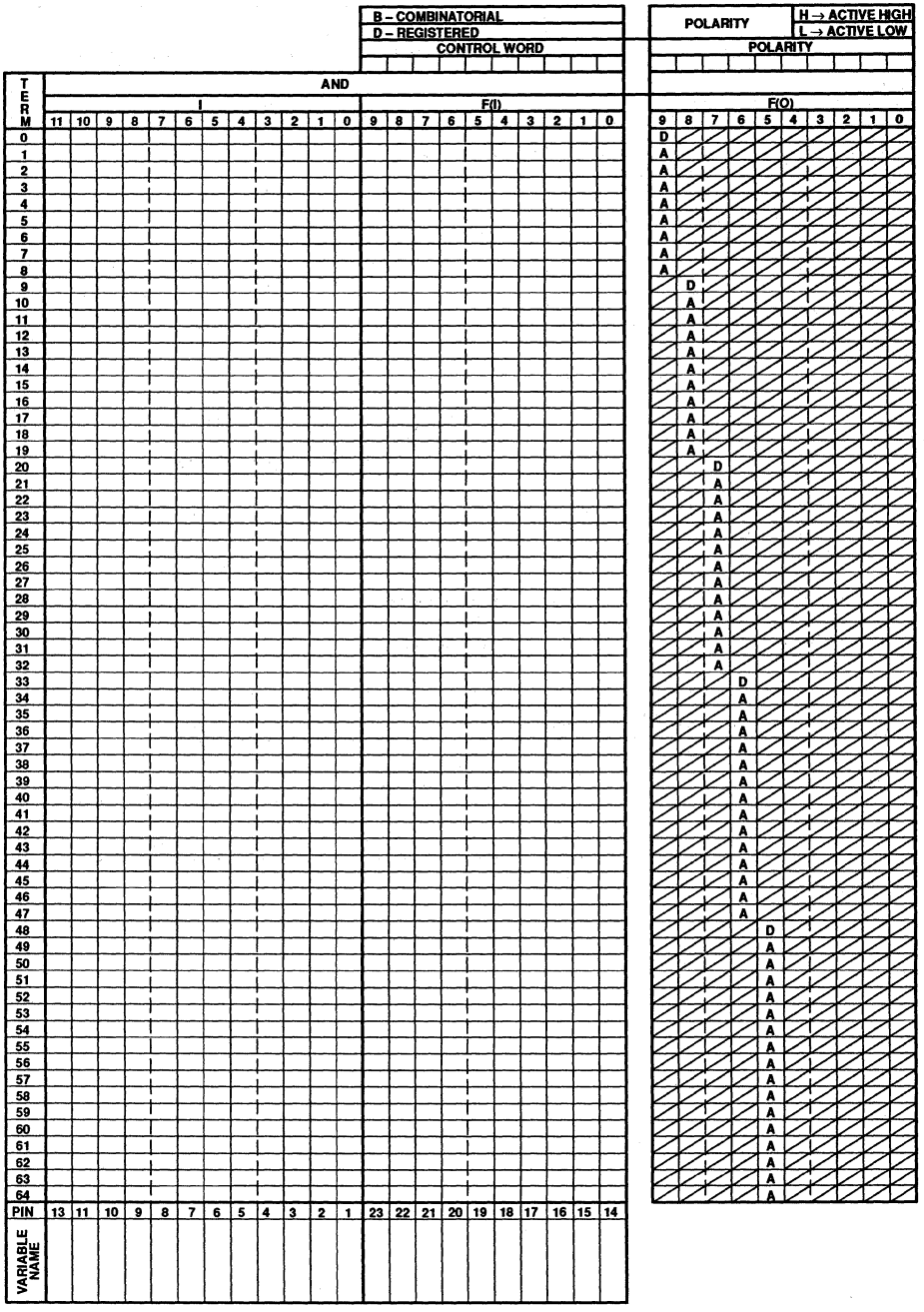
PRELOAD WAVEFORM



CMOS programmable electrically erasable logic device

PL22V10-12/-15, PL22V10I15

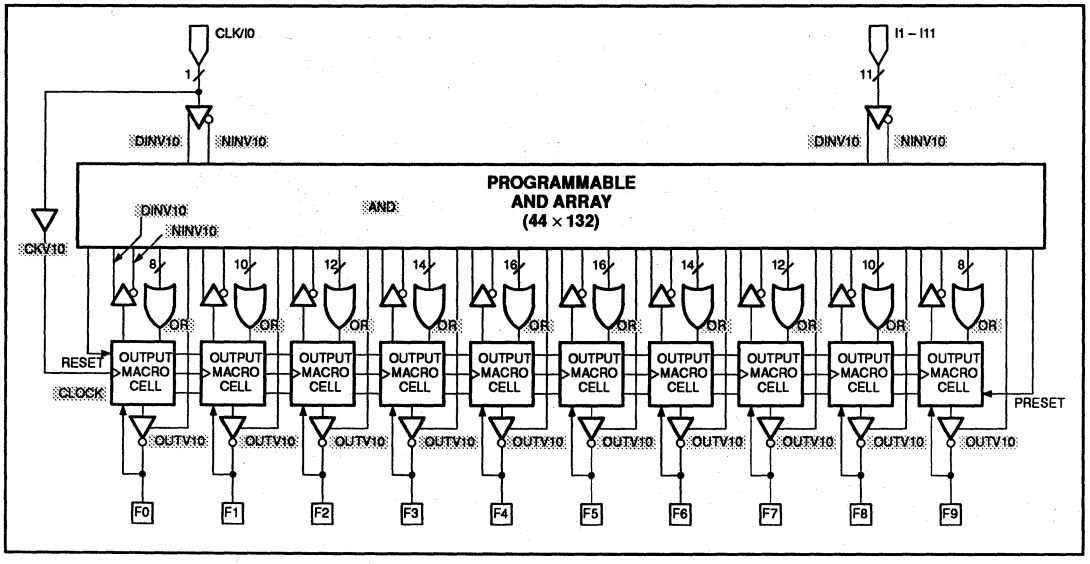
PROGRAM TABLE



PROGRAM TABLE (Continued)

TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	AND F(I) 9	8	7	6	5	4	3	2	1	0	F(O) 9	8	7	6	5	4	3	2	1	0
65																												D				
66																												A				
67																												A				
68																												A				
69																												A				
70																												A				
71																												A				
72																												A				
73																												A				
74																												A				
75																												A				
76																												A				
77																												A				
78																												A				
79																												A				
80																												A				
81																												A				
82																													D			
83																													A			
84																													A			
85																													A			
86																													A			
87																													A			
88																													A			
89																													A			
90																													A			
91																													A			
92																													A			
93																													A			
94																													A			
95																													A			
96																													A			
97																														D		
98																														A		
99																														A		
100																														A		
101																														A		
102																														A		
103																														A		
104																														A		
105																														A		
106																														A		
107																														A		
108																														A		
109																														A		
110																															D	
111																															A	
112																															A	
113																															A	
114																															A	
115																															A	
116																															A	
117																															A	
118																															A	
119																															A	
120																															A	
121																																D
122																																A
123																																A
124																																A
125																																A
126																																A
127																																A
128																																A
129																																A
AR																																
SP																																
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

SNAP RESOURCE SUMMARY DESIGNATIONS



BiCMOS versatile PAL device — ABT22V10-7

DESCRIPTION

The ABT22V10 is a versatile PAL[1] device fabricated with the Philips BiCMOS process known as QUBiC. The QUBiC process produces a very high speed device (7.5ns worst case) which has excellent noise immunity. The ground bounce, with 9 outputs switching and the 10th held low is, typically, less than 0.8V.

The ABT22V10 uses the familiar AND/OR logic array structure, which allows direct implementation of sum-of-product equations. This device has a programmable AND array which drives a fixed OR array. The AND array is programmed to create custom product terms while the fixed OR array sums selected terms at the output.

The OR sum of the products feeds the "Output Macro Cell" (OMC) which can be individually configured as a dedicated input, a combinatorial output, or a registered output with internal feedback. In other words, the architecture provides maximum design flexibility by allowing the Output Macro Cell to be configured by the user.

The ABT22V10 is designed so the outputs can never display a metastable state due to set up and hold time violations. If set up and hold times are violated, the outputs will not glitch or display a metastable state (the propagation delays may, however, be extended).

This device is pin and JEDEC file compatible with industry standard 22V10 and can be used in all standard applications where speed is to be maximized.

Order codes can be found in the Ordering Information table.

FEATURES

- Ultra fast 7.5ns t_{PD} and 6ns t_{CO}
- Pin and JEDEC file compatible to industry standard 22V10
- 10 input/output macro cells for architectural flexibility
- Metastable immune flip-flops
- Low ground bounce (<0.8V typical)
- Varied product term distribution with up to 16 product terms per output for complex functions
- Programmable output polarity
- Power-up reset on all registers
- Synchronous Preset/Asynchronous Reset
- Programmable on standard PAL-type device programmers
- Design support provided using SNAP software development package and other CAD tools for PLDs
- Available in 300mil-wide 24-Pin Plastic Dual In-Line Package and 28-Pin Plastic Leaded Chip Carrier

APPLICATIONS

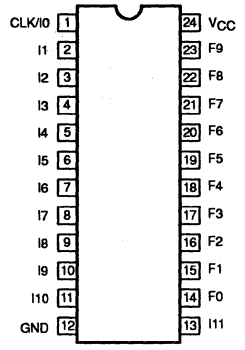
- DMA control
- State machine implementation
- High speed graphics processing
- Counters/shift registers
- SSI/MSI random logic replacement
- High speed memory decoder

PIN LABEL DESCRIPTIONS

I1 – I11	Dedicated Input
NC	Not Connected
F0 – F9	Macro Cell Input/Output
CLK/I0	Clock Input/Dedicated Input
V_{CC}	Supply Voltage
GND	Ground

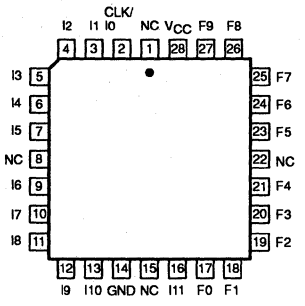
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



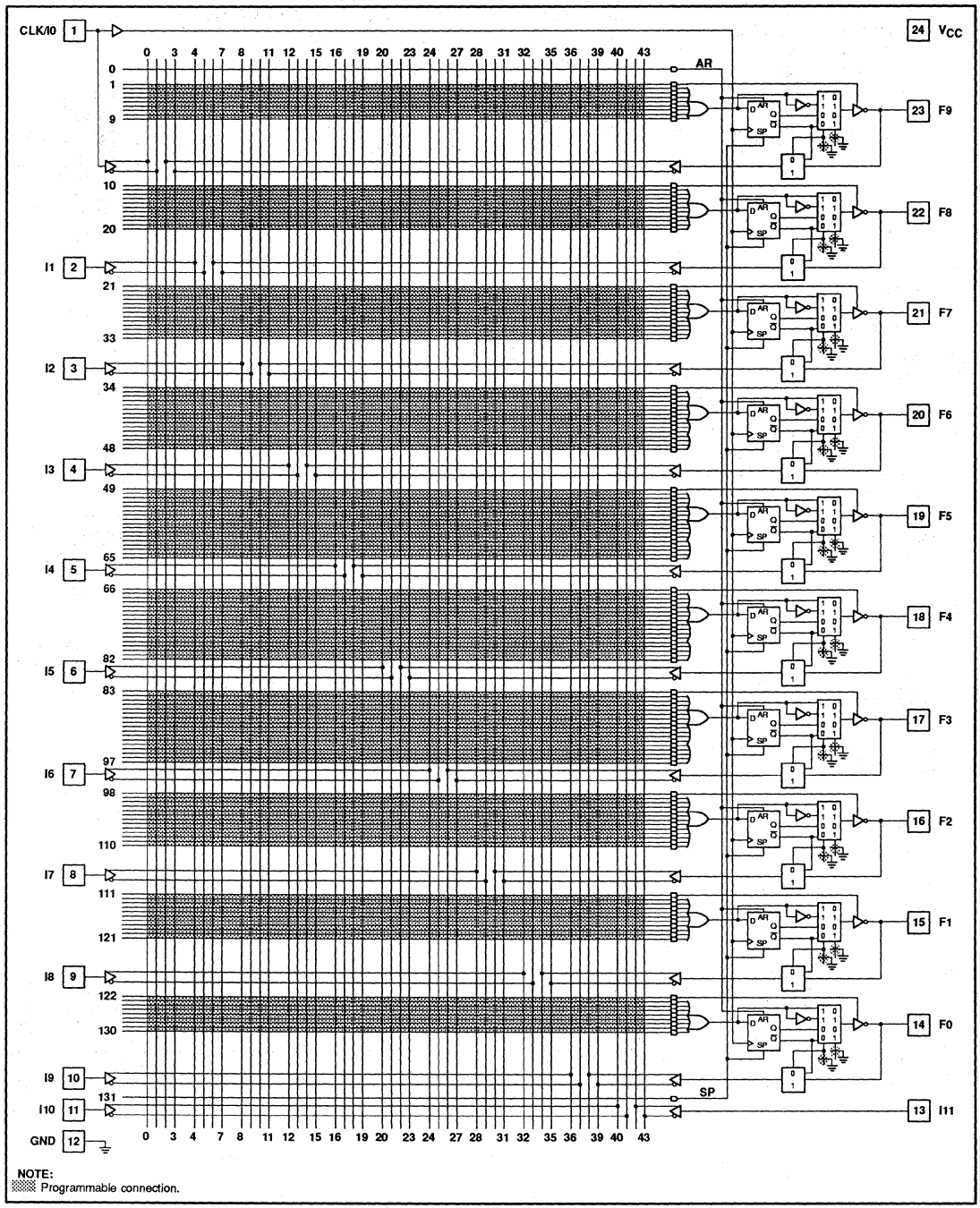
A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Plastic Dual-In-Line Package 300mil-wide	ABT22V10–7N	0410D
28-Pin Plastic Leaded Chip Carrier	ABT22V10–7A	0401F

1. ®PAL is a registered trademark of Advanced Micro Devices, Inc.

LOGIC DIAGRAM



NOTE:
▒▒ Programmable connection.

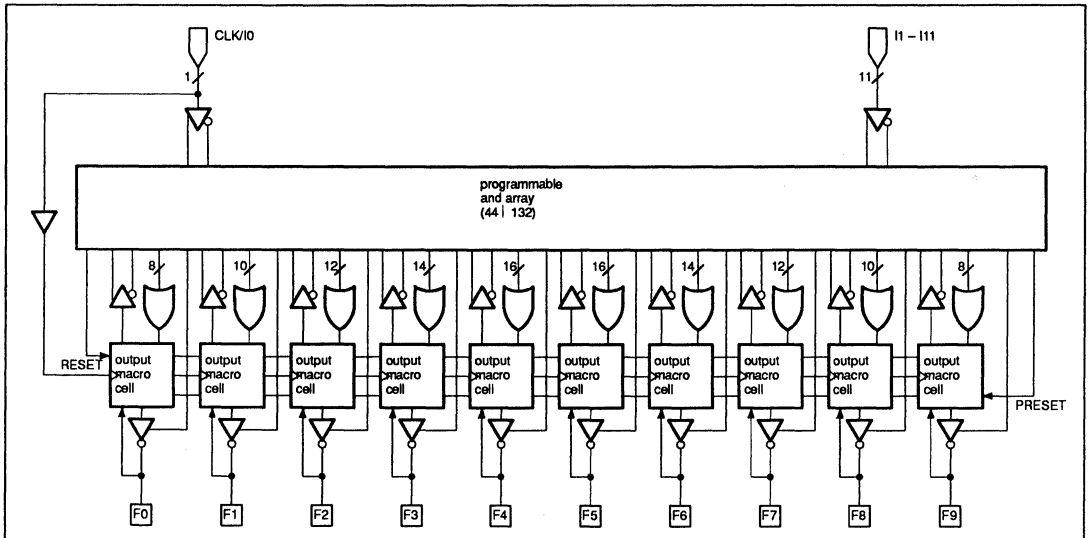


Figure 1. Functional Diagram

FUNCTIONAL DESCRIPTION

The ABT22V10 allows the systems engineer to implement the design on-chip, by opening fuse links to configure AND and OR gates within the device, according to the desired logic function.

Product terms with all fuses opened assume the logical HIGH state; product terms connected to both True and Complement of any single input assume the logical LOW state.

The ABT22V10 has 12 inputs and 10 I/O Macro Cells (Figure 1). The Macro Cell allows one of four potential output configurations, registered output or combinatorial I/O, Active-HIGH or Active-LOW (see Figure 7). The configuration choice is made according to the user's design specification and corresponding programming of the configuration bits $S_0 - S_1$. Multiplexer controls are connected to ground (0) through a programmable fuse link, selecting the "0" path through the multiplexer. Programming the fuse disconnects the control line from GND and it floats to V_{CC} (1), selecting the "1" path.

The device is produced with a fuse link at each input to the AND gate array, and connections may be selectively removed by applying appropriate voltages to the circuit. Utilizing an easily-implemented programming algorithm, these products can be rapidly programmed to any customized pattern. Information on approved programmers can be found in the Programmer Reference Guide. Extra test fuses are pre-programmed during manufacturing to ensure extremely high field programming yields, and provide extra test paths to achieve excellent parametric correlation.

Metastable Immune Characteristics

Philips Semiconductors uses the term 'metstable immune' to describe the output characteristics of registered logic devices. This term means that the outputs will not gitch or display an output anomaly under any circumstances, including set up and hold time violations. This claim is easily verified by following 74F5074 Synchronizing Dual flip-flop example. The ABT22V10 device has been designed using the same metastable immune flip-flop.

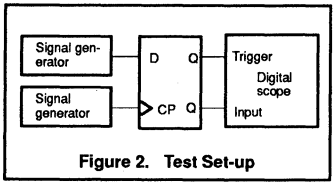


Figure 2. Test Set-up

By running two independent signal generators (see Figure 2) at nearly the same frequency (in this case 10MHz clock and 10.02MHz data) the device-under-test can be often be driven into a metastable state. If the Q output is then used to trigger a digital scope set to infinite persistence, the $\overline{Q}$ output will build a waveform. An experiment was run by continuously operating the devices in the region where metastability will occur.

When the device-under-test is an 74F74 (which was not designed with metastable immune characteristics) the waveform will appear as in Figure [3].

Figure 3 shows clearly that the $\overline{Q}$ output can vary in time with respect to the Q trigger point. This also implies that the Q or $\overline{Q}$ output waveshapes may be distorted. This can be verified on an analog scope with a charge plate CRT. Perhaps of even greater interest are the dots running along the 3.5V volt line in the upper right hand quadrant. These show that the $\overline{Q}$ output did not change state even though the Q output glitched to at least 1.5 volts, the trigger point of the scope.

When the device-under-test is a metastable immune part, such as the 74F5074, the waveform will appear as in Figure 4. The 74F5074 output will not vary with respect to the Q trigger point even when the a part is driven into a metastable state. Any tendency towards internal metastability is resolved by Philips Semiconductors patented circuitry. If a metastable event occurs within the flop the

COMPARISON OF METASTABLE IMMUNE AND NON–IMMUNE CHARACTERISTICS

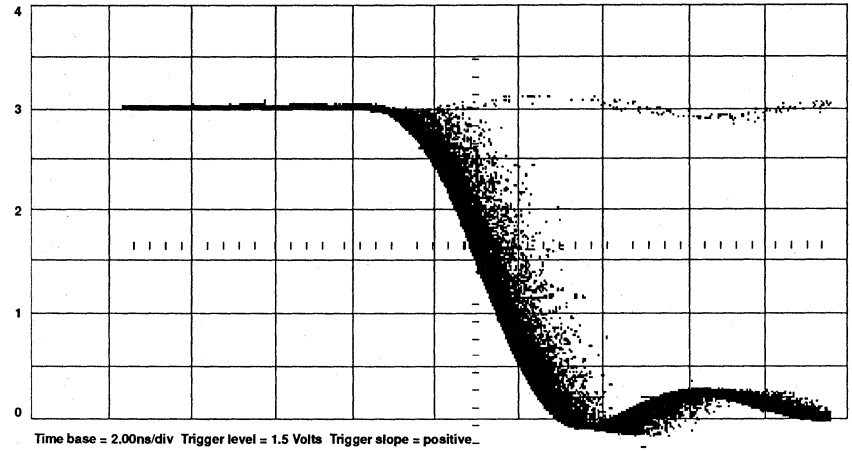


Figure 3. 74F74 $\overline{Q}$ output triggered by Q output, Setup and Hold times violated

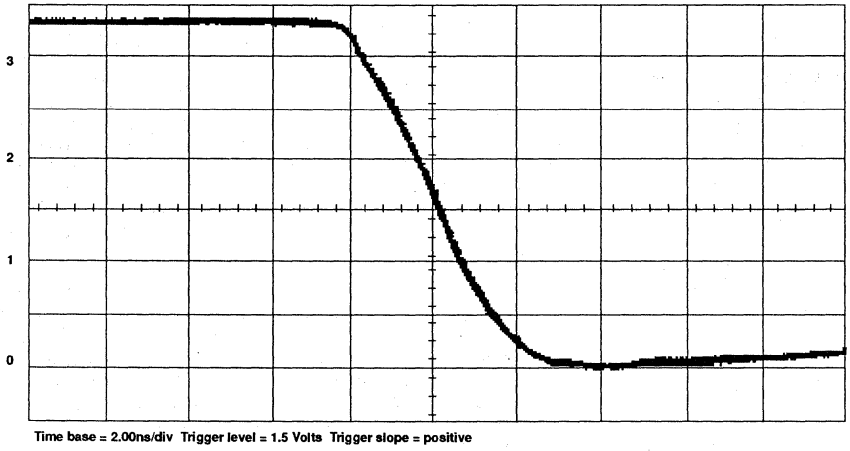


Figure 4. 74F5074 $\overline{Q}$ output triggered by Q output, Setup and Hold times violated

only outward manifestation of the event will be an increased clock-to-Q/$\overline{Q}$ propagation delay. This propagation delay is, of course, a function of the metastability characteristics of the part defined by τ and T_0.

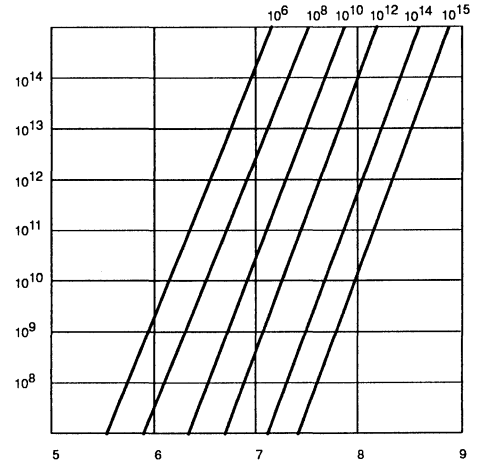
The metastability characteristics of the ABT22V10-7 and related part types represent state-of-the-art BiCMOS technology.

After determining the T_0 and t of the flop, calculating the mean time between failures (MTBF) is simple. Suppose a designer wants to use the 74F5074 for synchronizing asynchronous data that is arriving at 10MHz (as measured by a frequency counter), has a clock frequency of 50MHz, and has decided that he would like to sample the output of the 74F5074 10ns after the clock edge. He simply plugs his number into the equation below:

$$MTBF = e^{(t'/\tau)} / T_0 f_C f_I$$

In this formula, f_C is the frequency of the clock, f_I is the average input event frequency, and t' is the time after the clock pulse that the output is sampled (t' < h, h being the normal propagation delay). In this situation the f_I will be twice the data frequency of (20 MHz) because input events consist of both of low and high transitions. Multiplying f_I by f_C gives an answer of 10^{15} Hz^2. From Figure 5 it is clear that the MTBF for the 74F5074 device is greater than 10^{10} seconds. Using the above formula the actual MTBF is 1.51 X 10^{10} seconds or about 480 years. The MTBF for the ABT22V10-7, under the same condition is 5.6 trillion years.

MEAN TIME BETWEEN FAILURES (MTBF) VERSUS t'



NOTE: V_{CC} = 5V, T_{amb} = 25°C
F_C = clock frequency
F_i = input event frequency

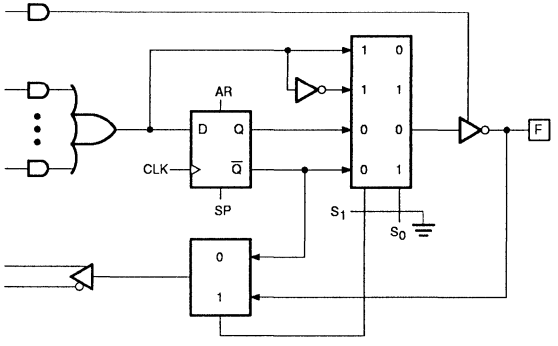
MTBF = exp(t'/t)/(to*Fc*Fi)

Figure 5. MTBF versus t' for ABT22V10-7 at τ = 83ps and T_O = 2.2 x 10^{17} sec

TYPICAL VALUES FOR τ AND T_0 AT VARIOUS V_{CC}S AND TEMPERATURES

	T_{amb} = 0°C		T_{amb} = 25°C		T_{amb} = 70°C	
V_{CC}	τ	T_0	τ	T_0	τ	T_0
5.5V	83ps	8.1 X 10^{18} sec	82ps	7.5 X 10^{18} sec	101ps	3.0 X 10^{12} sec
5.0V	80ps	4.0 X 10^{18} sec	83ps	2.2 X 10^{17} sec	98ps	4.4 X 10^{11} sec
4.5V	85ps	3.4 X 10^{14} sec	91ps	2.5 X 10^{12} sec	106ps	1.1 X 10^8 sec

OUTPUT MACRO CELL



S_1	S_0	OUTPUT CONFIGURATION
0	0	Registered/Active-LOW
0	1	Registered/Active-HIGH
1	0	Combinatorial/Active-LOW
1	1	Combinatorial/Active-LOW

0 = Unprogrammed fuse
1 = Programmed fuse

Figure 6. Output Macro Cell Logic Diagram

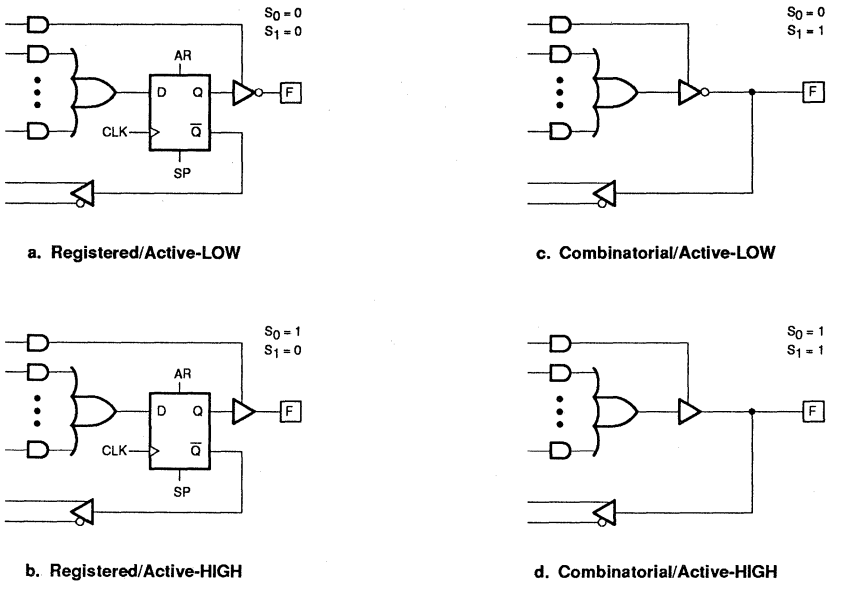


Figure 7. Output Macro Cell Configurations

Registered Output Configuration

Each Macro Cell of the ABT22V10 includes a D-type flip-flop for data storage and synchronization. The flip-flop is loaded on the LOW-to-HIGH transition of the clock input. In the registered configuration ($S_1 = 0$), the array feedback is from $\overline{Q}$ of the flip-flop.

Combinatorial I/O Configuration

Any Macro Cell can be configured as combinatorial by selecting the multiplexer path that bypasses the flip-flop ($S_1 = 1$). In the combinatorial configuration, the feedback is from the pin.

Variable Input/Output Pin Ratio

The ABT22V10 has twelve dedicated input lines, and each Macro Cell output can be an I/O pin. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Unused input pins should be tied to V_{CC} or GND.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7.0	V_{DC}
V_{IN}	Input voltage	–1.2	V_{CC} + 0.5	V_{DC}
V_{OUT}	Output voltage	–0.5	V_{CC} + 0.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

DC ELECTRICAL CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS[1]	LIMITS MIN	LIMITS MAX	UNIT
Input voltage					
V_{IL}	Low	V_{CC} = MIN		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V
V_I	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–1.2	V
Output voltage					
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}			
V_{OL}	Low	I_{OL} = 16mA		0.5	V
V_{OH}	High	I_{OH} = –3.2 mA	2.4		V
Input current					
		V_{CC} = MAX			
I_{IL} (except Pin 1)	Low	V_{IN} = 0.40V		–100	µA
I_{IL} (Pin 1)	Low	V_{IN} = 0.40V		–150	µA
I_{IH}	High	V_{IN} = 2.7V		25	µA
I_I	Maximum input current	V_{IN} = 5.5V		1.0	mA
Output current					
		V_{CC} = MAX			
I_{OZH}	Output leakage[3]	V_{IN} = V_{IL} or V_{IH}, V_{OUT} = 2.7V		100	µA
I_{OZL}	Output leakage[3]	V_{IN} = V_{IL} or V_{IH}, V_{OUT} = 0.4V		–100	µA
I_{SC}	Short circuit[2]	V_{OUT} = 0.5V	–30	–130	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX		210	mA

NOTES:
1. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
2. No more than one output should be tested at a time. Duration of the short-circuit test should not exceed one second. V_{OUT} = 0.5V has been chosen to avoid test problems caused by tester ground degradation.
3. I/O pin leakage is the worst case of I_{OZX} or I_{IX} (where X = H or L).

AC ELECTRICAL CHARACTERISTICS

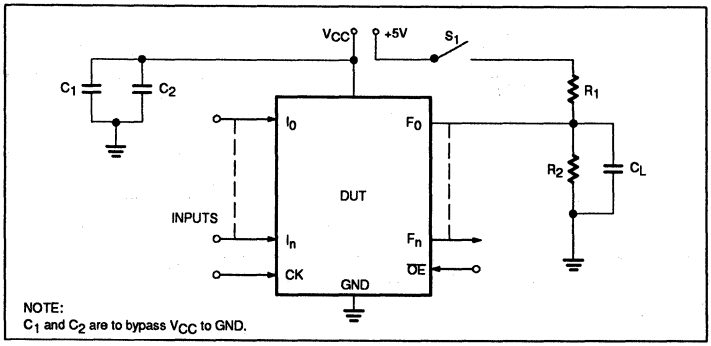
Over commercial operating range unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS		LIMITS[1] MIN	TYP	MAX	UNIT
t_{PD}	Input or feedback to non-registered output[2]	Active-LOW				7.5	
		Active-HIGH				7.5	ns
t_S	Setup time from input, feedback or SP to Clock			5.5			ns
t_H	Hold time			0			ns
t_{CO}	Clock to output					6.0	ns
t_{CF}	Clock to feedback[3]					2.5	ns
t_{AR}	Asynchronous Reset to registered output					10.0	ns
t_{ARW}	Asynchronous Reset width			7.5			ns
t_{ARR}	Asynchronous Reset recovery time			5.5			ns
t_{SPR}	Synchronous Preset recovery time			5.5			ns
t_{WL}	Width of Clock LOW			4.0			ns
t_{WH}	Width of Clock HIGH			4.0			ns
f_{MAX}	Maximum frequency; External feedback $1/(t_S + t_{CO})$[4]			87			MHz
	Maximum frequency; Internal feedback $1/(t_S + t_{CF})$[4]			125			MHz
t_{EA}	Input to Output Enable[5]					9.0	ns
t_{ER}	Input to Output Disable[5]					9.0	ns
Capacitance[6]							
C_{IN}	Input Capacitance (Pin 1)	V_{IN} = 2.0V	V_{CC} = 5.0V		6		pF
	Input Capacitance (Others)	V_{IN} = 2.0V	T_{amb} = 25°C		6		pF
C_{OUT}	Output Capacitance	V_{OUT} = 2.0V	f = 1MHz		8		pF

NOTES:

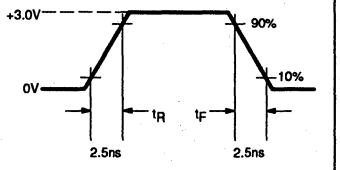
1. Commercial Test Conditions: R_1 = 300Ω, R_2 = 390Ω (see Test Load Circuit).
2. t_{PD} is tested with switch S_1 closed and C_L = 50pF (including jig capacitance). V_{IH} = 3V, V_{IL} = 0V, V_T = 1.5V.
3. Calculated from measured f_{MAX} internal.
4. These parameters are not 100% tested, but are calculated at initial characterization and at any time the design is modified where frequency may be affected.
5. For 3-State output; output enable times are tested with C_L = 50pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
6. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

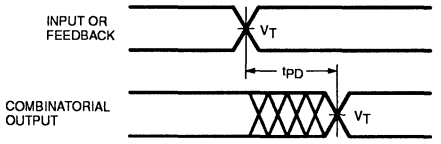
VOLTAGE WAVEFORM



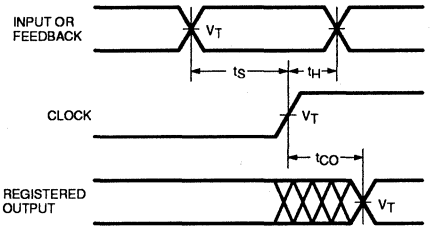
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

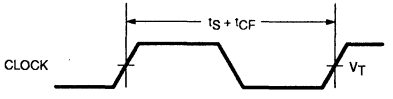
SWITCHING WAVEFORMS



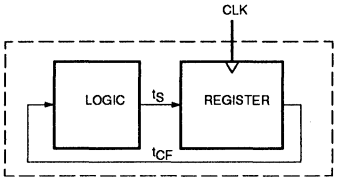
Combinatorial Output

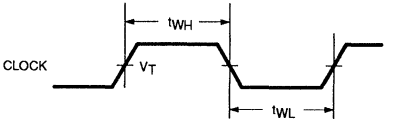


Registered Output

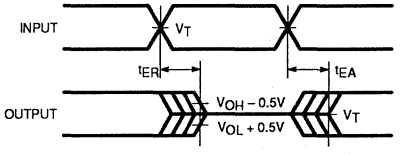


Clock to Feedback (f_{MAX} Internal)
(See Path at Right)

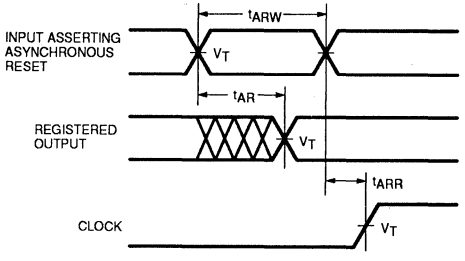




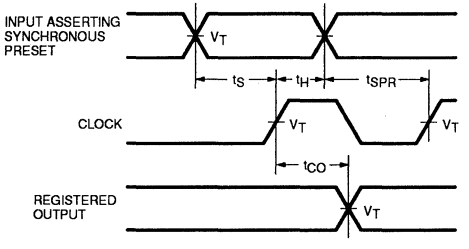
Clock Width



Input to Output Disable/Enable



Asynchronous Reset



Synchronous Preset

NOTES:
1. $V_T = 1.5V$.
2. Input pulse amplitude 0V to 3.0V.
3. Input rise and fall times 2.5ns max.

Programmable 3-State Outputs

Each output has a 3-State output buffer with 3-State control. A product term controls the buffer, allowing enable and disable to be a function of any product of device inputs or output feedback. The combinatorial output provides a bidirectional I/O pin, and may be configured as a dedicated input if the buffer is always disabled.

Programmable Output Polarity

The polarity of each macro cell output can be Active-HIGH or Active-LOW, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is controlled by programmable bit S_0 in the Output Macro Cell, and affects both registered and combinatorial outputs. Selection is automatic, based on the design specification and pin definitions. If the pin definition and output equation have the same polarity, the output is programmed to be Active-HIGH ($S_0 = 1$).

Preset/Reset

For initialization, the ABT22V10 has additional Preset and Reset product terms. These terms are connected to all registered outputs. When the Synchronous Preset (SP) product term is asserted high, the output registers will be loaded with a HIGH on the next LOW-to-HIGH clock transition. When the Asynchronous Reset (AR) product term is asserted high, the output registers will be immediately loaded with a LOW, independent of the clock.

Note that Preset and Reset control the flip-flop, not the output pin. The output level is determined by the output polarity selected.

Metastable Immune Flip-Flops

The D-type flip-flops have been designed such that the outputs will not glitch or display an output anomaly if the input set up or hold times are violated. Based on a τ of < 90 ps, and sampling the output 8ns after the clock edge, the typical MTBF is 170.3 years. If the sample is taken 10ns after the clock the MTBF is 5.6 trllion years.

Power-Up Reset

All flip-flops power-up to a logic LOW for predictable system initialization. Outputs of the ABT22V10 will depend on the programmed output polarity. The V_{CC} rise must be monotonic and the reset delay time is 1–10µs maximum.

Security Fuse

After programming and verification, a ABT22V10 design can be secured by programming the security fuse link. Once programmed, this fuse defeats readback of the internal programmed pattern by a device programmer, securing proprietary designs from competitors. When the security fuse is programmed, the array will read as if every fuse is programmed.

Quality and Testability

The ABT22V10 offers a very high level of built-in quality. Extra programmable fuses provide a means of verifying performance of all AC and DC parameters. In addition, this verifies programmability and functionality of the device to provide the highest programming and post-programming functional yields.

Low Ground Bounce

The Philips Semiconductors BiCMOS QUBiC process produces exceptional noise immunity. The typical ground bounce, with 9 outputs simultaneously switching and the 10th output held low, is less than 0.8V.

Technology

The BiCMOS ABT22V10 is fabricated with the Philips Semiconductors process known as QUBiC. QUBiC combines an advanced, state-of-the-art 1.0µm (drawn feature size) CMOS process with an ultra fast bipolar process to achieve superior speed and drive capabilities. QUBiC incorporates three layers of Al/Cu interconnects for reduced chip size, and our proven Ti-W fuse technology ensures highest programming yields.

Programming

The ABT22V10–7 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the ABT22V10–7 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Support Material)* of this data handbook for additional information.

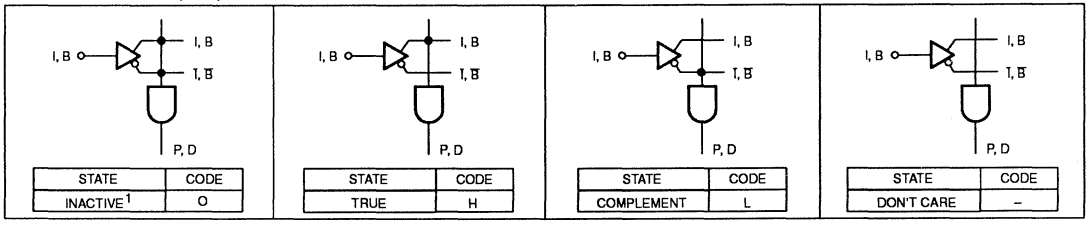
OUTPUT REGISTER PRELOAD

The register on the ABT22V10 can be preloaded from the output pins to facilitate functional testing of complex state machine designs. This feature allows direct loading of arbitrary states, making it unnecessary to cycle through long test vector sequences to reach a desired state. In addition, transitions from illegal states can be verified by loading illegal states and observing proper recovery. The procedure for preloading follows:

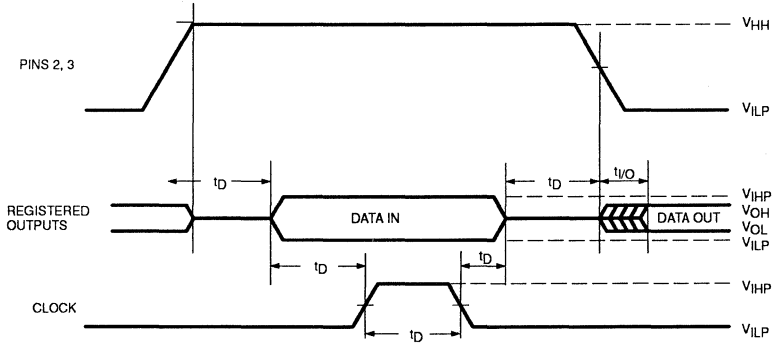
1. Raise V_{CC} to 5.0V ± 0.25V.
2. Set Pin 2 or 3 to V_{HH} to disable outputs and enable preload.
3. Apply the desired value (V_{ILP}/V_{IHP}) to all registered output pins. Leave combinatorial output pins floating.
4. Clock Pin 1 from V_{ILP} to V_{IHP}.
5. Remove V_{ILP}/V_{IHP} from all registered output pins.
6. Lower Pin 2 or 3 to V_{ILP}.
7. Enable the output registers according to the programmed pattern.
8. Verify V_{OL}/V_{OH} at all registered output pins. Note that the output pin signal will depend on the output polarity.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

"AND" ARRAY – (I, B)

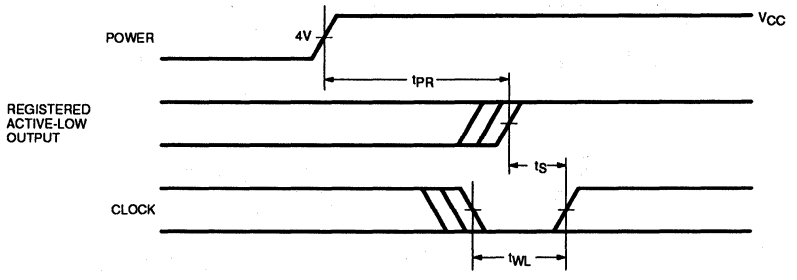


SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	REC	MAX	
V_{HH}	Super-level input voltage	9.5	9.5	10	V
V_{ILP}	Low-level input voltage	0	0	0.8	V
V_{IHP}	High-level input voltage	2.4	5.0	5.5	V
t_D	Delay time	100	200	1000	ns
$t_{I/O}$	I/O valid after Pin 2 or 3 drops from V_{HH} to V_{ILP}	100			ns



Output Register Preload Waveform

NOTE:
1. This is the initial state.



Power-Up Reset Waveform

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
t_{PR}	Power-up Reset Time		1	µs
t_S	Input or Feedback Setup Time	See AC Electrical Characteristics		
t_{WL}	Clock Width LOW			

POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways V_{CC} can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The V_{CC} rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

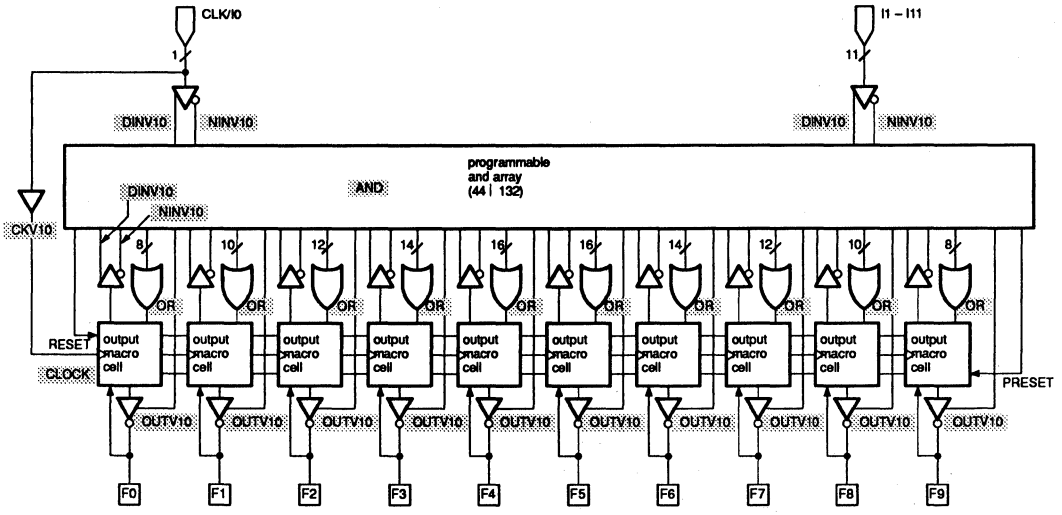
PROGRAM TABLE

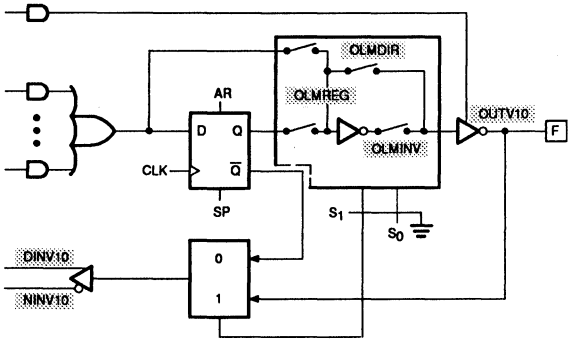
TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	CONTROL WORD F(I) 9	8	7	6	5	4	3	2	1	0	POLARITY F(O) 9	8	7	6	5	4	3	2	1	0
0																							D								
1																							A								
2																							A								
3																							A								
4																							A								
5																							A								
6																							A								
7																							A								
8																							A								
9																								D								
10																								A								
11																								A								
12																								A								
13																								A								
14																								A								
15																								A								
16																								A								
17																								A								
18																								A								
19																								A								
20																									D							
21																									A							
22																									A							
23																									A							
24																									A							
25																									A							
26																									A							
27																									A							
28																									A							
29																									A							
30																									A							
31																									A							
32																									A							
33																										D						
34																										A						
35																										A						
36																										A						
37																										A						
38																										A						
39																										A						
40																										A						
41																										A						
42																										A						
43																										A						
44																										A						
45																										A						
46																										A						
47																										A						
48																											D					
49																											A					
50																											A					
51																											A					
52																											A					
53																											A					
54																											A					
55																											A					
56																											A					
57																											A					
58																											A					
59																											A					
60																											A					
61																											A					
62																											A					
63																											A					
64																											A					
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

PROGRAM TABLE (Continued)

TERM	AND I 11	10	9	8	7	6	5	4	3	2	1	0	AND F(I) 9	8	7	6	5	4	3	2	1	0	F(O) 9	8	7	6	5	4	3	2	1	0
65																												D				
66																												A				
67																												A				
68																												A				
69																												A				
70																												A				
71																												A				
72																												A				
73																												A				
74																												A				
75																												A				
76																												A				
77																												A				
78																												A				
79																												A				
80																												A				
81																												A				
82																												D				
83																												A				
84																												A				
85																												A				
86																												A				
87																												A				
88																												A				
89																												A				
90																												A				
91																												A				
92																												A				
93																												A				
94																												A				
95																												A				
96																												A				
97																													D			
98																													A			
99																													A			
100																													A			
101																													A			
102																													A			
103																													A			
104																													A			
105																													A			
106																													A			
107																													A			
108																													A			
109																													A			
110																														D		
111																														A		
112																														A		
113																														A		
114																														A		
115																														A		
116																														A		
117																														A		
118																														A		
119																														A		
120																														A		
121																																D
122																																A
123																																A
124																																A
125																																A
126																																A
127																																A
128																																A
129																																A
AR																																
SP																																
PIN	13	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	19	18	17	16	15	14										
VARIABLE NAME																																

SNAP RESOURCE SUMMARY DESIGNATIONS





Output Macro Cell

ECL programmable array logic — 10H20EV8/10020EV8

DESCRIPTION

The 10H20EV8/10020EV8 is an ultra high-speed universal ECL PAL® device. Combining versatile output macrocells with a standard AND/OR single programmable array, this device is ideal in implementing a user's custom logic. The use of Philips Semiconductors state-of-the-art bipolar oxide isolation process enables the 10H20EV8/10020EV8 to achieve optimum speed in any design. The SNAP design software package from Philips Semiconductors simplifies design entry based upon Boolean or state equations.

The 10H20EV8/10020EV8 is a two-level logic element comprised of 11 fixed inputs, an input pin that can either be used as a clock or 12th input, 90 AND gates, and 8 Output Logic Macrocells. Each Output Macrocell can be individually configured as a dedicated input, dedicated output with polarity control, a bidirectional I/O, or as a registered output that has both output polarity control and feedback to the AND array. This gives the part the capability of having up to 20 inputs and eight outputs.

The 10H20EV8/10020EV8 has a variable number of product terms that can be OR'd per output. Four of the outputs have 12 AND terms available and the other four have 8 terms per output. This allows the designer the extra flexibility to implement those functions that he couldn't in a standard PAL device. Asynchronous Preset and Reset product terms are also included for system design ease. Each output has a separate output enable product term. Another feature added for the system designer is a power-up Reset on all registered outputs.

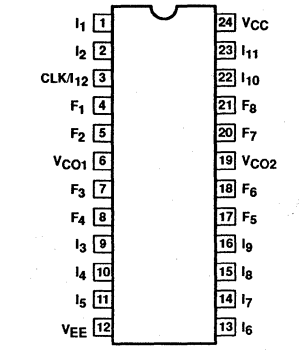
The 10H20EV8/10020EV8 also features the ability to Preload the registers to any desired state during testing. The Preload is not affected by the pattern within the device, so can be performed at any step in the testing sequence. This permits full logical verification even after the device has been patterned.

FEATURES

- Ultra high speed ECL device
  - t_{PD} = 4.5ns (max)
  - t_{IS} = 2.6ns (max)
  - t_{CKO} = 2.3ns (max)
  - f_{MAX} = 208MHz
- Universal ECL Programmable Array Logic
  - 8 user programmable output macrocells
  - Up to 20 inputs and 8 outputs
  - Individual user programmable output polarity
- Variable product term distribution allows increased design capability
- Asynchronous Preset and Reset capability
- 10KH and 100K options
- Power-up Reset and Preload function to enhance state machine design and testing
- Design support provided via SNAP and other CAD tools
- Security fuse for preventing design duplication
- Available in 24-Pin 300mil-wide DIP and 28-Pin PLCC.

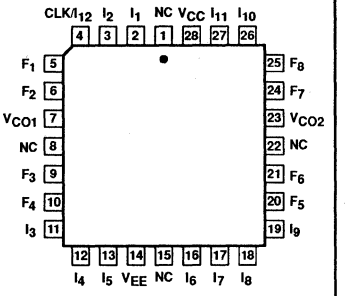
PIN CONFIGURATIONS

F Package



F = Ceramic DIP (300mil-wide)

A Package



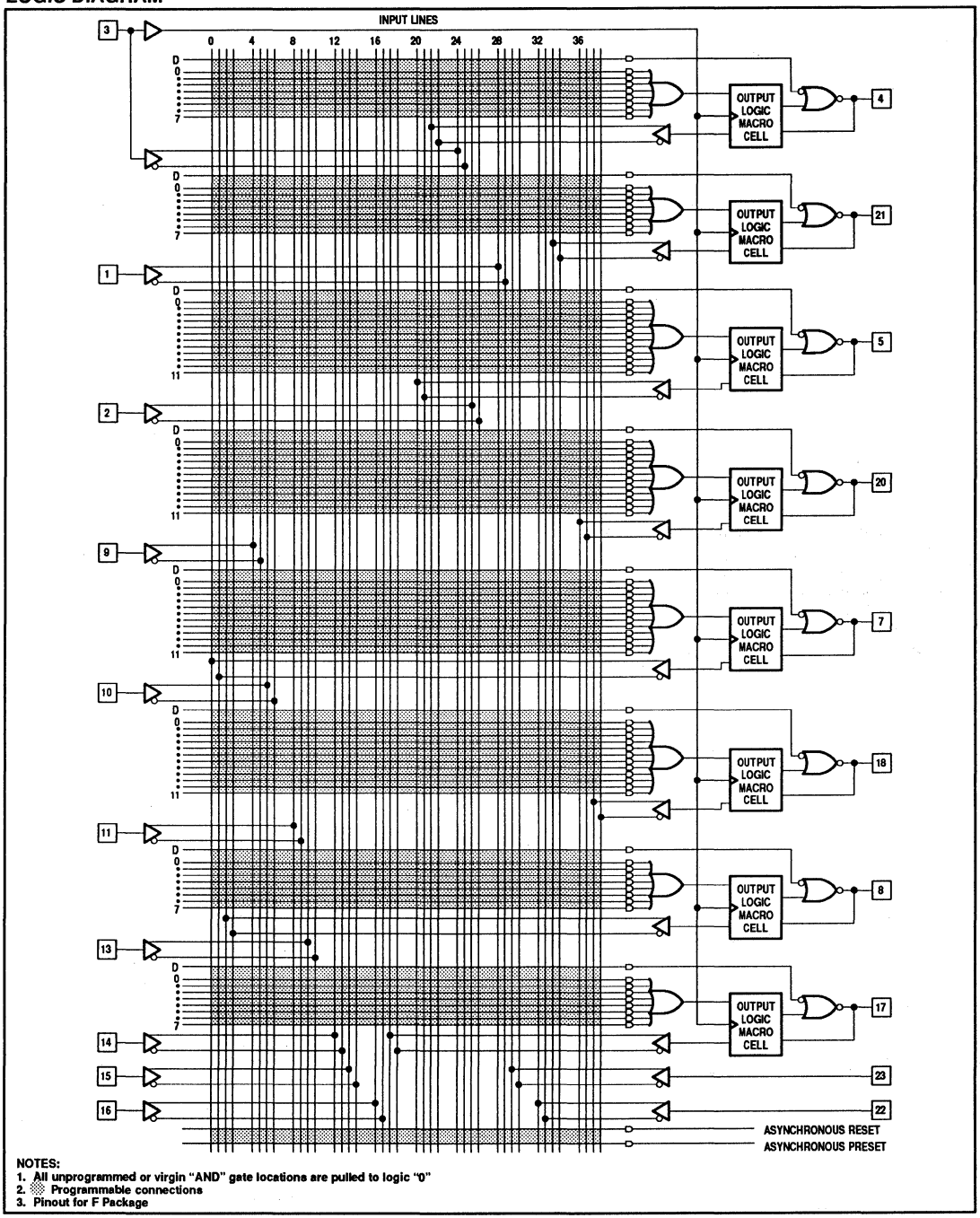
A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Ceramic Dual In-Line (300mil-wide)	10H20EV8–4F 10020EV8–4F	0586B
28-Pin Plastic Leaded Chip Carrier	10H20EV8–4A 10020EV8–4A	0401F

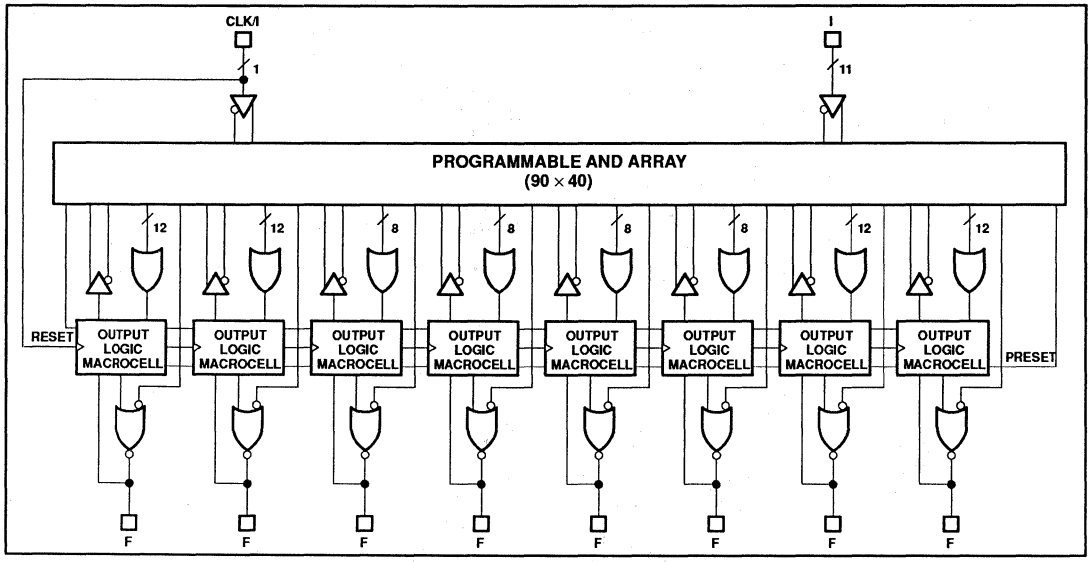
®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices, Inc.

LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections
3. Pinout for F Package

FUNCTIONAL DIAGRAM



FUNCTIONAL DESCRIPTION

The 10H20EV8/10020EV8 is an ultra high-speed universal ECL PAL-type device. Combining versatile Output Macrocells with a standard AND/OR single programmable array, this device is ideal in implementing a user's custom logic.

As can be seen in the Logic Diagram, the device is a two-level logic element with a programmable AND array. The 20EV8 can have up to 20 inputs and 8 outputs. Each output has a versatile Macrocell whereby the output can either be configured as a dedicated input, a dedicated combinatorial output with polarity control, a bidirectional I/O, or as a registered output that has both output polarity control and feedback into the AND array.

The device also features 90 product terms. Two of the product terms can be used for a global asynchronous preset and/or reset. Eight of the product terms can be used for individual output enable control of each Macrocell. The other 80 product terms are distributed among the outputs. Four of the outputs have eight product terms, while the other four have 12. This arrangement allows the utmost in flexibility when implementing user patterns.

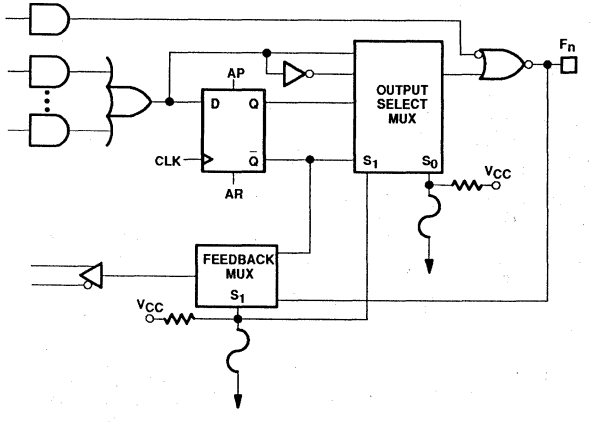


Figure 1. Output Logic Macrocell

Output Logic Macrocell

The 10H20EV8/10020EV8 incorporates an extremely versatile Output Logic Macrocell that allows the user complete flexibility when configuring outputs.

As seen in Figure 1, the 10H20EV8/10020EV8 Output Logic Macrocell consists of an edge-triggered D-type flip-flop, an output select MUX, and a feedback select MUX. Fuses S_0 and S_1 allow the user to select between the various cells. S_1 controls whether the output will be either registered with internal feedback or combinatorial I/O. S_0 controls the polarity of the output (Active-HIGH or Active-LOW). This allows the user to achieve the following configurations: Registered Active-HIGH output, Registered Active-LOW output, Combinatorial Active-HIGH output, and Combinatorial Active-LOW output. With the output enable product term, this list can be extended by adding the configurations of a Combinatorial I/O with Polarity or another input.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER		RATING	UNIT
V_{EE}	Supply voltage		–8.0	V
V_{IN}	Input voltage (V_{IN} should never be more negative than V_{EE})		0 to V_{EE}	V
I_O	Output source current		–50	mA
T_S	Operating Temperature range		–55 to +150	°C
T_J	Storage Temperature range	Ceramic Package	+165	°C
		Plastic Package	+150	°C

NOTE:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC OPERATING CONDITIONS 10H20EV8

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	NOM	MAX	
V_{CC}, V_{CO1}, V_{CO2}	Circuit ground		0	0	0	V
V_{EE}	Supply voltage (negative)			–5.2		V
V_{IH}	High level input voltage	T_{amb} = 0°C	–1170		–840	mV
		T_{amb} = +25°C	–1130		–810	mV
		T_{amb} = +75°C	–1070		–735	mV
V_{IL}	Low level input voltage	T_{amb} = 0°C	–1950		–1480	mV
		T_{amb} = +25°C	–1950		–1480	mV
		T_{amb} = +75°C	–1980		–1450	mV
T_{amb}	Operating ambient temperature range		0	+25	+75	°C

NOTE:
When operating at other than the specified V_{EE} voltage (–5.2V), the DC and AC Electrical Characteristics will vary slightly from specified values.

DC OPERATING CONDITIONS 10020EV8

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	NOM	MAX	
V_{CC}, V_{CO1}, V_{CO2}	Circuit ground		0	0	0	V
V_{EE}	Supply voltage		–4.8	–4.5	–4.2	V
V_{EE}	Supply voltage when opetating with the 10K or 10KH ECL family		–5.7			V
V_{IH}	High level input voltage	V_{EE} = –4.2V	–1150		–880	mV
		V_{EE} = –4.5V	–1165			
		V_{EE} = –4.8V	–1165			
V_{IL}	Low level input voltage	V_{EE} = –4.2V	–1810		–1475	mV
		V_{EE} = –4.5V			–1475	mV
		V_{EE} = –4.8V			–1490	mV
T_{amb}	Operating ambient temperature range		0	+25	+85	°C

NOTE:
When operating at other than the specified V_{EE} voltages (–4.2V, –4.5V, –4.8V), the DC and AC Electrical Characteristics will vary slightly from their specified values.

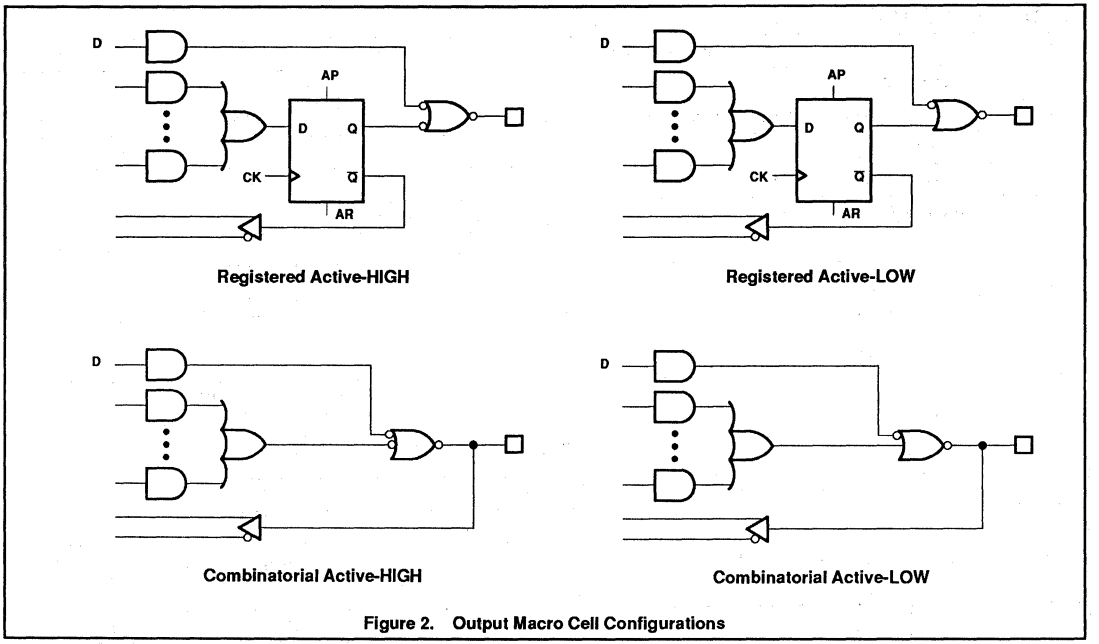


Figure 2. Output Macro Cell Configurations

OUTPUT MACRO CELL CONFIGURATION

Shown in Figure 2 are the four possible configurations of the output macrocell using fuses S_0 and S_1. As seen, the output can either be registered Active-HIGH/LOW with feedback or combinatorial Active-HIGH/LOW with feedback. If the registered mode is chosen, the feedback from the $\overline{Q}$ output to the AND array enables one to make state machines or shift registers without having to tie the output to one of the inputs. If a combinatorial output is chosen, the feedback gate is enabled from the pin and allows one to create permanent outputs, permanent inputs, or I/O pins through the use of the output enable (D) product term.

OUTPUT ENABLE

Each output on the 10H20EV8/10020EV8 has its own individual product term for output enable. The use of the D product term (direction control) allows the user three possible configurations of the outputs. They are: always enabled, always disabled, and controlled by a programmed pattern. A HIGH on the D term enables the output, while a LOW performs the disable function. Output enable control can be achieved by programming a pattern on the D term.

The output enable control can also be used to expand a designer's possibilities once a combinatorial output has been chosen. If the D term is always HIGH, the pin becomes a permanent Active-HIGH/LOW output. If the D term is always LOW (all fuses left intact), the pin now becomes an extra input.

PRESET AND RESET

The 10H20EV8/10020EV8 also includes a separate product term for asynchronous Preset and asynchronous Reset. These lines are common for all registers and are asserted when the specific product term goes HIGH. Being asynchronous, they are independent of the clock. It should be noted that the actual state of the output is dependent on how the polarity of the particular output has been chosen. If the outputs are a mix of Active-HIGH and Active-LOW, a Preset signal will force the Active-HIGH outputs HIGH while the Active-LOW outputs would go LOW, even though the Q output of all flip-flops would go HIGH. A Reset signal would force the opposite conditions.

PRELOAD

To simplify testing, the 10H20EV8/10020EV8 has also included PRELOAD circuitry. This allows a user to load any particular data desired into the registers regardless of the programmed pattern. This means that the PRELOAD can be done on a blank part and after that same part has been programmed to facilitate any post-fuse testing desired.

It can also be used by a designer to help debug a circuit. This could be important if a state machine was implemented in the 10H20EV8/ 10020EV8. The PRELOAD would allow the entry of any state in the sequence desired and start clocking from that particular point. Any or all transitions could be verified.

DC ELECTRICAL CHARACTERISTICS 10H20EV8

$0°C \le T_{amb} \le +75°C$, $V_{EE} = -5.2V \pm 5\%$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

SYMBOL	PARAMETER[1]	TEST CONDITIONS[2]	T_{amb}	LIMITS[4] MIN	LIMITS[4] MAX	UNITS
V_{OH}	High level output voltage	$V_{IN} = V_{IH}$ MIN or V_{IL} MAX	0°C +25°C +75°C	–1020 –980 –920	–840 –810 –735	mV
V_{OL}	Low level output voltage	$V_{IN} = V_{IH}$ MIN or V_{IL} MAX	0°C +25°C +75°C	–1950 –1950 –1950	–1630 –1630 –1600	mV
I_{IH}	High level input current	$V_{IN} = V_{IH}$ MAX	0°C +75°C		220	µA
I_{IL}	Low level input current	$V_{IN} = V_{IL}$ MIN Except I/O Pins	0°C +75°C	0.3		µA
$-I_{EE}$	Supply current	V_{EE} = MAX All inputs = V_{IH} MAX	0°C to +75°C		250	mA

DC ELECTRICAL CHARACTERISTICS 10020EV8

$0°C \le T_{amb} \le +85°C$, $-4.8V \le V_{EE} \le -4.2V$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

SYMBOL	PARAMETER[1]	TEST CONDITIONS[2]			LIMITS[4] MIN	LIMITS[4] TYP	LIMITS[4] MAX	UNITS
V_{OH}	High level output voltage	Outputs Loaded with 50Ω to –2.0V ± 0.010V	$V_{IN} = V_{IH}$ MAX or V_{IL} MIN	$V_{EE} = -4.2V$	–1020		–870	mV
				$V_{EE} = -4.5V$	–1025	–955	–880	mV
				$V_{EE} = -4.8V$	–1035		–880	mV
V_{OHT}	High level output threshold voltage		Apply V_{IHMIN} or V_{ILMAX} to one input at a time, other inuts at V_{IHMAX} or V_{ILMIN}.	$V_{EE} = -4.2V$	–1030			mV
				$V_{EE} = -4.5V$	–1035			mV
				$V_{EE} = -4.8V$	–1045			mV
V_{OLT}	Low level output threshold voltage		Apply V_{IHMIN} or V_{ILMAX} to one input at a time, other inuts at V_{IHMAX} or V_{ILMIN}.	$V_{EE} = -4.2V$			–1595	mV
				$V_{EE} = -4.5V$			–1610	mV
				$V_{EE} = -4.8V$			–1610	mV
V_{OL}	Low level output voltage		Inuts at V_{IHMAX} or V_{ILMIN}.	$V_{EE} = -4.2V$	–1810		–1605	mV
				$V_{EE} = -4.5V$	–1810	–1705	–1620	mV
				$V_{EE} = -4.8V$	–1830		–1620	mV
I_{IH}	High level input current	One input under test at V_{IHMAX}. Other inputs at V_{ILMIN}.					220	µA
I_{IL}	Low level input current	One input under test at V_{ILMIN}. Other inputs at V_{IHMAX}.			0.5			µA
$-I_{EE}$	V_{EE} supply current	All inputs at V_{IHMAX}.					230	mA

NOTES:

1. All voltage measurements are referenced to the ground terminal.
2. Each ECL 10KH/100K series device has been designed to meet the DC specification after thermal equilibrium has been established. Thermal equilibrium is established by applying power for at least 2 minutes, while maintaining transverse airflow of 2.5 meters/sec (500 linear feet/min.) over the device, mounted either in a test socket or on a printed circuit board. Test voltage values are given in the DC operating conditions table. Conditions for testing shown in the tables are not necessarily worst case. For worst case testing guidelines, refer to DC Testing, Chapter 1, Section 3, of the *Philips Semiconductors 10/100K ECL Data Handbook*.
3. Terminals not specifically referenced can be left electrically open. Open inputs assume a logic LOW state. Any unused pins can be terminated to –2V. If tied to V_{EE}, it must be through a resistor > 10K. It is recommended that pins that have been programmed as RESET, PRESET, or CLOCK inputs not be left open due to the possibility of false triggering from internally and externally generated switching transients.
4. The specified limits represent the worst case values for the parameter. Since these worst case values normally occur at the supply voltage and temperature extremes, additional noise immunity can be achieved by decreasing the allowable operating condition ranges.

AC ELECTRICAL CHARACTERISTICS (for Ceramic Dual In-Line Package)

10H20EV8: $0°C \leq T_{amb} \leq +75°C$, $V_{EE} = -5.2V \pm 5\%$, $V_{CC} = V_{CO1} = V_{CO2} = GND$
10020EV8: $0°C \leq T_{amb} \leq +85°C$, $-4.8V \leq V_{EE} \leq -4.2V$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

SYMBOL	PARAMETER	FROM	TO	LIMITS[1]									UNIT
				0°C			+25°C			+75°C/+85°C			
				MIN[2]	TYP[3]	MAX[2]	MIN[2]	TYP[3]	MAX[2]	MIN[2]	TYP[3]	MAX[2]	
Pulse Width													
t_{CKH}	Clock High	CLK +	CLK –	2.0	0.6		2.0	0.6		2.0	0.6		ns
t_{CKL}	Clock Low	CLK –	CLK +	2.0	0.9		2.0	0.9		2.0	0.9		ns
t_{CKP}	Clock Period	CLK +	CLK +	4.0			4.0			4.0			ns
t_{PRH}	Preset/Reset Pulse	(I, I/O) ±	(I, I/O) ±	4.5	—		4.5	—		4.5	—		ns
Setup and Hold Time													
t_{IS}	Input	(I, I/O) ±	CLK +	2.6	1.0		2.6	1.1		2.7	1.4		ns
t_{IH}	Input	CLK +	(I, I/O) ±	0.1	< 0		0.1	< 0		0.1	< 0		ns
t_{PRS}	Clock Resume after Preset/Reset	(I, I/O) ±	CLK +	4.6	1.0		4.6	0.9		4.6	0.8		ns
Propagation Delay													
t_{PD}	Input	(I, I/O) ±	I/O ±		2.85	4.7		2.95	4.7		3.35	4.7	ns
t_{CKO}	Clock	CLK +	I/O ±		1.65	2.4		1.7	2.4		2.0	2.5	ns
t_{OE}	Output Enable	(I, I/O) ±	I/O		2.0	4.2		2.1	4.2		2.2	4.2	ns
t_{OD}	Output Disable	(I, I/O) ±	I/O		2.0	4.2		2.1	4.2		2.2	4.2	ns
t_{PRO}	Preset/Reset	(I, I/O) ±	I/O ±		2.8	4.7		3.0	4.7		3.5	4.7	ns
t_{PPR}	Power-on Reset	V_{EE}	I/O		—	10		—	10		—	10	ns
f_{MAX}				212	377		212	357		204	294		MHz

NOTES:
1. Refer to AC Test Circuit and Voltage Wafeforms diagrams.
2. Maximum loading conditions: 89 fuses intact per row.
3. Typical loading conditions: 15 fuses intact per row. (All "inactive" fuses, except those necessary for correct functionality, are removed.)

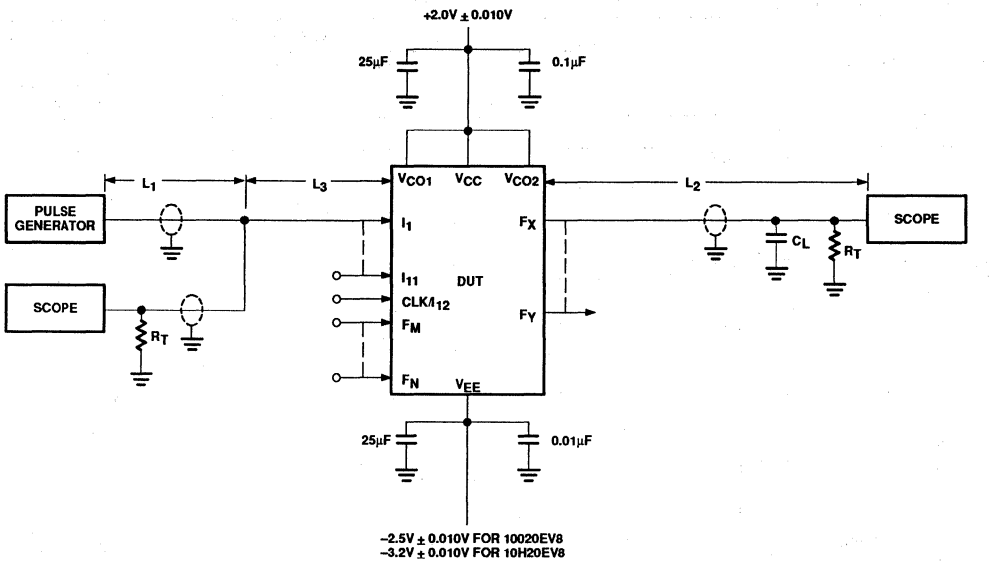
AC ELECTRICAL CHARACTERISTICS (for Plastic Leaded Chip Carrier)

10H20EV8: $0°C \leq T_{amb} \leq +75°C$, $V_{EE} = -5.2V \pm 5\%$, $V_{CC} = V_{CO1} = V_{CO2} = GND$
10020EV8: $0°C \leq T_{amb} \leq +85°C$, $-4.8V \leq V_{EE} \leq -4.2V$, $V_{CC} = V_{CO1} = V_{CO2} = GND$

SYMBOL	PARAMETER	FROM	TO	LIMITS[1]									UNIT
				0°C			+25°C			+75°C/+85°C			
				MIN	TYP[3]	MAX[2]	MIN	TYP[3]	MAX[2]	MIN	TYP[3]	MAX[2]	
Pulse Width													
t_{CKH}	Clock High	CLK +	CLK –	2.0	0.6		2.0	0.6		2.0	0.6		ns
t_{CKL}	Clock Low	CLK –	CLK +	2.0	0.9		2.0	0.9		2.0	0.9		ns
t_{CKP}	Clock Period	CLK +	CLK +	4.0			4.0			4.0			ns
t_{PRH}	Preset/Reset Pulse	(I, I/O) ±	(I, I/O) ±	4.5	—		4.5	—		4.5	—		ns
Setup and Hold Time													
t_{IS}	Input	(I, I/O) ±	CLK +	2.5	1.0		2.5	1.1		2.6	1.4		ns
t_{IH}	Input	CLK +	(I, I/O) ±	0	< 0		0	< 0		0	< 0		ns
t_{PRS}	Clock Resume after Preset/Reset	(I, I/O) ±	CLK +	4.5	1.0		4.5	0.9		4.5	0.8		ns
Propagation Delay													
t_{PD}	Input	(I, I/O) ±	I/O ±		2.85	4.5		2.95	4.5		3.35	4.5	ns
t_{CKO}	Clock	CLK +	I/O ±		1.65	2.2		1.7	2.2		2.0	2.3	ns
t_{OE}	Output Enable	(I, I/O) ±	I/O		2.0	4.0		2.1	4.0		2.2	4.0	ns
t_{OD}	Output Disable	(I, I/O) ±	I/O		2.0	4.0		2.1	4.0		2.2	4.0	ns
t_{PRO}	Preset/Reset	(I, I/O) ±	I/O ±		2.8	4.5		3.0	4.5		3.5	4.5	ns
t_{PPR}	Power-on Reset	V_{EE}	I/O		—	10		—	10		—	10	ns
f_{MAX}				212	377		212	357		204	294		MHz

NOTES:
1. Refer to AC Test Circuit and Voltage Wafeforms diagrams.
2. Maximum loading conditions: 89 fuses intact per row.
3. Typical loading conditions: 15 fuses intact per row. (All "inactive" fuses, except those necessary for correct functionality, are removed.)

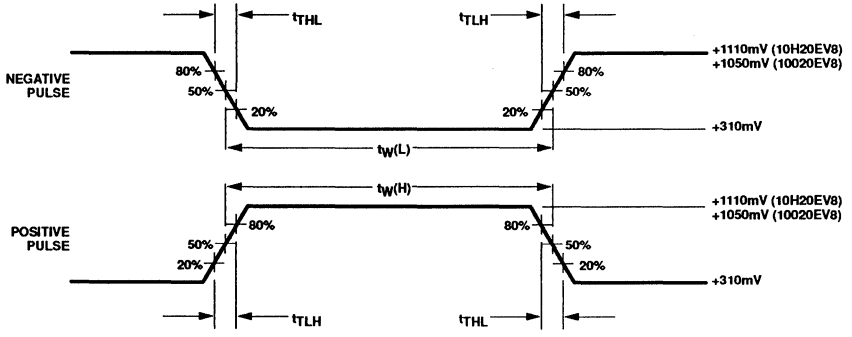
AC TEST CIRCUIT



NOTES:

1. Use decoupling capacitors of 0.1μF and 25μF from GND to V_{CC}, and 0.01μF and 25μF from GND to V_{EE} (0.01 and 0.1μF capacitors should be NPO Ceramic or MLC type). Decoupling capacitors should be placed as close as physically possible to the DUT and lead length should be kept to less than $^1/_4$ inch (6mm).
2. All unused inputs should be connected to either HIGH or LOW state consistent with the LOGIC function required.
3. All unused outputs are loaded with 50Ω to GND.
4. L_1 and L_2 are equal length 50Ω impedance lines. L_3, the distance from the DUT pin to the junction of the cable from the Pulse Generator and the cable to the Scope, should not exceed $^1/_4$ inch (6mm).
5. R_T = 50Ω terminator internal to Scope.
6. The unmatched wire stub between coaxial cable and pins under test must be less than $^1/_4$ inch (6mm) long for proper test.
7. C_L = Fixture and stray capacitance ≤ 3pF.
8. Any unterminated stubs connected anywhere along the transmission line between the Pulse Generator and the DUT or between the DUT and the Scope should not exceed $^1/_4$ inch (6mm) in length (refer to section on AC setup procedure).
9. All 50Ω resistors should have tolerance of ± 1% or better.
10. Test procedures are shown for only one input or set of input conditions. Other inputs are tested in the same manner.

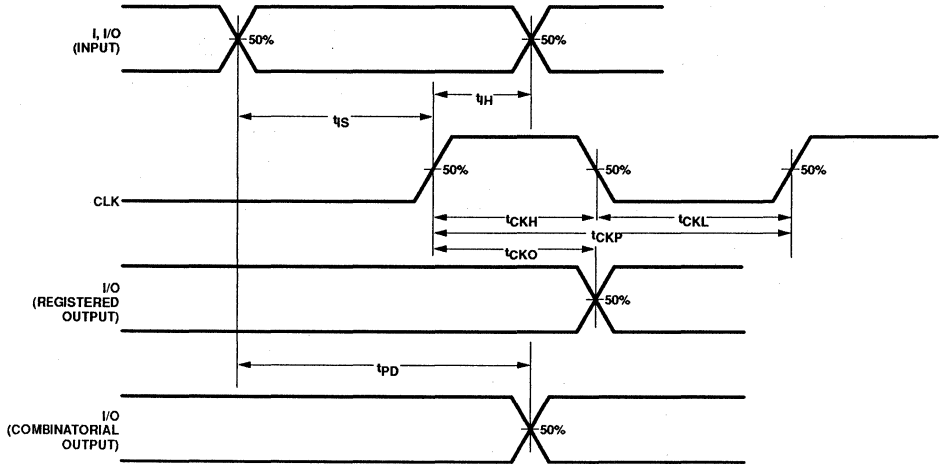
VOLTAGE WAVEFORMS



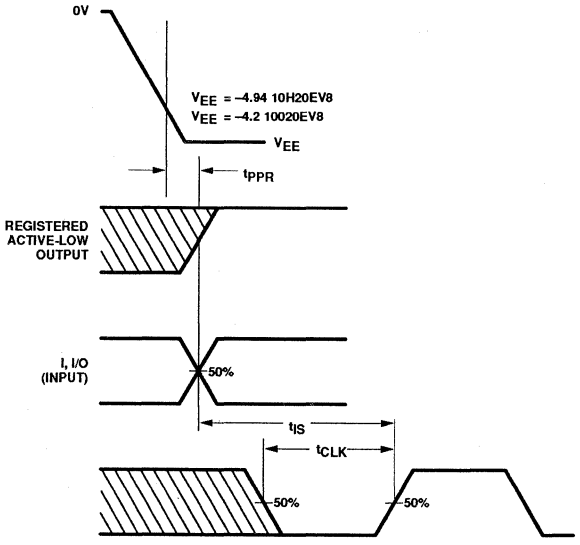
INPUT PULSE REQUIREMENTS $V_{CC} = V_{CO1} = V_{CO2} = +2.0V \pm 0.010V$, $V_{EE} = -3.2V \pm 0.010V$, V_T = GND (0V)					
FAMILY	AMPLITUDE	REP RATE	PULSE WIDTH	t_{TLH}	t_{THL}
10KH ECL	800mV$_{P-P}$	1MHz	500ns	1.3 ± 0.2ns	1.3 ± 0.2ns
INPUT PULSE REQUIREMENTS $V_{CC} = V_{CO1} = V_{CO2} = +2.0V \pm 0.010V$, $V_{EE} = -2.5V \pm 0.010V$, V_T = GND (0V)					
FAMILY	AMPLITUDE	REP RATE	PULSE WIDTH	t_{TLH}	t_{THL}
100K ECL	740mV$_{P-P}$	1MHz	500ns	0.7 ± 0.1ns	0.7 ± 0.1ns

Input Pulse Definition

TIMING DIAGRAMS

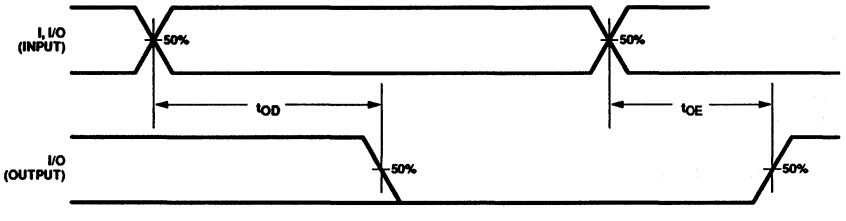


Flip-Flop and Gate Outputs

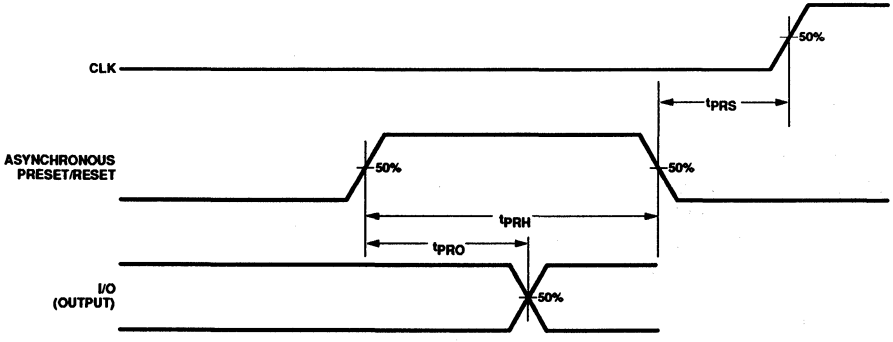


Power-On Reset

TIMING DIAGRAMS (Continued)



Output Enable/Disable

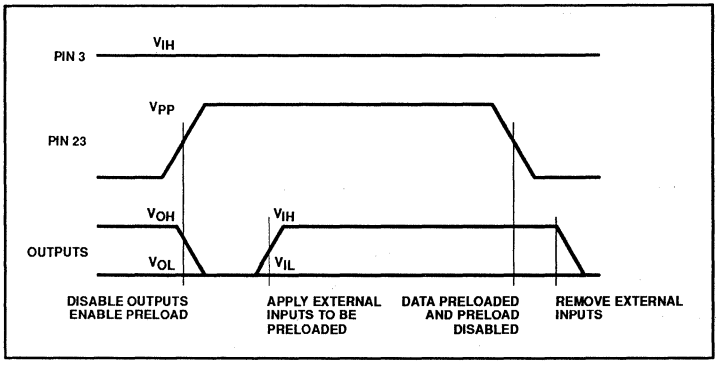


Asynchronous Preset/Reset

REGISTER PRELOAD

The 10H20EV8/10020EV8 has included circuitry that allows a user to load data into the output registers. Register PRELOAD can be done at any time and is not dependent on any particular pattern programmed into the device. This simplifies the ability to fully verify logic states and sequences even after the device has been patterned.

The pin levels and sequence necessary to perform the register PRELOAD are shown below.



SYMBOL	PARAMETER	LIMITS			UNIT
		MIN	TYP	MAX	
V_{IH}	Input HIGH level during PRELOAD and Verify	–1.1	–0.9	–0.7	V
V_{IL}	Input LOW level during PRELOAD and Verify	–1.85	–1.65	–1.45	V
V_{PP}	PRELOAD enable voltage applied to I_{11}	1.45	1.6	1.75	V

NOTE:

1. Unused inputs should be handled as follows:
   - Set at V_{IH} or V_{IL}
   - Terminated to –2V
   - Tied to V_{EE} through a resistor > 10K
   - Open

LOGIC PROGRAMMING

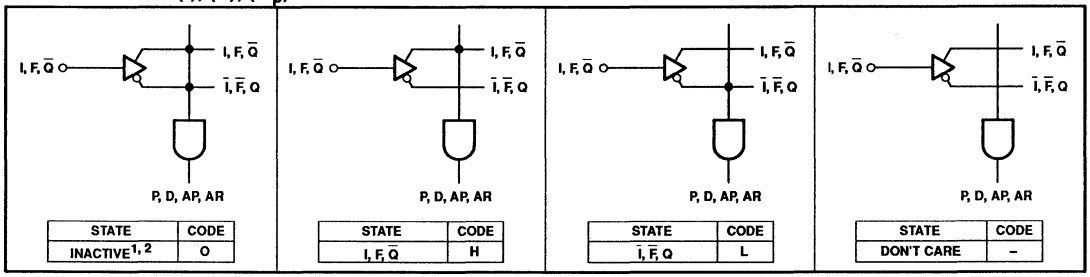
The 10H20EV8/10020EV8 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the 10H20EV8/10020EV8.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

10H20EV8/10020EV8 logic designs can also be generated using the program table entry format detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, F, Q, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

"AND" ARRAY – (I), (F), ($\overline{Q}_p$)



STATE	CODE
INACTIVE[1, 2]	0

STATE	CODE
I, F, Q̄	H

STATE	CODE
Ī, F̄, Q	L

STATE	CODE
DON'T CARE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate (P, D, AP, AR) will be unconditionally inhibited if any one of the I, F or Q link pairs is left intact.

OUTPUT MACROCELL CONFIGURATIONS

OUTPUT MACROCELL CONFIGURATION	CONTROL WORD FUSE	POLARITY FUSE
Registered Output, Active-HIGH	D	H
Registered Output, Active-LOW	D[1]	L[1]
Combinatorial I/O, Active-HIGH	B	H
Combinatorial I/O, Acitve-LOW	B	L

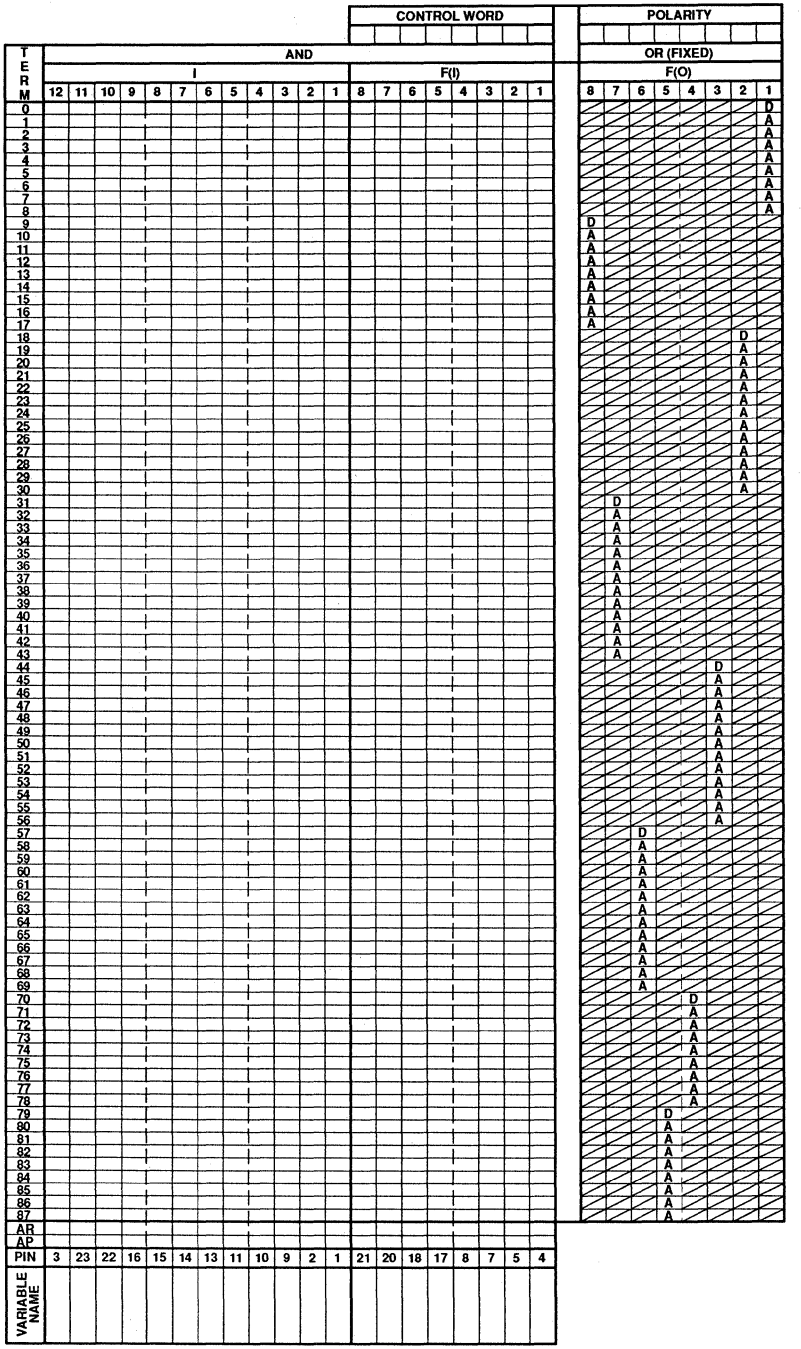
NOTE:
1. This is the initial (unprogrammed) state of the device.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of the 1992 PLD Data Handbook for additional information.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

PROGRAM TABLE



SNAP

Features

- Schematic entry using DASH™ 4.0 or above or OrCAD™ SDT III
- State Equation Entry
- Boolean Equation Entry
- Allows design entry in any combination of above formats
- Simulator
  - Logic and fault simulation
  - Timing model generation for device timing simulation
  - Synthetic logic analyzer format
- Macro library for standard TTL and user defined functions
- Device independent netlist generation
- JEDEC fuse map generated from netlist

SNAP (Synthesis, Netlist, Analysis and Program) is a versatile development tool that speeds the design and testing of PML. SNAP combines a user-friendly environment and powerful modules that make designing with PML simple. The SNAP environment gives the user the freedom to design independent of the device architecture.

The flexibility in the variations of design entry methodologies allows design entry in the most appropriate terms. SNAP merges the inputs, regardless of the type, into a high-level netlist for simulation or compilation into a JEDEC fuse map. The JEDEC fuse map can then be transferred from the host computer to the device programer.

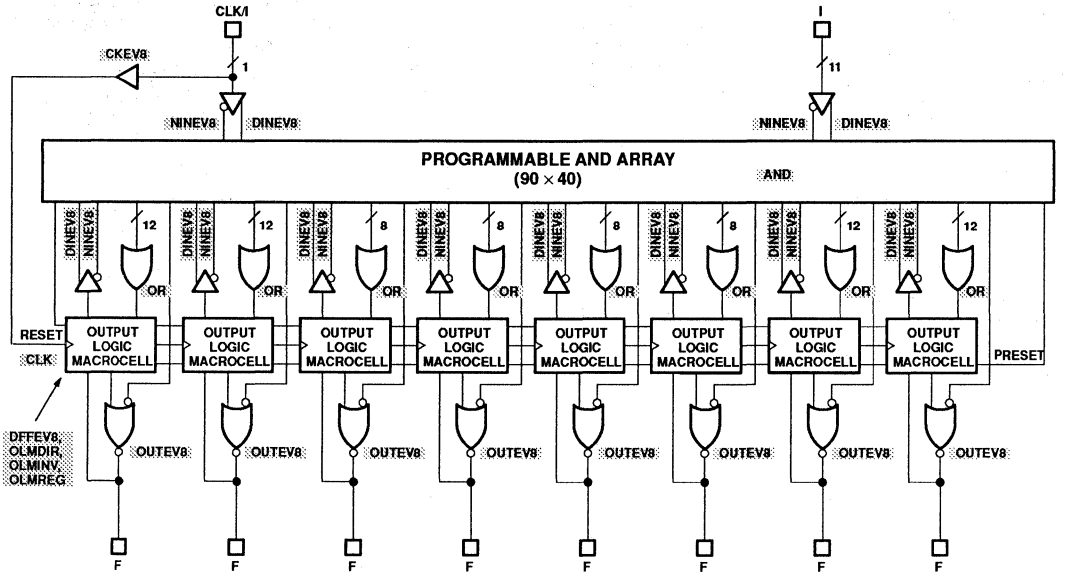
SNAP's simulator uses a synthetic logic analyzer format to display and set the nodes of the design. The SNAP simulator provides complete timing information, setup and hold-time checking, plus toggle and fault grading analysis.

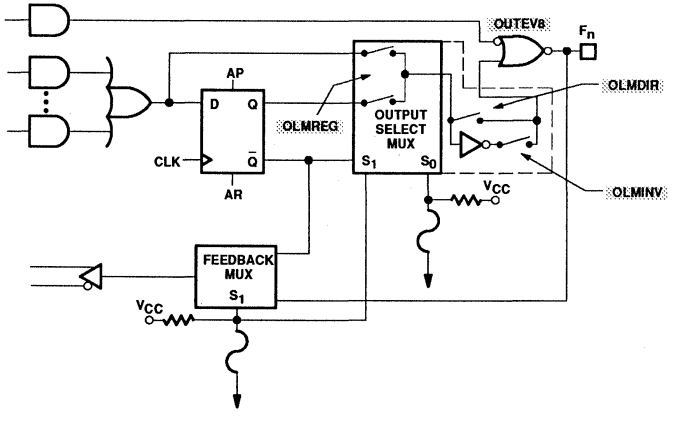
SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. A minimum of 640K bytes of RAM is required together with a hard disk.

DESIGN SECURITY

The 10H20EV8/10020EV8 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

SNAP RESOURCE SUMMARY DESIGNATIONS





Output Logic Macrocell

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

DESCRIPTION

The PHD48N22-7 is an ultra fast Programmable High-speed Decoder featuring a 7.5ns maximum propagation delay. The architecture has been optimized using Philips Semiconductors state-of-the-art bipolar oxide isolation process coupled with titanium-tungsten fuses to achieve superior speed in any design.

The PHD48N22–7 is a two level logic element comprised of 36 fixed inputs, 73 AND gates, 10 outputs, and 12 bidirectional I/Os. This gives the device the ability to have as many as 48 inputs. Individual 3-State control of all outputs is also provided.

The device is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment. Proprietary designs can be protected by programming the security fuse.

The SLICE and SNAP software packages from Philips Components–Philips Semiconductors support easy design entry for the PHD48N22–7 as well as other PLD devices.

Order codes are listed below.

FEATURES

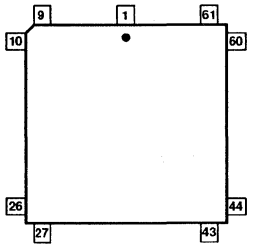
- Ideal for high speed system decoding
- Super high speed at 7.5ns t_{PD}
- 36 dedicated inputs
- 22 outputs
  - 12 bidirectional I/O
  - 10 dedicated outputs
- Security fuse to prevent duplication of proprietary designs.
- Individual 3-State control of all outputs
- Field-programmable on industry standard programmers
- Available in 68-Pin Plastic Leaded Chip Carrier (PLCC)

APPLICATIONS

- High speed memory decoders
- High speed code detectors
- Random logic
- Peripheral selectors
- Machine state decoders

PIN CONFIGURATION

A Package



A = Plastic Leaded Chip Carrier

Pin	Function	Pin	Function
1	I0	35	I29
2	I1	36	I30
3	I2	37	I31
4	V_{CC3}	38	V_{CC4}
5	I3	39	I32
6	I4	40	I33
7	I5	41	I34
8	I6	42	I35
9	I7	43	B0
10	I8	44	B1
11	I9	45	B2
12	I10	46	B3
13	I11	47	B4
14	GND5	48	GND6
15	GND1	49	GND2
16	I12	50	B5
17	I13	51	B6
18	I14	52	B7
19	I15	53	O0
20	I16	54	O1
21	V_{CC2}	55	V_{CC1}
22	I17	56	O2
23	I18	57	O3
24	I19	58	O4
25	I20	59	O5
26	I21	60	O6
27	I22	61	O7
28	I23	62	O8
29	I24	63	O9
30	I25	64	OR0
31	I26	65	OR1
32	GND3	66	GND4
33	I27	67	OR2
34	I28	68	OR3

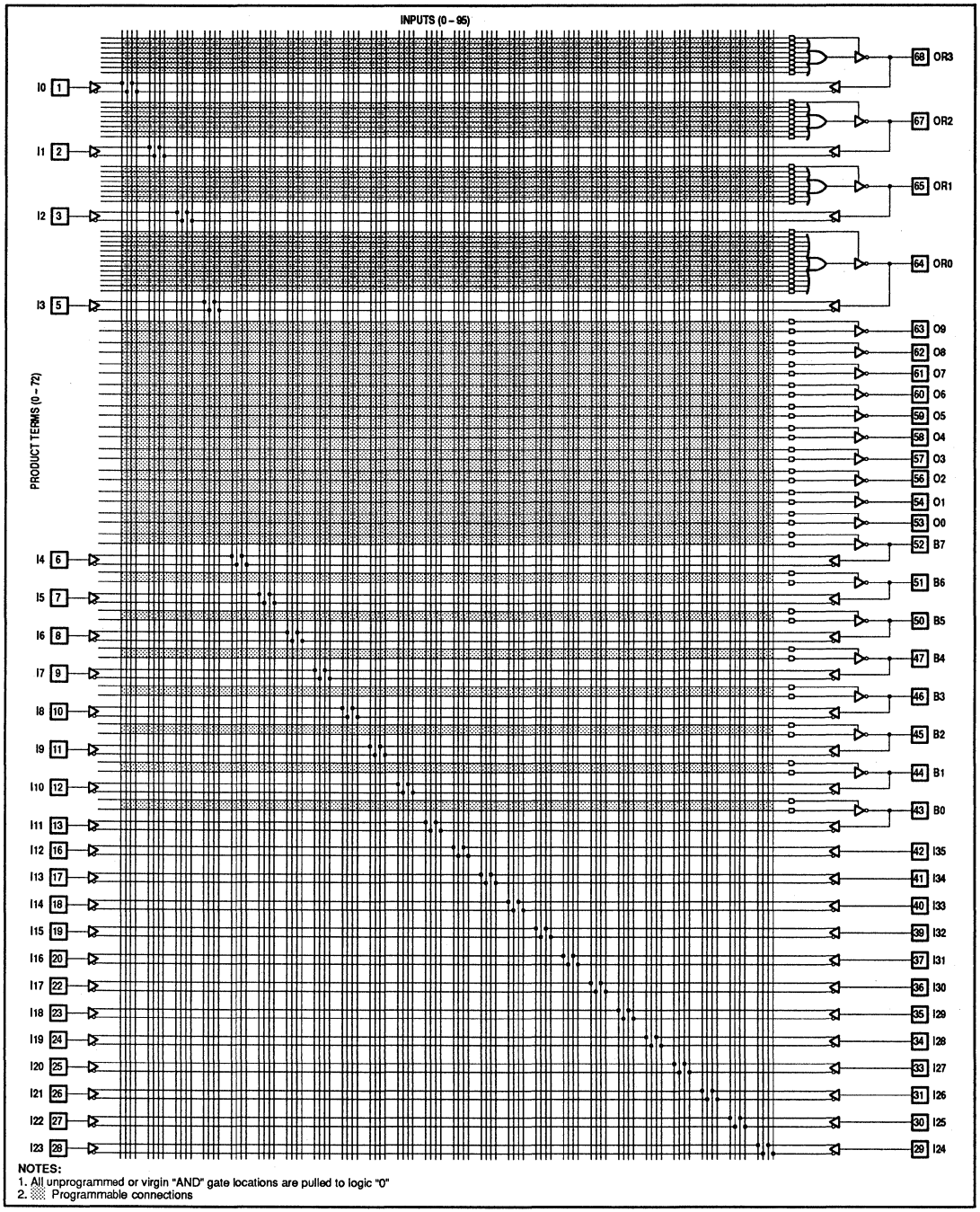
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
68-Pin Plastic Leaded Chip Carrier	PHD48N22–7A	0398E

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

LOGIC DIAGRAM

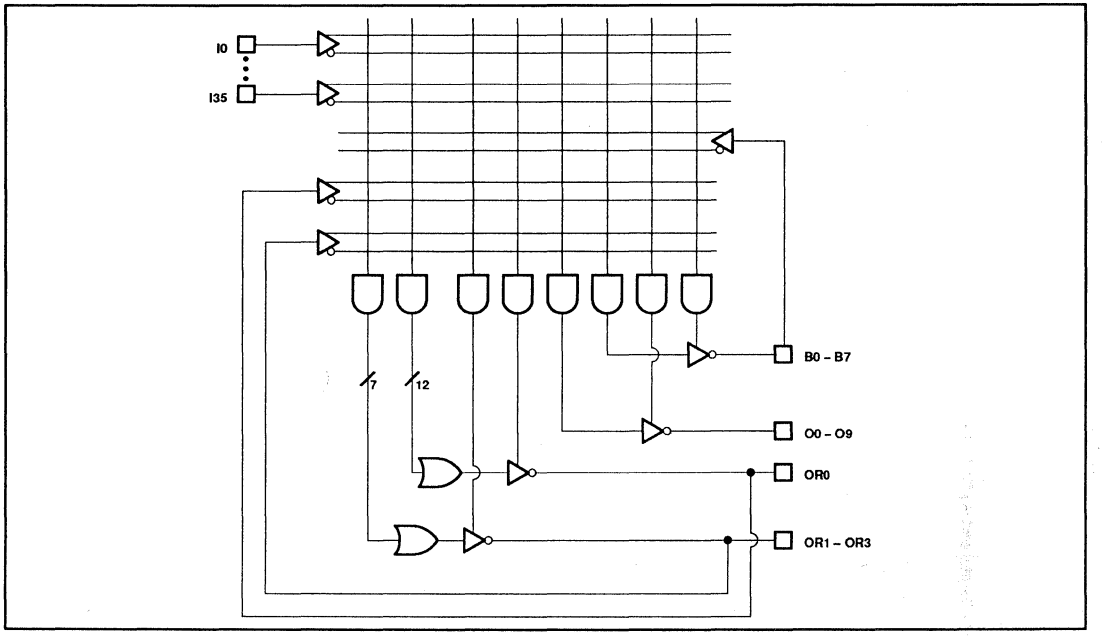


NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	–0.5	+7	V_{DC}
V_{IN}	Input voltage	–0.5	+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

OPERATING RANGES

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage	+4.75	+5.25	V_{DC}
T_{amb}	Operating free-air temperature	0	+75	°C

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –18mA		–0.8	–1.5	V
Output voltage						
		V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}				
V_{OL}	Low	I_{OL} = +24mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = +0.40V		–20	–250	µA
I_{IH}	High	V_{IN} = +2.7V			25	µA
I_I	High	V_{IN} = V_{CC} = $V_{CC\ MAX}$			100	µA
Output current						
		V_{CC} = MAX				
I_{OZH}	Output leakage [3]	V_{OUT} = +2.7V			100	µA
I_{OZL}	Output leakage [3]	V_{OUT} = +0.40V			–100	µA
I_{OS}	Short circuit [4]	V_{OUT} = +0V	–30	–60	–90	mA
I_{CC}	V_{CC} current	V_{CC} = MAX			420	mA
Capacitance[5]						
		V_{CC} = +5V				
C_{IN}	Input	V_{IN} = 2.0V @ f = 1MHz		8		pF
C_{OUT}	I/O	V_{OUT} = 2.0V @ f = 1MHz		8		pF

NOTES:
1. Typical limits are at V_{CC} = 5.0V and T_{amb} = +25°C.
2. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
3. Leakage current for bidirectional pins is the worst case of I_{IL} and I_{OZL} or I_{IH} and I_{OZL}.
4. Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second.
5. These parameters are not 100% tested, but are periodically sampled.

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 200\Omega$, $R_2 = 390\Omega$
Operating temerature at 200 CFM Minimum air flow.

SYMBOL	PARAMETER	FROM	TO	TEST CONDITIONS	LIMITS MIN	LIMITS MIN	UNIT
t_{PD1}[1]	Propagation delay through B/O outputs	(I, B, OR) ±	Output ±	$C_L = 50pF$		7.5	ns
t_{PD2}[1]	Propagation delay through OR outputs	(I, B, OR) ±	Output ±	$C_L = 50pF$		8.0	ns
t_{OE}[2]	Output Enable	(I, B, OR) ±	Output enable	$C_L = 50pF$		10	ns
t_{OD}[2]	Output Disable	(I, B, OR) ±	Output disable	$C_L = 5pF$		10	ns

NOTES:
1. $t_{PD1, 2}$ are tested with switch S_1 closed and $C_L = 50pF$.
2. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

VIRGIN STATE

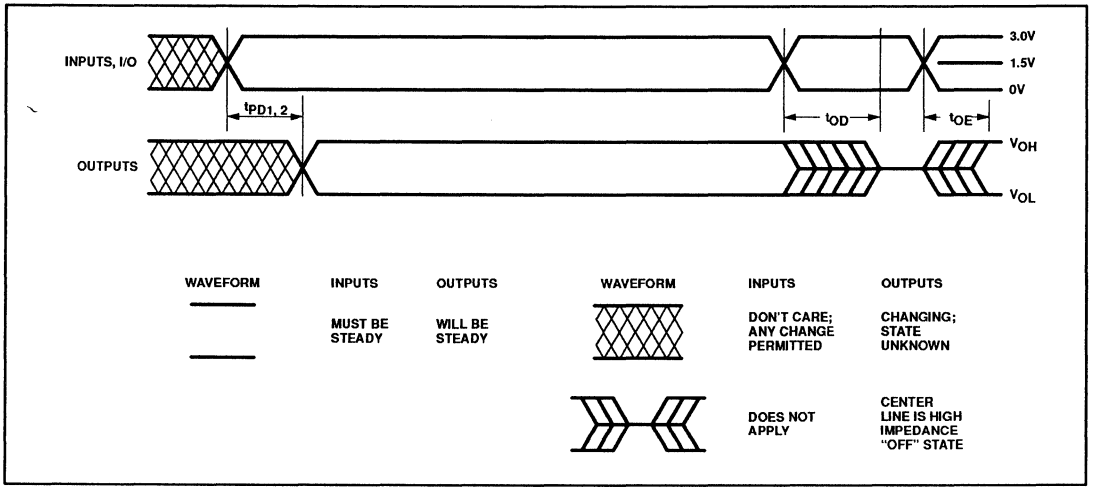
A factory shipped virgin device contains all fusible links intact, such that:

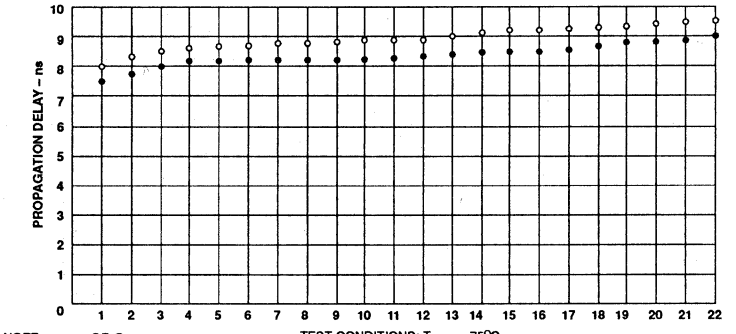
1. All outputs are disabled.
2. All p-terms are disabled in the AND array.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD1}	Input to output propagation delay (through B/O outputs).
t_{PD2}	Input to output propagation delay (through OR outputs).
t_{OD}	Input to Output Disable (3-State) delay (Output Disable).
t_{OE}	Input to Output Enable delay (Output Enable).

TIMING DIAGRAM



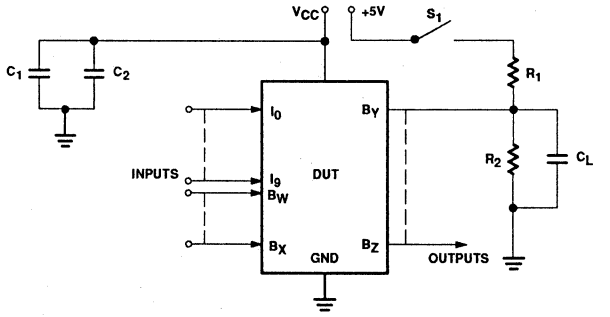


NOTE: ○ = OR Outputs
● = B/O Outputs

TEST CONDITIONS: T_{amb} = 75°C;
V_{CC} = 4.75V; C_L = 50pF;
R_1 = 200Ω; R_2 = 390Ω

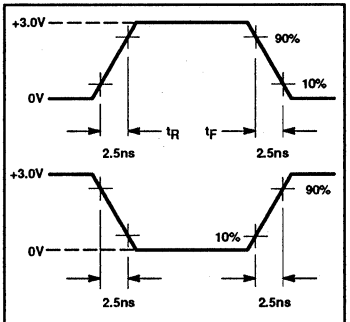
Worst-Case Propagation Delay vs. Number of Outputs Switching

AC TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

LOGIC PROGRAMMING

The PHD48N22–7 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ 90 design software packages also support the architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

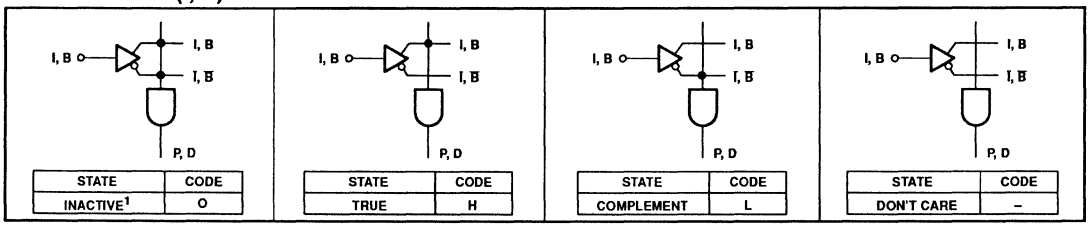
PHD48N22–7 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

To implement the desired logic functions, each logic variable (I, B, P and D) from the logic equations is assigned a symbol. TRUE (High), COMPLEMENT (Low), DON'T CARE and INACTIVE symbols are defined below.

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I, B)



NOTE:
1. This is the initial state.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable high-speed decoder logic (48 × 73 × 22)

PHD48N22–7

PROGRAM TABLE

TERM	INPUTS																																															
	I																																			B								OR			
	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0
0																																																
1																																																
2																																																
3																																																
4																																																
5																																																
6																																																
7																																																
8																																																
9																																																
10																																																
11																																																
12																																																
13																																																
14																																																
15																																																
16																																																
17																																																
18																																																
19																																																
20																																																
21																																																
22																																																
23																																																
24																																																
25																																																
26																																																
27																																																
28																																																
29																																																
30																																																
31																																																
32																																																
33																																																
34																																																
35																																																
36																																																
37																																																
38																																																
39																																																
40																																																
41																																																
42																																																
43																																																
44																																																
45																																																
46																																																
47																																																
48																																																
49																																																
50																																																
51																																																
52																																																
53																																																
54																																																
55																																																
56																																																
57																																																
58																																																
59																																																
60																																																
61																																																
62																																																
63																																																
64																																																
65																																																
66																																																
67																																																
68																																																
69																																																
70																																																
71																																																
72																																																
PIN	42	41	40	39	37	36	35	34	33	31	30	29	28	27	26	25	24	23	22	20	19	18	17	16	13	12	11	10	9	8	7	6	5	3	2	1	52	51	50	47	46	45	44	43	68	67	65	64
VARIABLE NAME																																																

Programmable high-speed decoder logic (48 × 73 × 22)

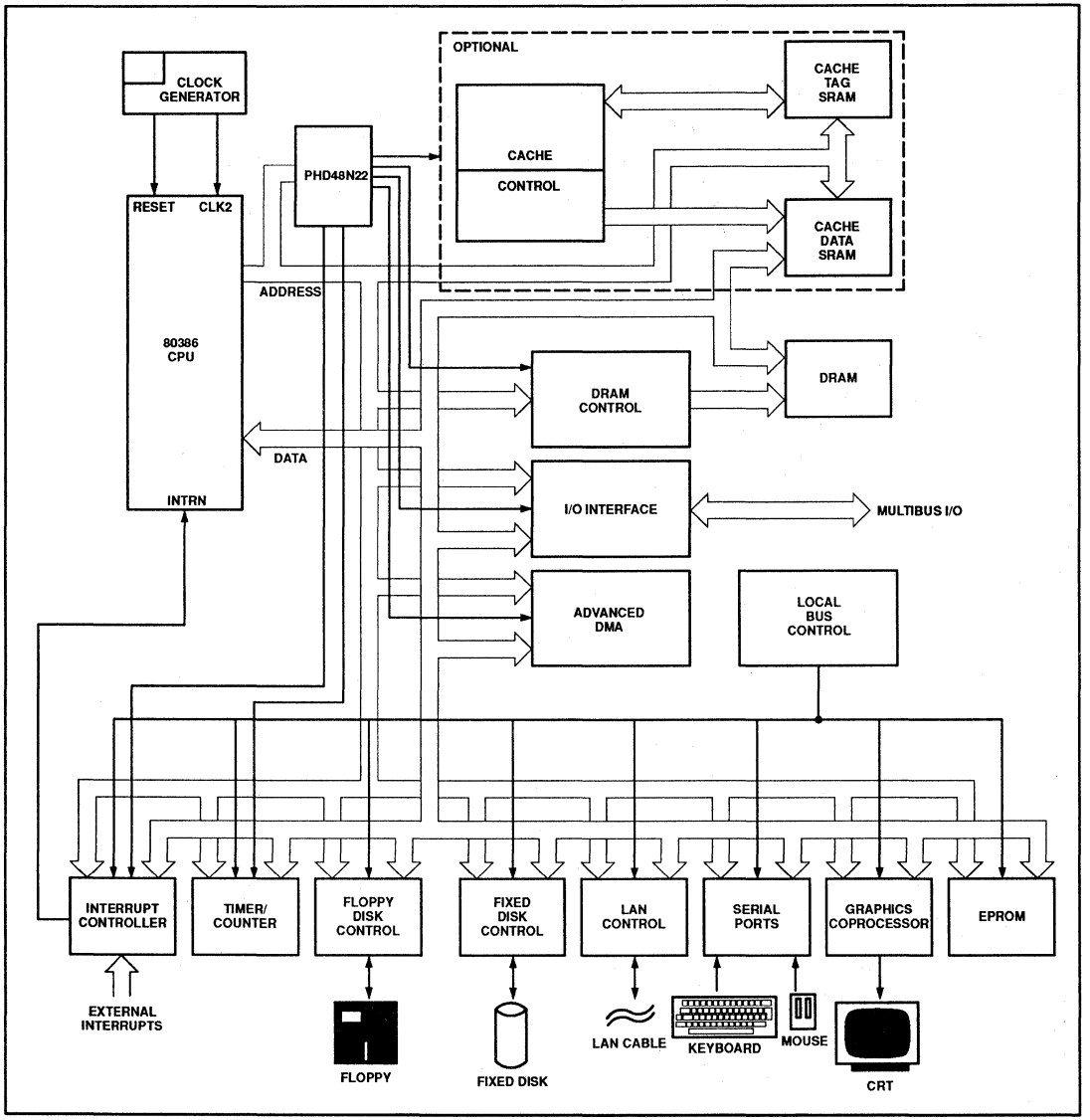
PHD48N22–7

PROGRAM TABLE (Continued)

TERM	OUTPUTS																					
	O										B								OR			
	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0
0																			D			
1																			A			
2																			A			
3																			A			
4																			A			
5																			A			
6																			A			
7																			A			
8																				D		
9																				A		
10																				A		
11																				A		
12																				A		
13																				A		
14																				A		
15																				A		
16																					D	
17																					A	
18																					A	
19																					A	
20																					A	
21																					A	
22																					A	
23																					A	
24																						D
25																						A
26																						A
27																						A
28																						A
29																						A
30																						A
31																						A
32																						A
33																						A
34																						A
35																						A
36																						A
37	D																					
38	A																					
39		D																				
40		A																				
41			D																			
42			A																			
43				D																		
44				A																		
45					D																	
46					A																	
47						D																
48						A																
49							D															
50							A															
51								D														
52								A														
53									D													
54									A													
55										D												
56										A												
57											D											
58											A											
59												D										
60												A										
61													D									
62													A									
63														D								
64														A								
65															D							
66															A							
67																D						
68																A						
69																	D					
70																	A					
71																		D				
72																		A				
PIN	63	62	61	60	59	58	57	56	54	53	52	51	50	47	46	45	44	43	68	67	65	64
VARIABLE NAME																						

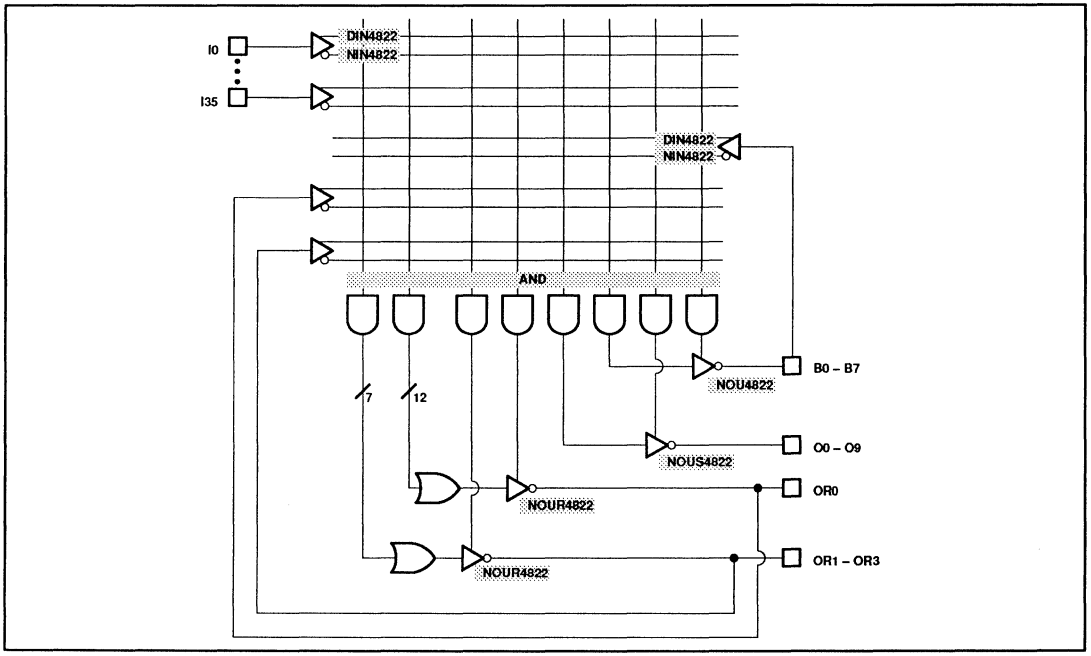
Programmable high-speed decoder logic (48 × 73 × 22) — PHD48N22–7

TYPICAL SYSTEM APPLICATION



Programmable high-speed decoder logic (48 × 73 × 22) | PHD48N22–7

SNAP RESOURCE SUMMARY DESIGNATIONS



Section 4
PLA Devices

CONTENTS

Programmable logic arrays (18 × 42 × 10)

PLS153/A

DESCRIPTION

The PLS153 and PLS153A are two-level logic elements, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On-chip T/C buffers couple either True (I, B) or Complement ($\overline{I}$, $\overline{B}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS153 and PLS153A are field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

FEATURES

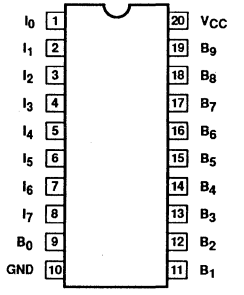
- Field-Programmable (Ni-Cr links)
- 8 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay:
  - PLS153: 40ns (max)
  - PLS153A: 30ns (max)
- Input loading: −100μA (max)
- Power dissipation: 650mW (typ)
- 3-State outputs
- TTL compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

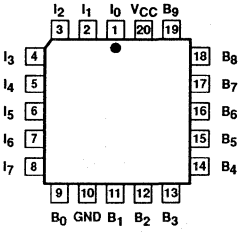
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual In-Line, 300mil-wide	PLS153N, PLS153AN	0408B
20-Pin Plastic Leaded Chip Carrier	PLS153A, PLS153AA	0400E

LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$Pn = A \cdot B \cdot C \cdot D \cdot \ldots$

TYPICAL LOGIC FUNCTION:

AT OUTPUT POLARITY = H

$Z = P0 + P1 + P2 \ldots$

AT OUTPUT POLARITY = L

$Z = \overline{P0 + P1 + P2 + \ldots}$

$Z = \overline{P0} \cdot \overline{P1} \cdot \overline{P2} \cdot \ldots$

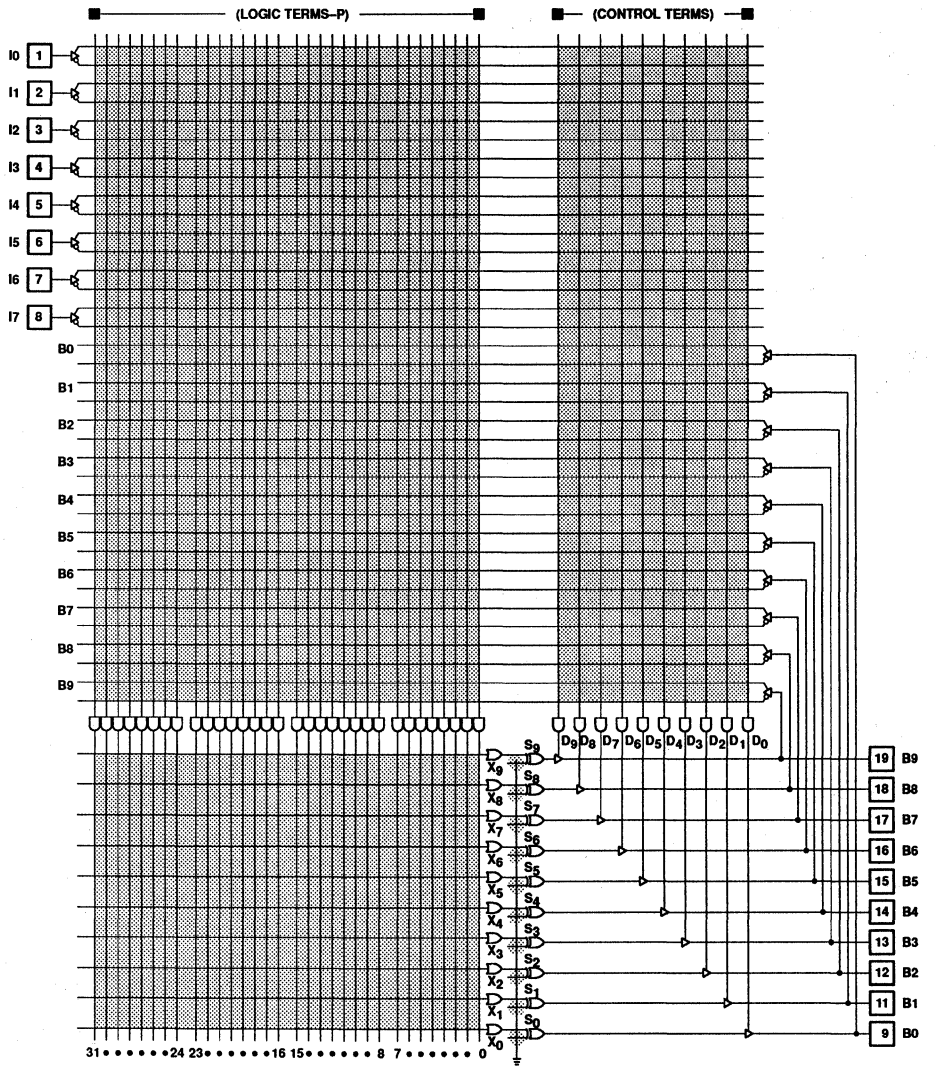
NOTES:

1. For each of the 10 outputs, either function Z (Active-High) or Z (Active-Low) is available, but not both. The desired output polarity is programmed via the Ex-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

Programmable logic arrays (18 × 42 × 10)

PLS153/A

LOGIC DIAGRAM

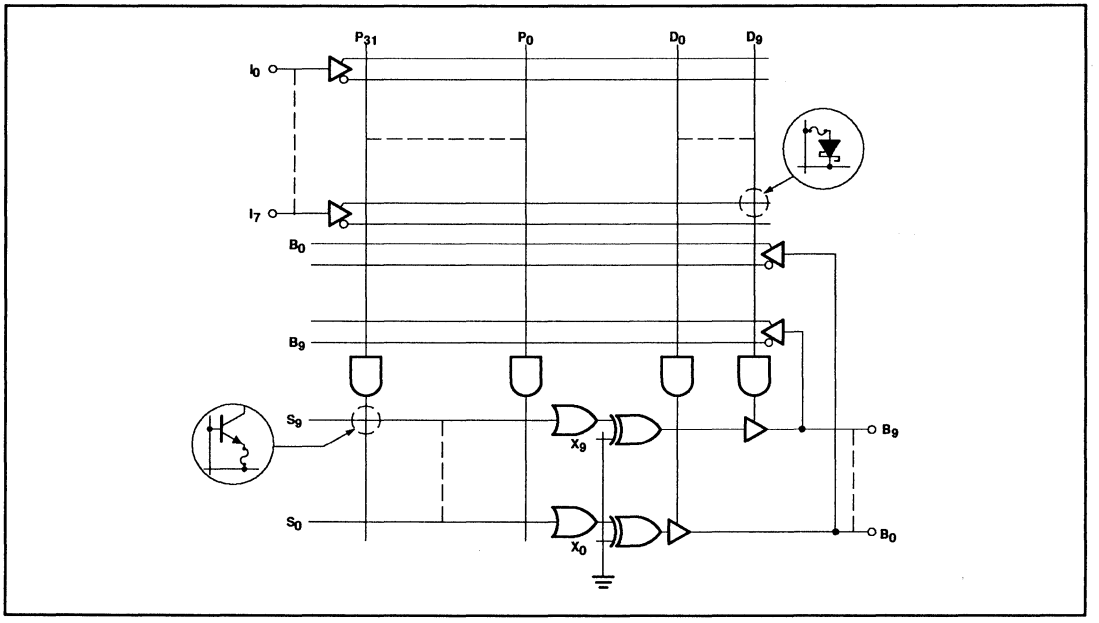


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ⁘ Programmable connection.

Programmable logic arrays (18 × 42 × 10)

PLS153/A

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS153/A devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Philips Semiconductors Military Data Handbook.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = −2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			−100	μA
I_{IH}	High	V_{IN} = 5.5V			40	μA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 5.5V			80	μA
		V_{OUT} = 0.45V			−140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		130	155	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with +10V applied to I_7.
5. Measured with +10V applied to I_{0-7}. Output sink current is supplied through a resistor to V_{CC}.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with I_0, I_1 at 0V, $I_2 - I_7$ and B_{0-9} at 4.5V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_7$).

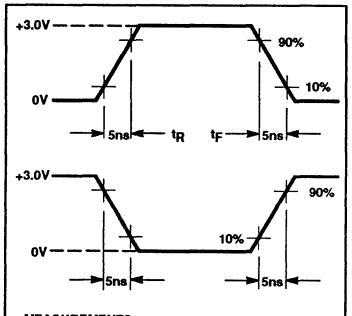
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLS153			PLS153A			
					MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
t_{PD}	Propagation delay	Input ±	Output ±	C_L = 30pF		30	40		20	30	ns
t_{OE}	Output enable[2]	Input ±	Output –	C_L = 30pF		25	35		20	30	ns
t_{OD}	Output disable[2]	Input ±	Output +	C_L = 5pF		25	35		20	30	ns

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays are measured and specified under worst case conditions.

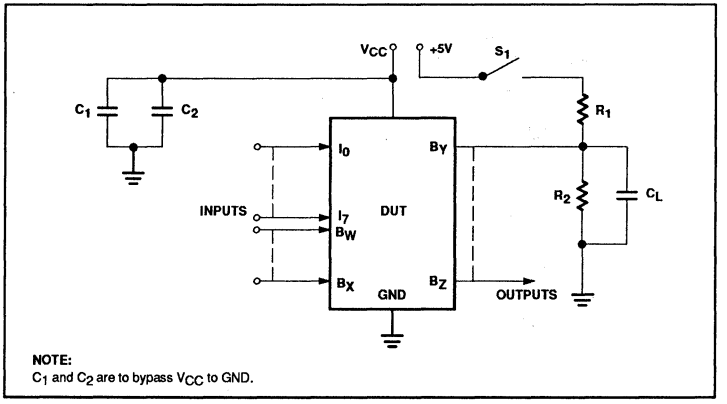
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

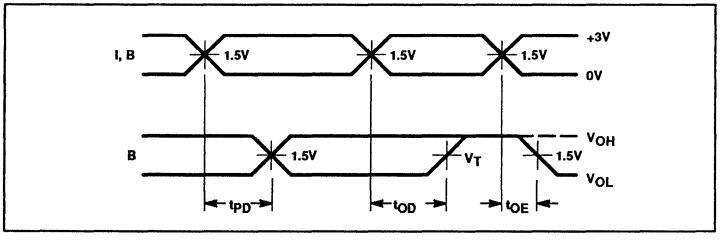


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (18 × 42 × 10)

PLS153/A

LOGIC PROGRAMMING

The PLS153/A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP, Data I/O's ABEL™ and Logical Devices, Inc. CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

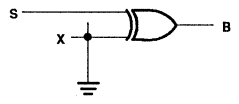
PLS153/A logic designs can also be generated using the program table entry format detailed on the following page. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

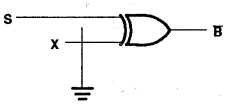
PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information

OUTPUT POLARITY – (B)

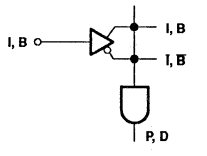


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

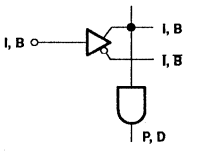


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

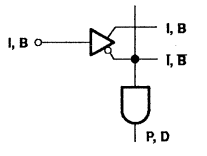
AND ARRAY – (I, B)



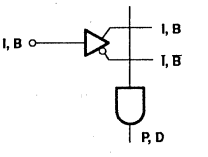
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

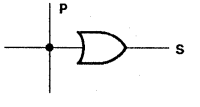


STATE	CODE
I, B̄	L

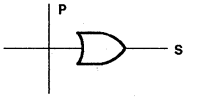


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditioanlly inhibited if both the True and Complement of an input (either I or B) are left intact.

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

CAUTION: PLS153A TEST COLUMNS

The PLS153A incorporates two columns not shown in the logic block diagram. These columns are used for in-house testing of the device in the unprogrammed state. These columns must be disabled prior to using the PLS153A in your application. If you are using a Philips Semiconductors-approved programmer, the disabling is accomplished during the device programming sequence. If these columns are not disabled, abnormal operation is possible.

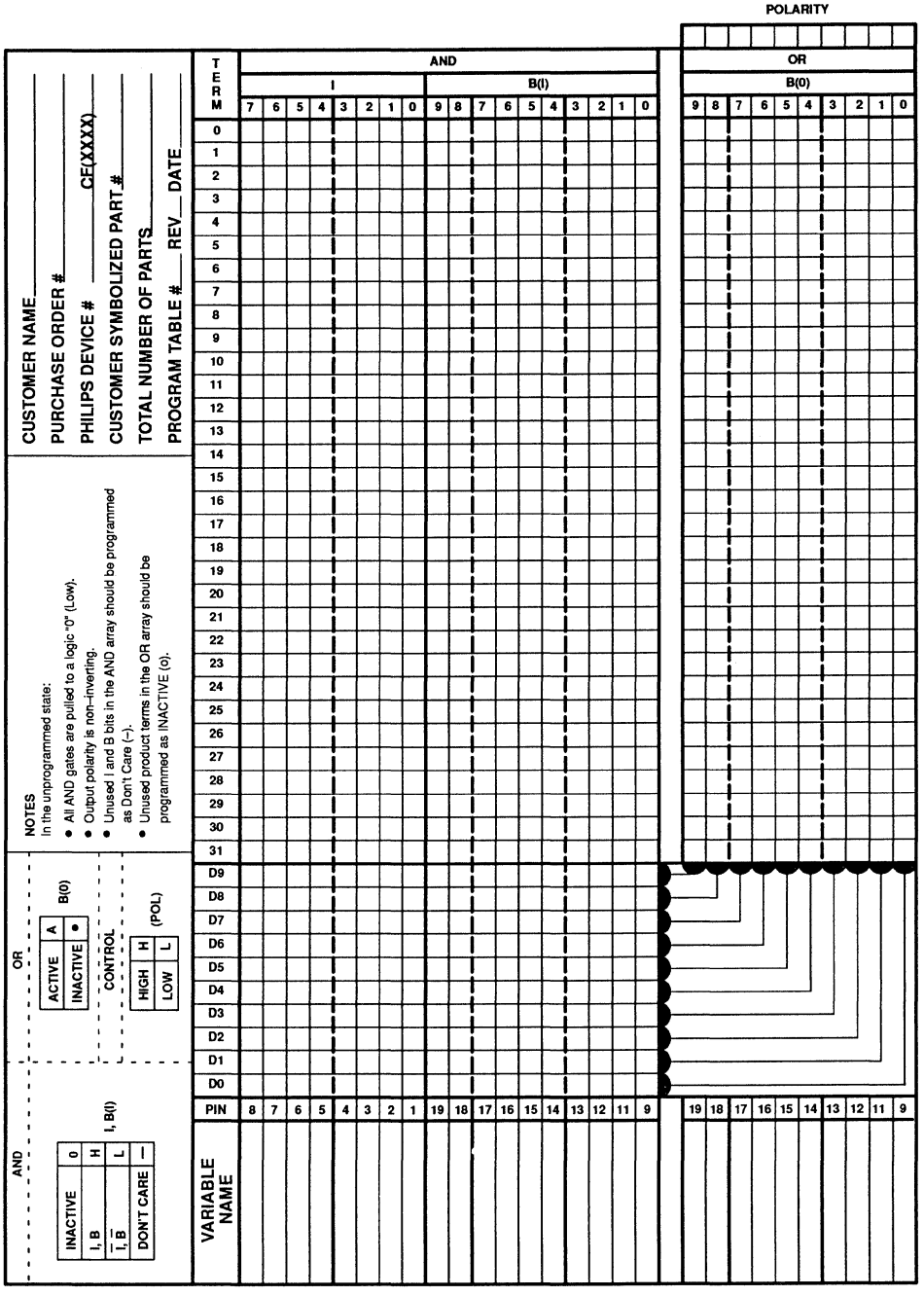
Furthermore, because of these test columns, the PLS153A cannot be programmed using the programmer algorithm for the PLS153.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

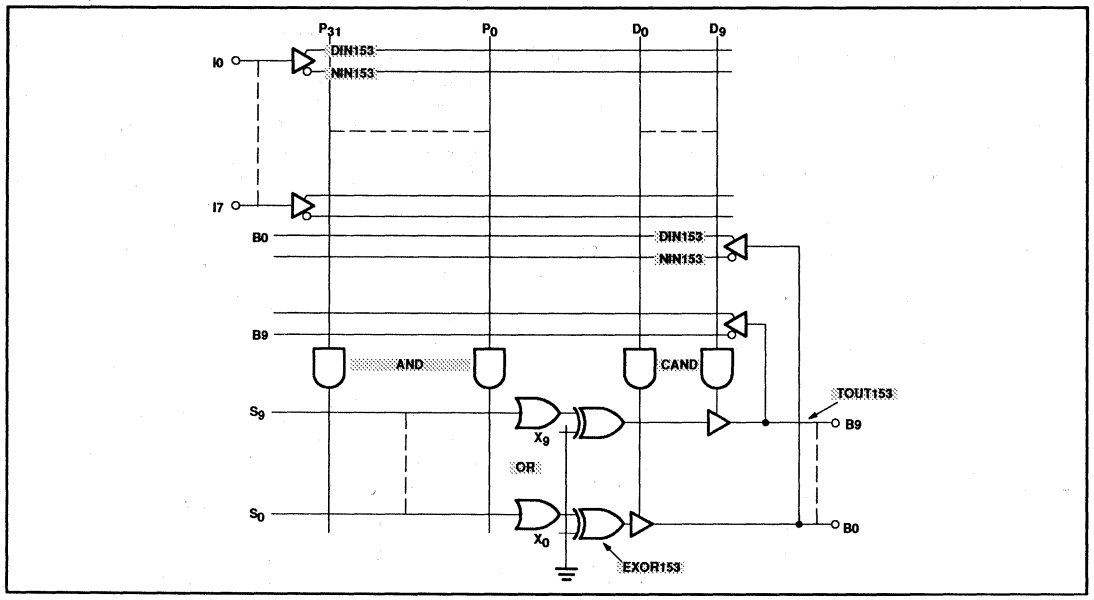
Programmable logic arrays (18 × 42 × 10)

PLS153/A

PROGRAM TABLE



SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

DESCRIPTION

The PLUS153 PLDs are high speed, combinatorial Programmable Logic Arrays. The Philips Semiconductors state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce propagation delays as short as 12ns.

The 20-pin PLUS153 devices have a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS153 devices can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either active-High or active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS153 devices are user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

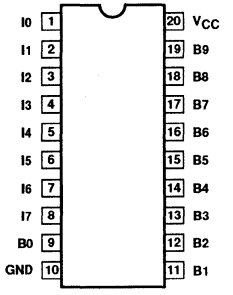
- I/O propagation delays (worst case)
  - PLUS153B – 15ns max.
  - PLUS153D – 12ns max.
- Functional superset of 16L8 and most other 20-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 8 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 750mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

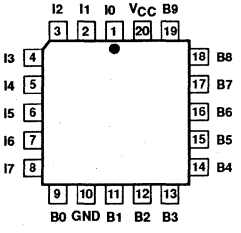
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

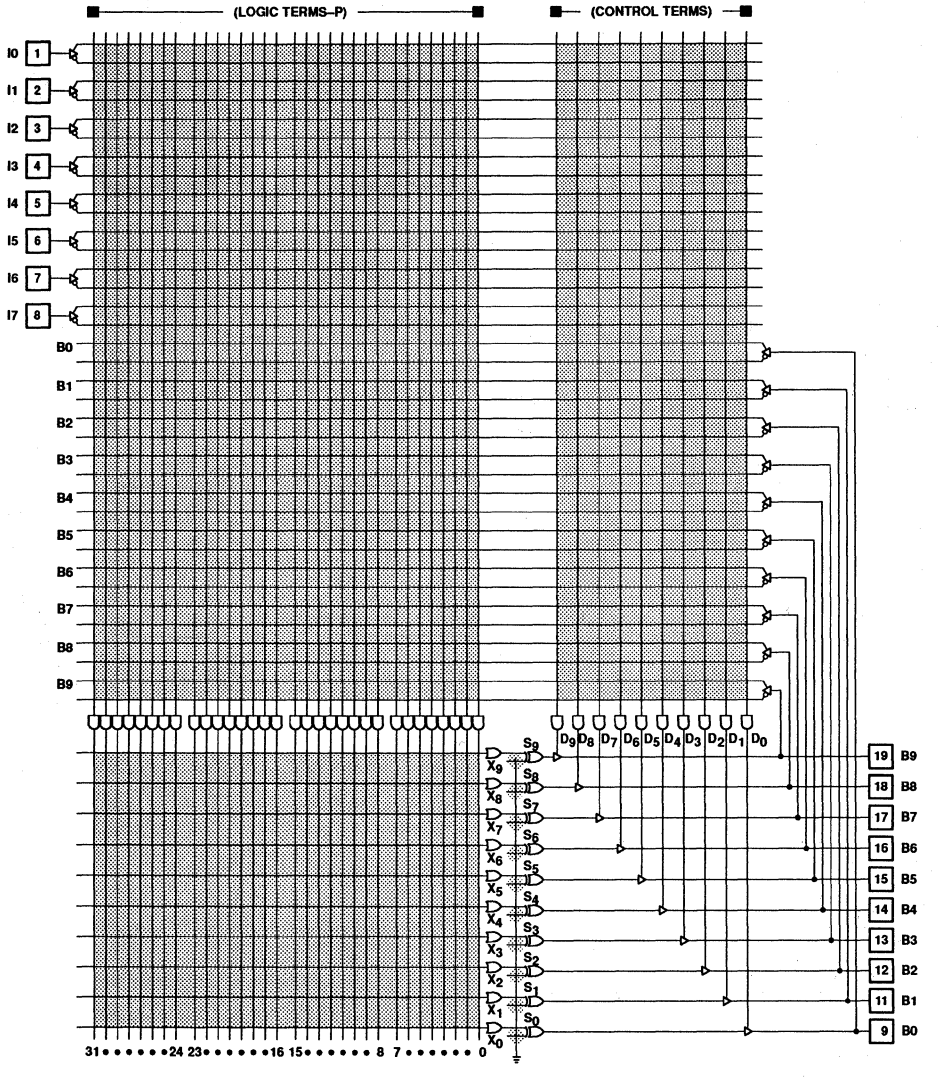
DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual-In-Line 300mil-wide	15ns	PLUS153BN	0408D
20-Pin Plastic Dual-In-Line 300mil-wide	12ns	PLUS153DN	0408D
20-Pin Plastic Leaded Chip Carrier	15ns	PLUS153BA	0400E
20-Pin Plastic Leaded Chip Carrier	12ns	PLUS153DA	0400E

®PAL is a registered trademark of Monolithic Memories, Inc., a wholly owned subsidiary of Advanced Micro Devices Corporation.

Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

LOGIC DIAGRAM

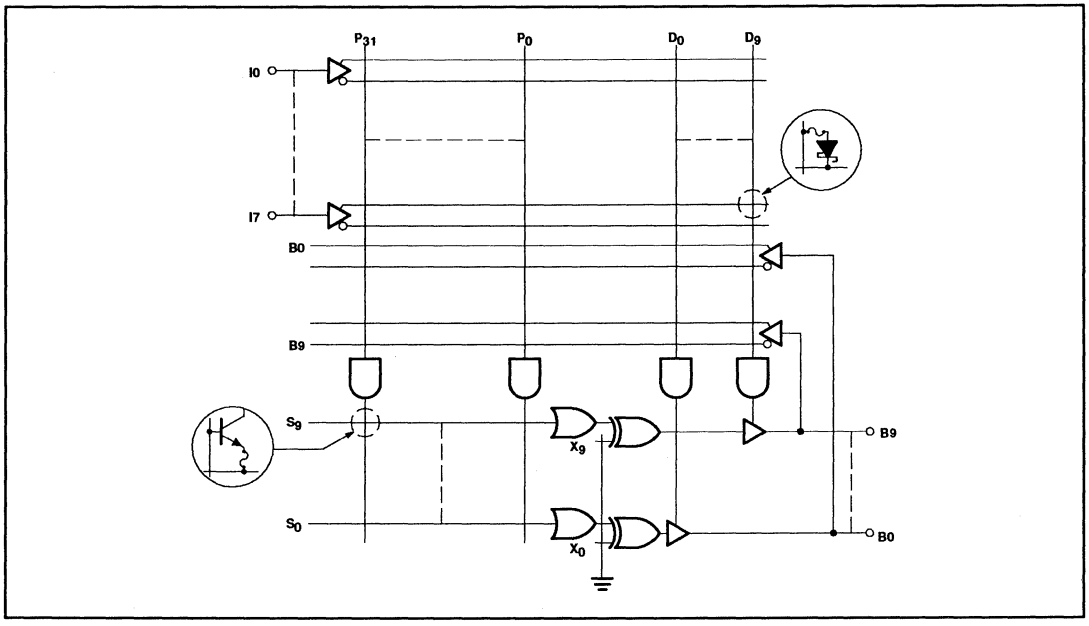


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = −2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			−100	μA
I_{IH}	High	V_{IN} = V_{CC}			40	μA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	μA
		V_{OUT} = 0.45V			−140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	200	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I2 = 0V, inputs I3 – I5 = 4.5V, inputs I7 = 4.5V and I6 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I7 = 0V.
5. Same conditions as Note 4 except I7 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I7 and B0 – B9 = 0V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only (I0 – I7).

Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

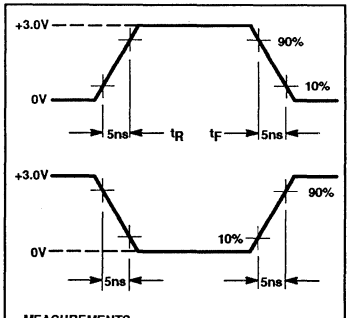
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLUS153B			PLUS153D			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		11	15		10	12	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		11	15		10	12	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		11	15		10	12	ns

NOTES:
1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

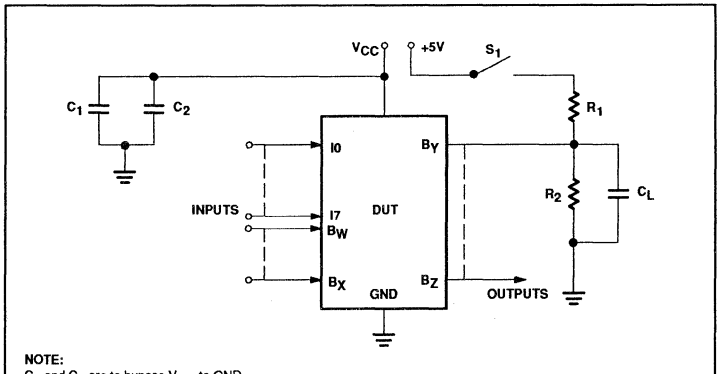
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

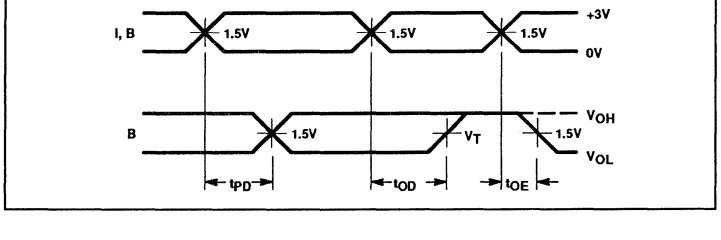


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

LOGIC PROGRAMMING

The PLUS153B/D is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLUS153B/D architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

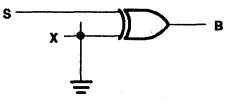
PLUS153B/D logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

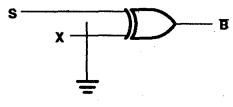
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/Software Support)* of this data handbook for additional information.

OUTPUT POLARITY – (B)

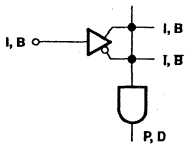


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

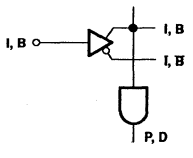


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

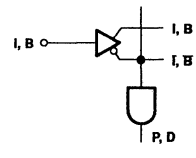
AND ARRAY – (I, B)



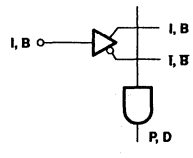
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

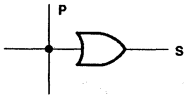


STATE	CODE
I, B̄	L

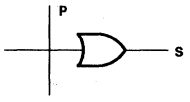


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

NOTES:

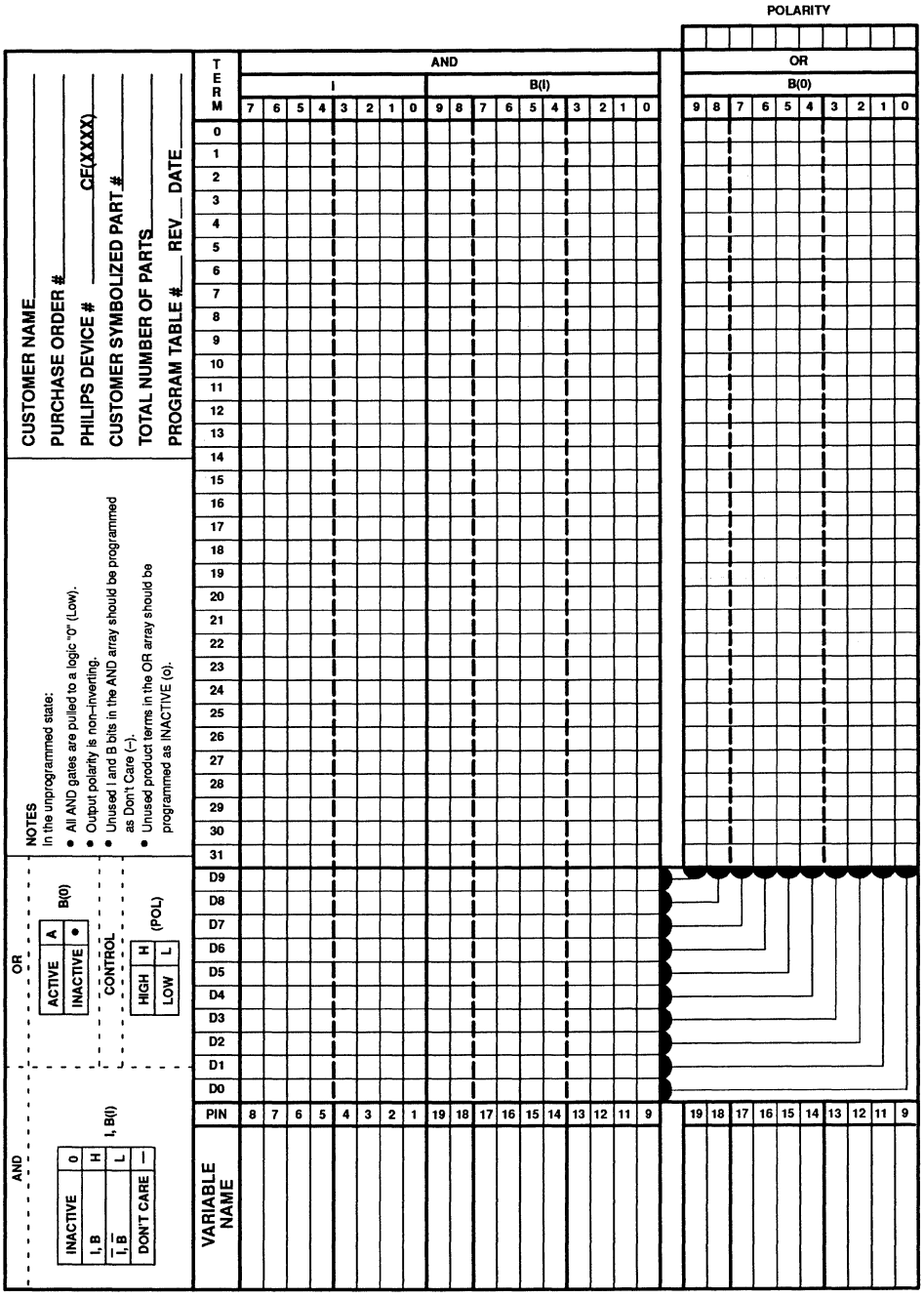
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

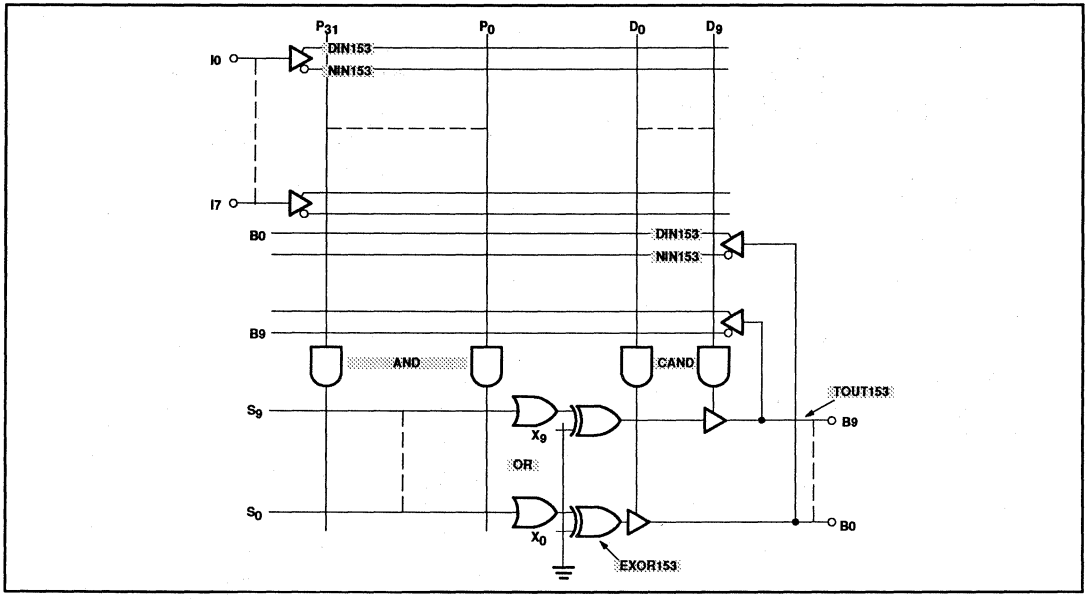
Programmable logic arrays (18 × 42 × 10)

PLUS153B/D

PROGRAM TABLE



SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic array (18 × 42 × 10)

PLUS153–10

DESCRIPTION

The PLUS153–10 PLD is a high speed, combinatorial Programmable Logic Array. The Philips Semiconductors state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce maximum propagation delays of 10ns or less.

The 20-pin PLUS153 device has a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS153–10 can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS153–10 device is user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

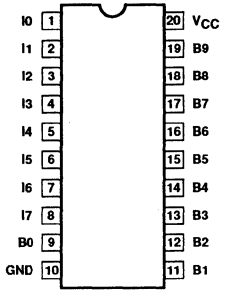
- I/O propagation delays (worst case)
  - PLUS153–10 – 10ns max.
- Functional superset of 16L8 and most other 20-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 8 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 825mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

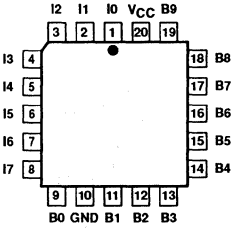
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

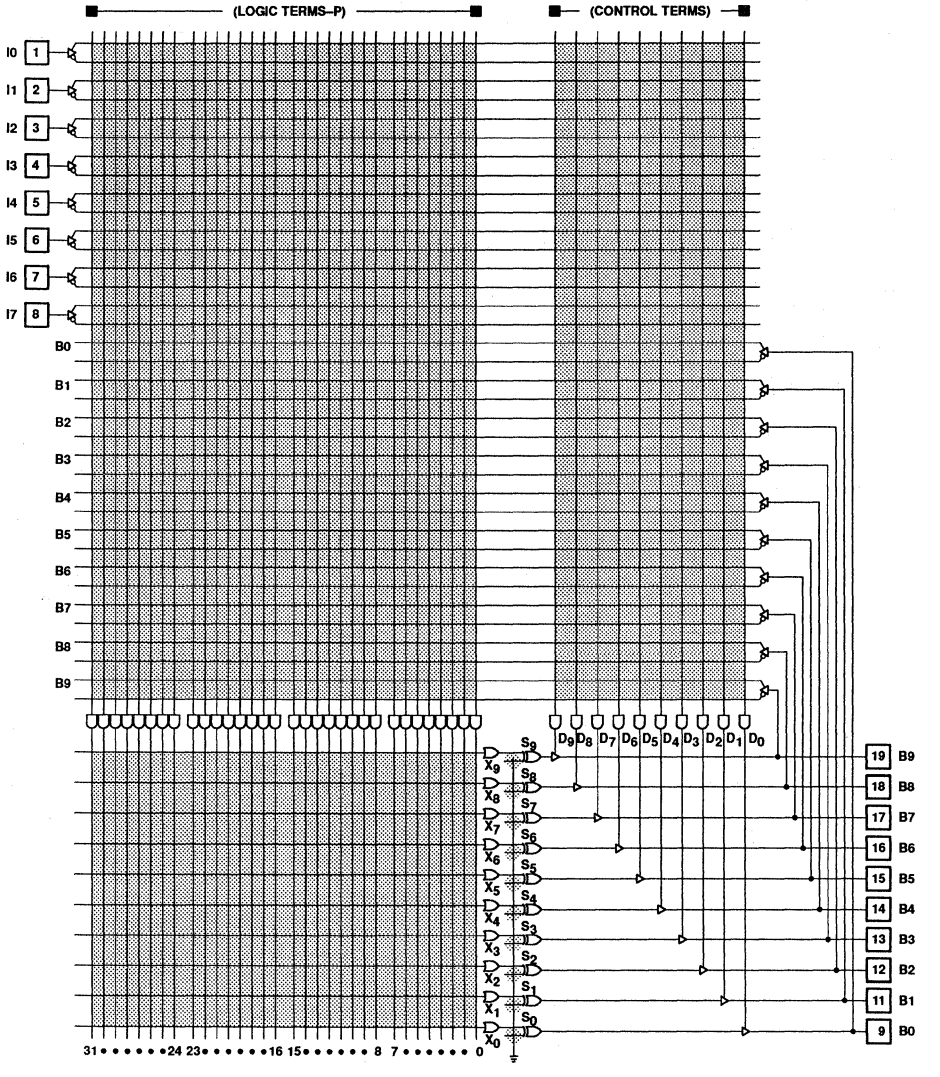
DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual-In-Line 300mil-wide	10ns	PLUS153–10N	0408D
20-Pin Plastic Leaded Chip Carrier	10ns	PLUS153–10A	0400E

®PAL is a registered trademark of Advanced Micro Devices Corporation.

Programmable logic array (18 × 42 × 10)

PLUS153–10

LOGIC DIAGRAM

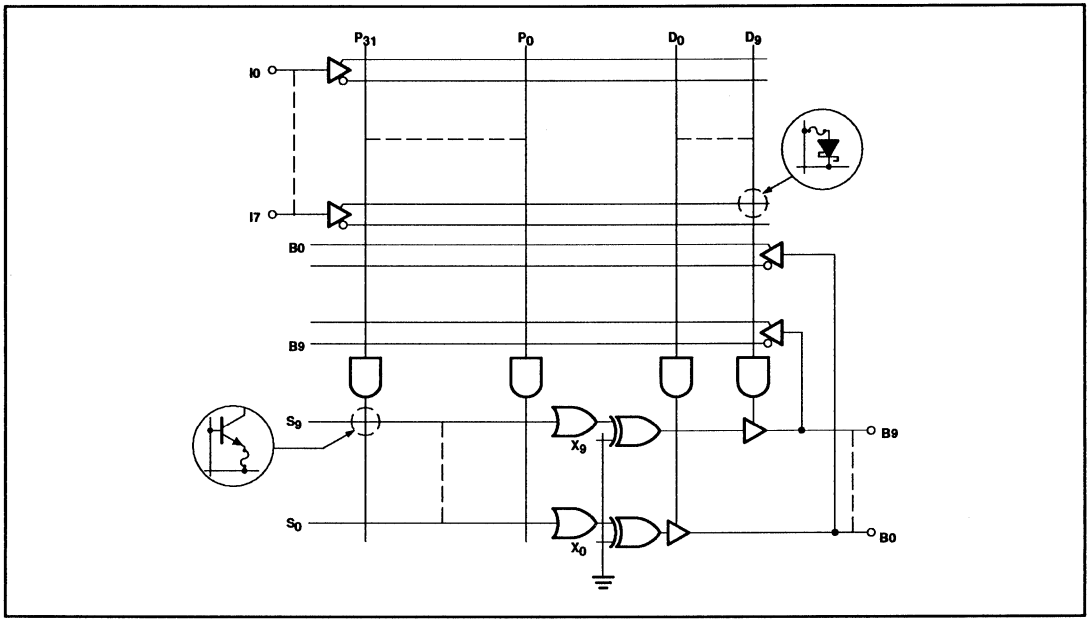


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic array (18 × 42 × 10)

PLUS153–10

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{In}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA		0.4	0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4	2.9		V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V		–20	–100	µA
I_{IH}	High	V_{IN} = V_{CC}		1	40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V		0	80	µA
		V_{OUT} = 0.45V		–15	–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15	–30	–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		165	200	mA
Capacitance						
		V_{CC} = 5V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I2 = 0V, inputs I3 – I5 = 4.5V, inputs I7 = 4.5V and I6 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I7 = 0V.
5. Same conditions as Note 4 except I7 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I7 and B0 – B9 = 0V.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only (I0 – I7).

Programmable logic array (18 × 42 × 10)

PLUS153–10

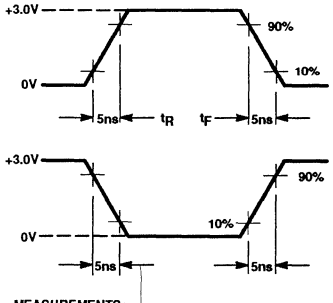
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		8	10	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		8	10	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		8	10	ns

NOTES:
1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

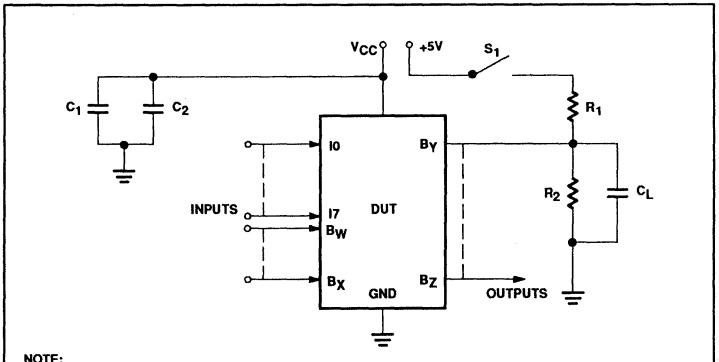
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

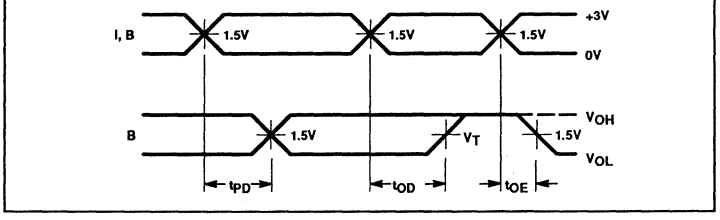


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (18 × 42 × 10)

PLUS153–10

LOGIC PROGRAMMING

The PLUS153–10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLUS153–10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

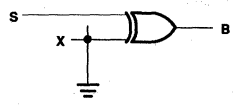
PLUS153–10 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

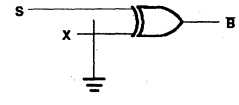
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/Software Support)* of this data handbook for additional information.

OUTPUT POLARITY – (B)

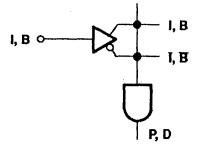


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

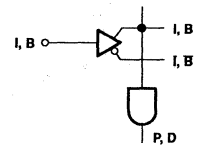


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

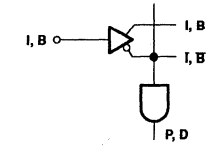
AND ARRAY – (I, B)



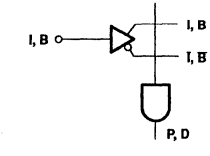
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

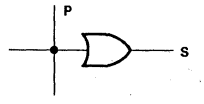


STATE	CODE
I, B̄	L

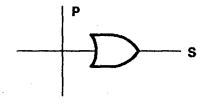


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:
1. This is the initial unprogrammed state of all links.
2. Any gate P_n will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

VIRGIN STATE

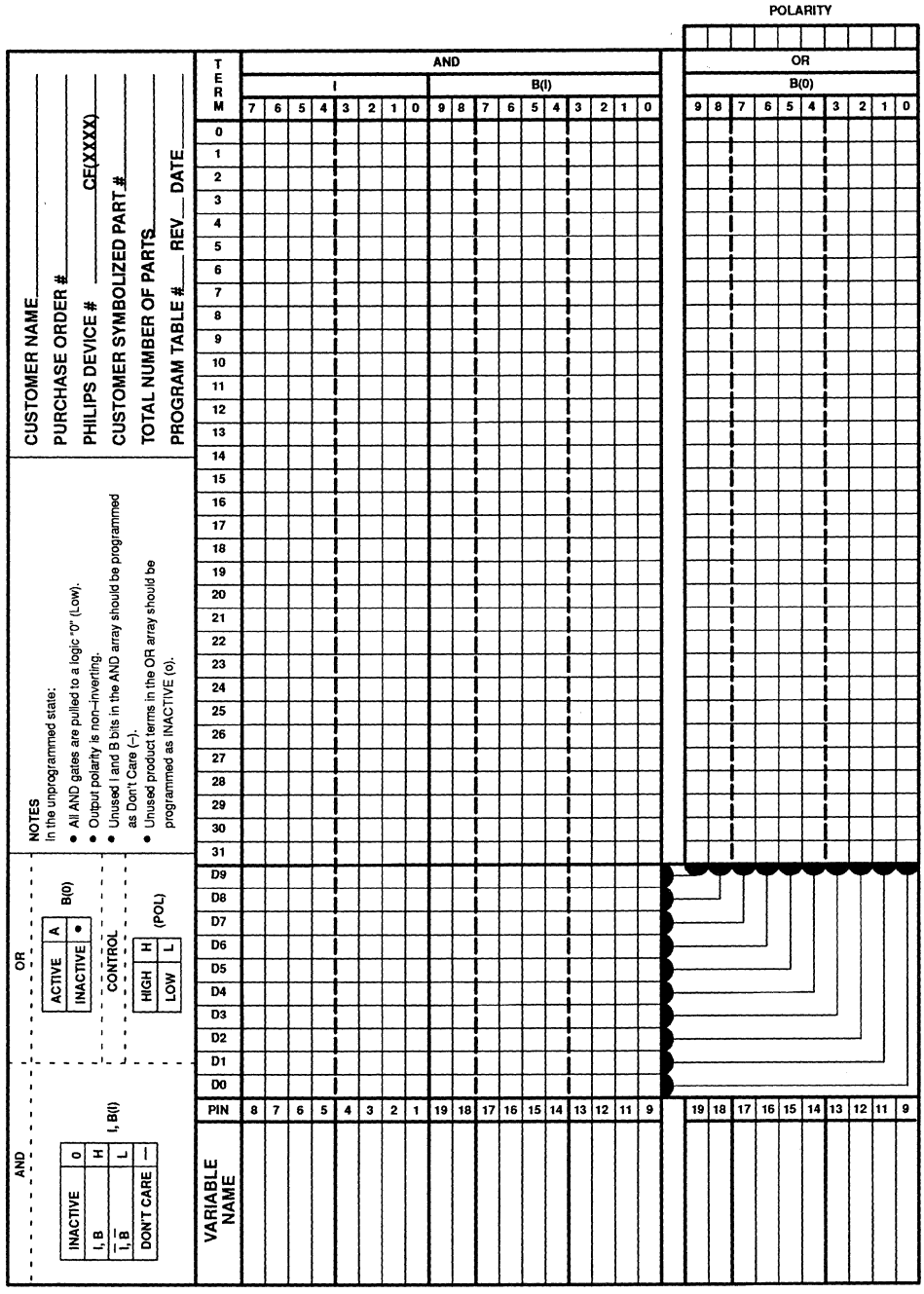
A factory shipped virgin device contains all fusible links intact, such that:
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic array (18 × 42 × 10)

PLUS153–10

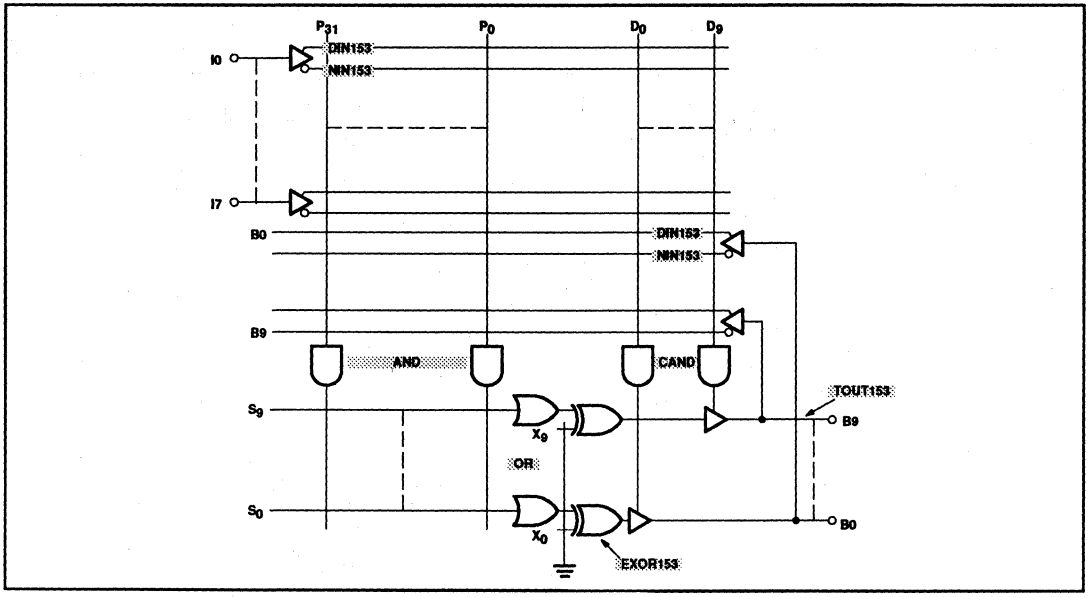
PROGRAM TABLE



Programmable logic array (18 × 42 × 10)

PLUS153–10

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic array (22 × 42 × 10)

PLS173

DESCRIPTION

The PLS173 is a two-level logic element consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 12 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 22 inputs to 10 outputs.

On-chip T/C buffers couple either True (I, B) or Complement ($\overline{I}$, $\overline{B}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS173 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes for this device are listed below.

FEATURES

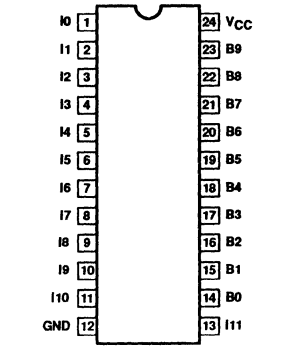
- I/O propagation delay: 30ns (max.)
- 12 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- Ni-Cr programmable links
- Input loading: –100μA (max.)
- Power dissipation: 750mW (typ.)
- 3-State outputs
- TTL compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

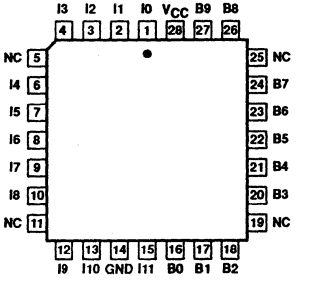
PIN CONFIGURATIONS

N Package



N = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

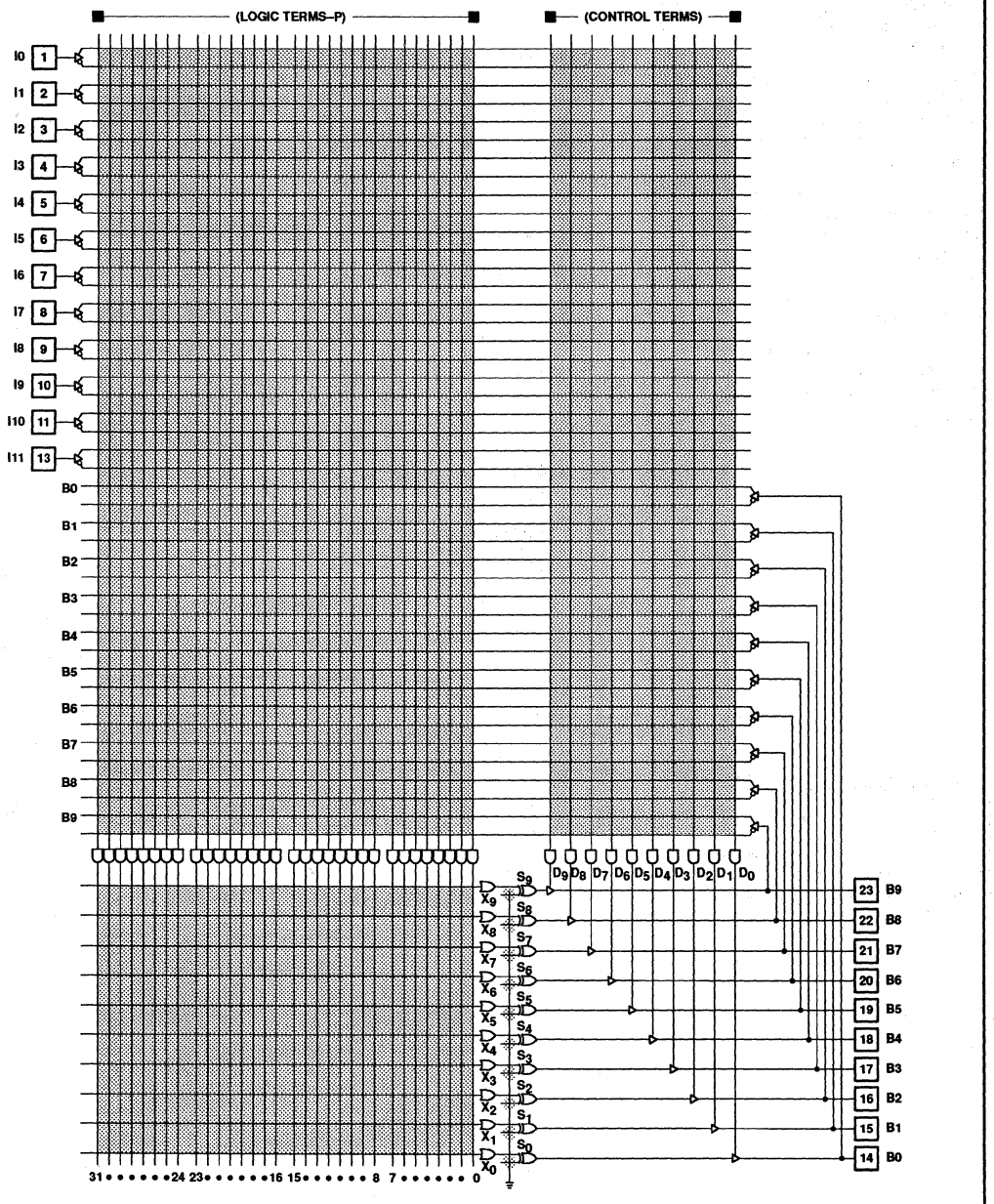
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING N'IMBER
24-Pin Plastic Dual-In-Line 300mil-wide	PLS173N	0410D
28-Pin Plastic Leaded Chip Carrier	PLS173A	0401F

Programmable logic array (22 × 42 × 10)

PLS173

LOGIC DIAGRAM

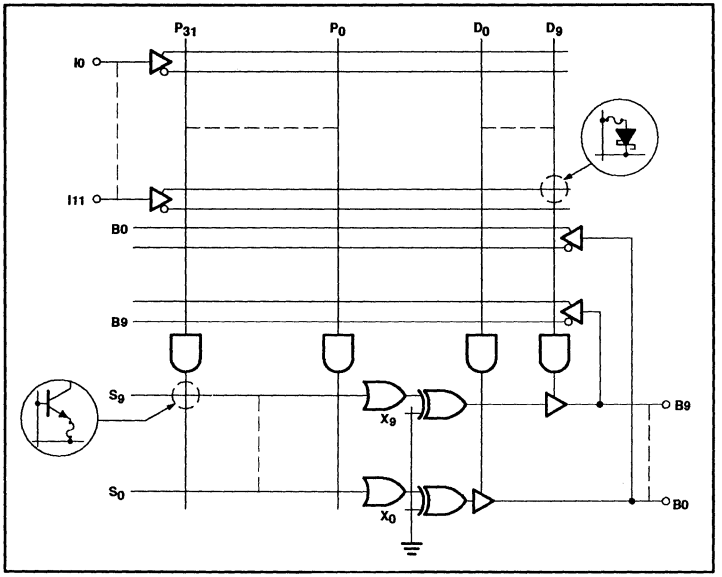


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic array (22 × 42 × 10)

PLS173

FUNCTIONAL DIAGRAM



LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$Pn = A \cdot \overline{B} \cdot C \cdot D \cdot \ldots$

TYPICAL LOGIC FUNCTION:

AT OUTPUT POLARITY = H

$Z = P0 + P1 + P2 \ldots$

AT OUTPUT POLARITY + L

$Z = \overline{P0 + P1 + P2 + \ldots}$

$Z = \overline{P0} \cdot \overline{P1} \cdot \overline{P2} \cdot \ldots$

NOTES:

1. For each of the 10 outputs, either function Z (Active-High) or $\overline{Z}$ (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. ZX, A, B, C, etc. are user defined connections to fixed inputs (I), and bidirectional pins (B).

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING Min	RATING Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS173 is also processed to military requirements for operation over the military temperature range. For specifications and ordering information, consult the Philips Semiconductors Military Data Handbook.

Programmable logic array (22 × 42 × 10)

PLS173

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 5.5V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	170	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs V_{IL} applied to I_{11}. Pins 1–5 = 0V, Pins 6–10 = 4.5V, Pin 11 = 0V and Pin 13 = 10V.
5. Same conditions as Note 4 except Pin 11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with I_0 and I_1 = 0V, and $I_2 - I_{11}$ and $B_0 - B_9$ = 4.5V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only ($I_0 - I_{11}$).

Programmable logic array (22 × 42 × 10)

PLS173

AC ELECTRICAL CHARACTERISTICS

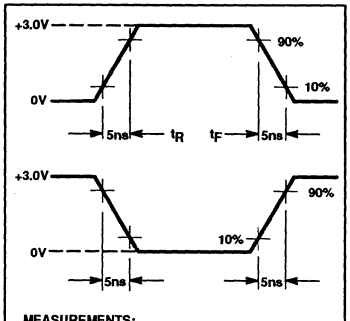
$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation delay[2]	Input ±	Output ±	C_L = 30pF		20	30	ns
t_{OE}	Output enable[1]	Input ±	Output –	C_L = 30pF		20	30	ns
t_{OD}	Output disable[1]	Input ±	Output +	C_L = 5pF		20	30	ns

NOTES:

1. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

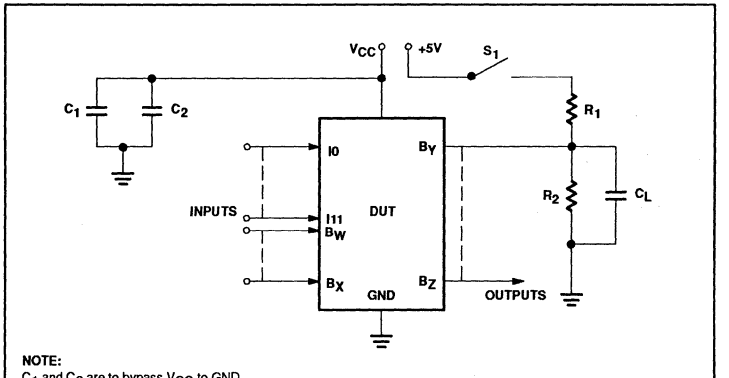
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

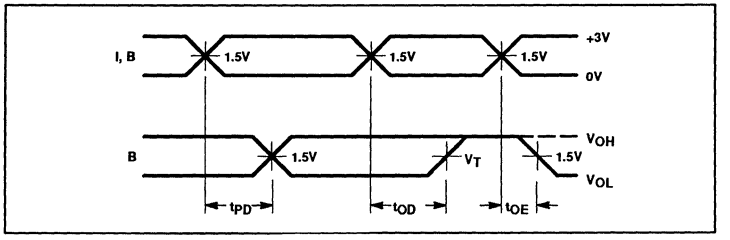


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (22 × 42 × 10)

PLS173

LOGIC PROGRAMMING

The PLS173 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP, Data I/O Corporation's ABEL™, and Logical Devices Incorporated's CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

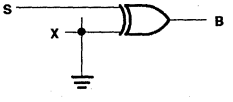
PLS173 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

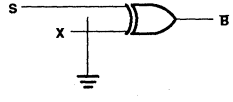
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/Software Support)* of this data handbook for addtional information.

OUTPUT POLARITY – (B)

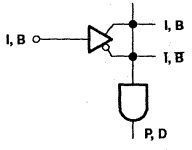


ACTIVE LEVEL	CODE
HIGH[1] (NON-NVERTING)	H

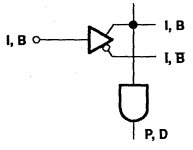


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

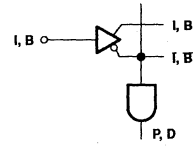
AND ARRAY – (I, B)



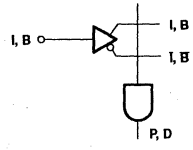
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

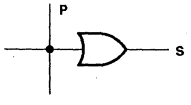


STATE	CODE
I, B	L

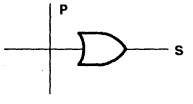


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the True and Complement of any input (I, B) are left intact.

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

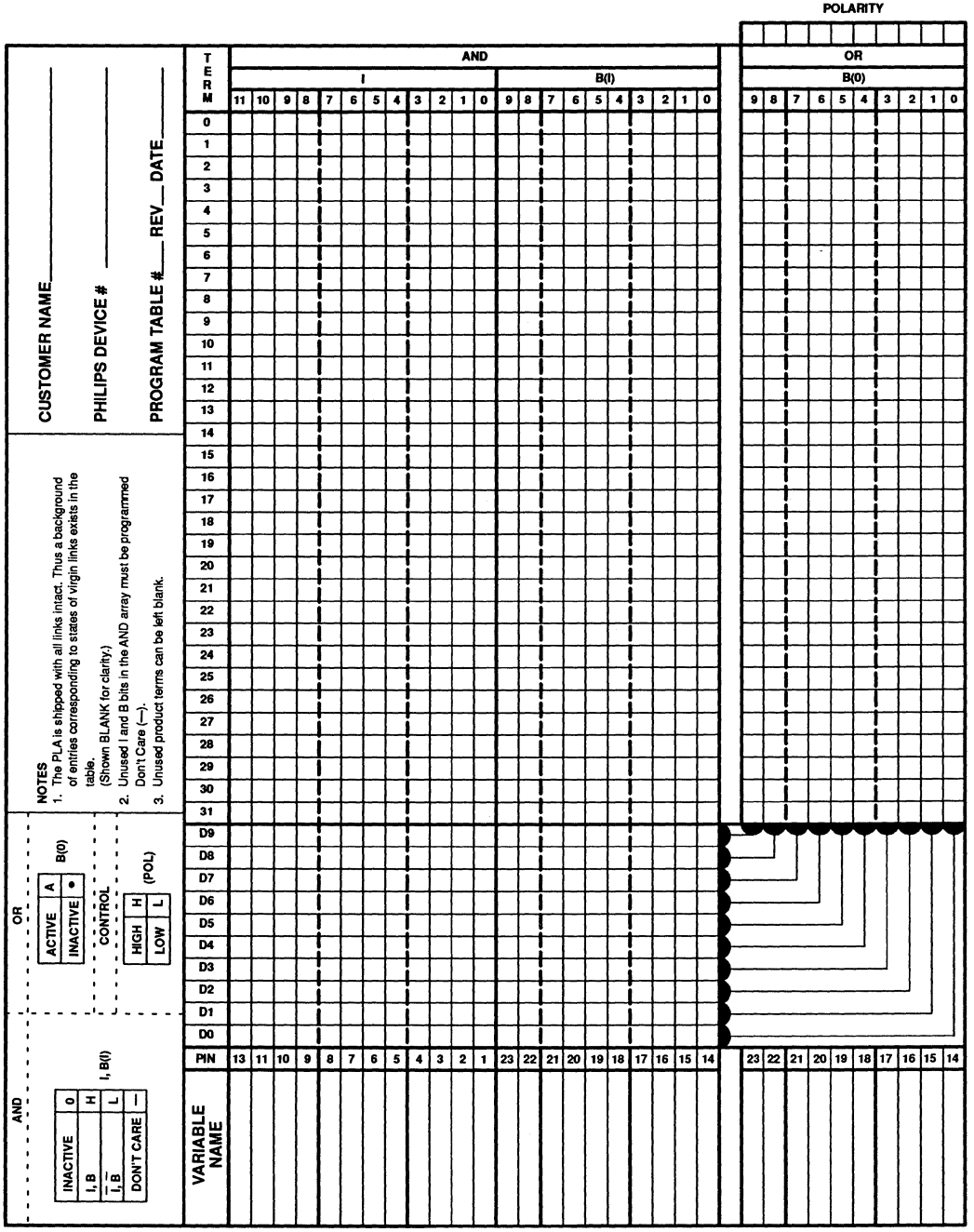
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic array (22 × 42 × 10)

PLS173

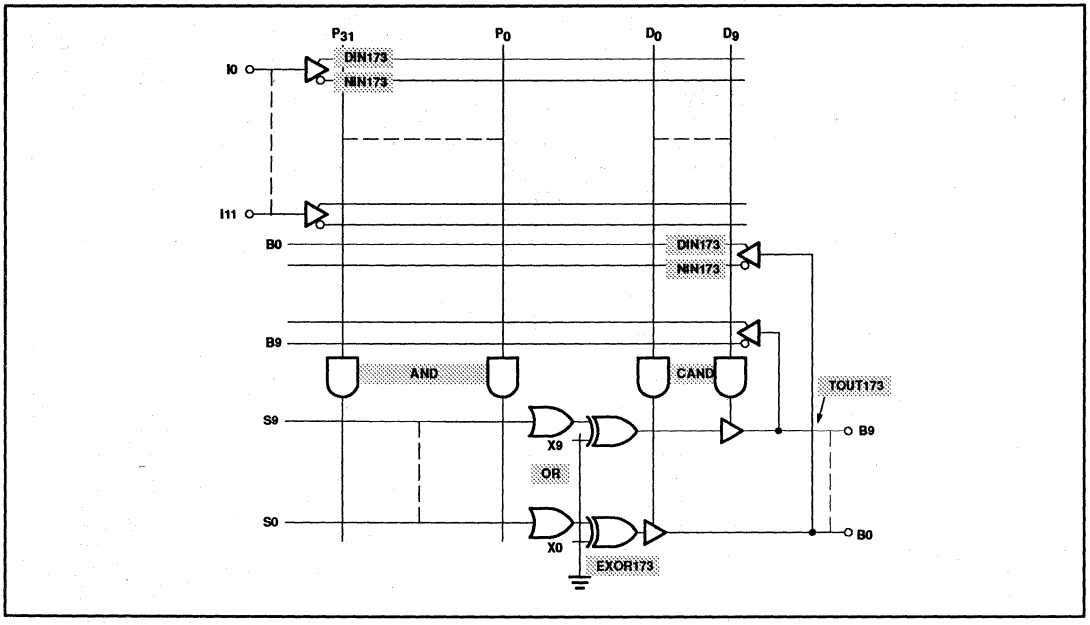
PROGRAM TABLE



Programmable logic array (22 × 42 × 10)

PLS173

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

DESCRIPTION

The PLUS173 PLDs are high speed, combinatorial Programmable Logic Arrays. The Philips Semiconductors state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce propagation delays as short as 12ns.

The 24-pin PLUS173 devices have a programmable AND array and a programmable OR array. Unlike PAL ® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS173 devices can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS173 devices are user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

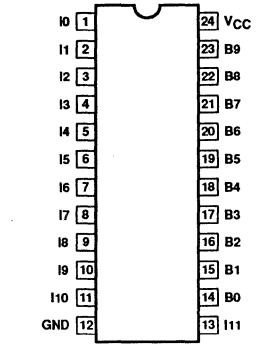
- I/O propagation delays (worst case)
  - PLUS173B – 15ns max.
  - PLUS173D – 12ns max.
- Functional superset of 20L10 and most other 24-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 12 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 750mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

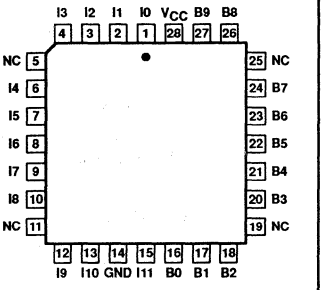
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

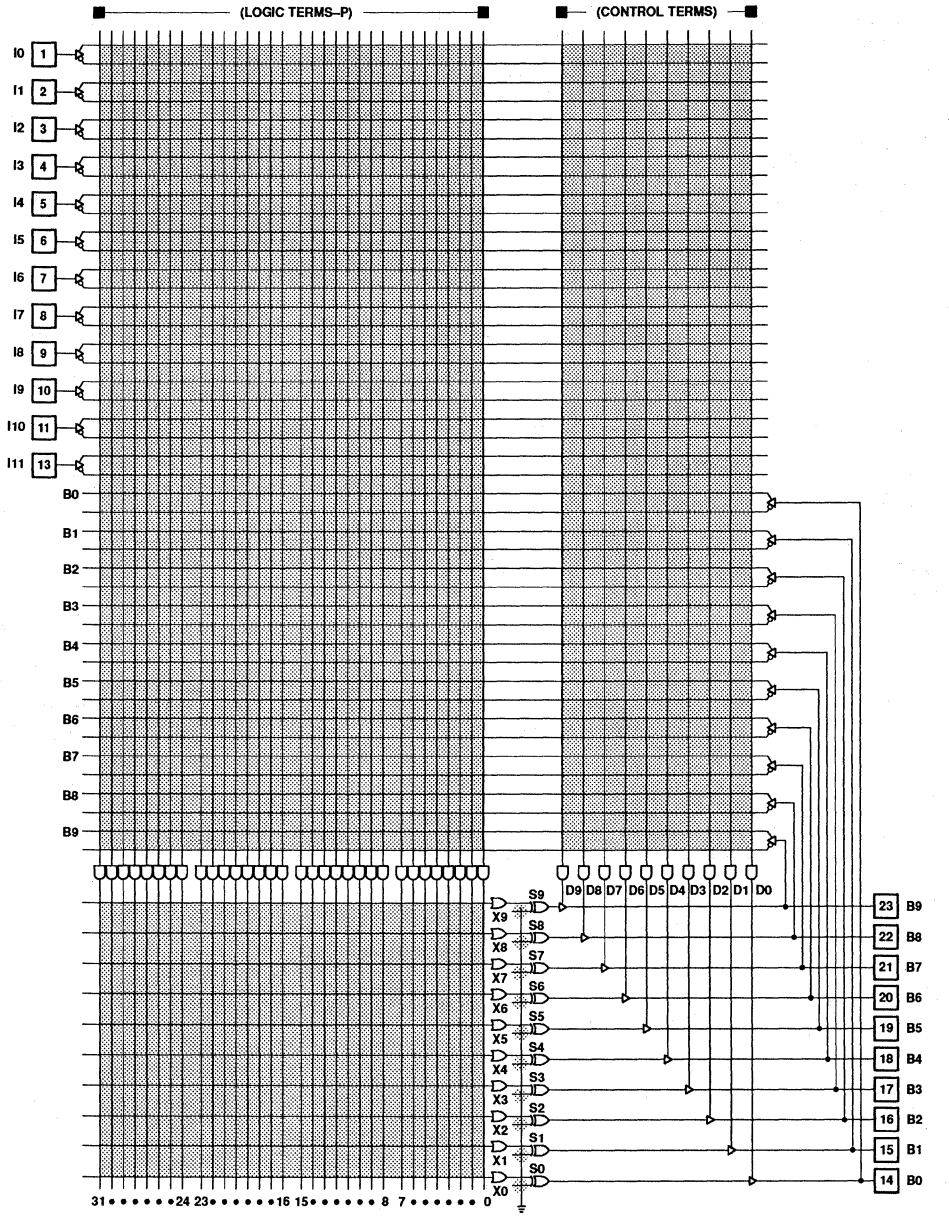
DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
24-Pin Plastic Dual In-Line 300mil-wide	15ns	PLUS173BN	0410D
24-Pin Plastic Dual In-Line 300mil-wide	12ns	PLUS173DN	0410D
28-Pin Plastic Leaded Chip Carrier	15ns	PLUS173BA	0401F
28-Pin Plastic Leaded Chip Carrier	12ns	PLUS173DA	0401F

®PAL is a registered trademark of Advanced Micro Devices Corporation.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

LOGIC DIAGRAM

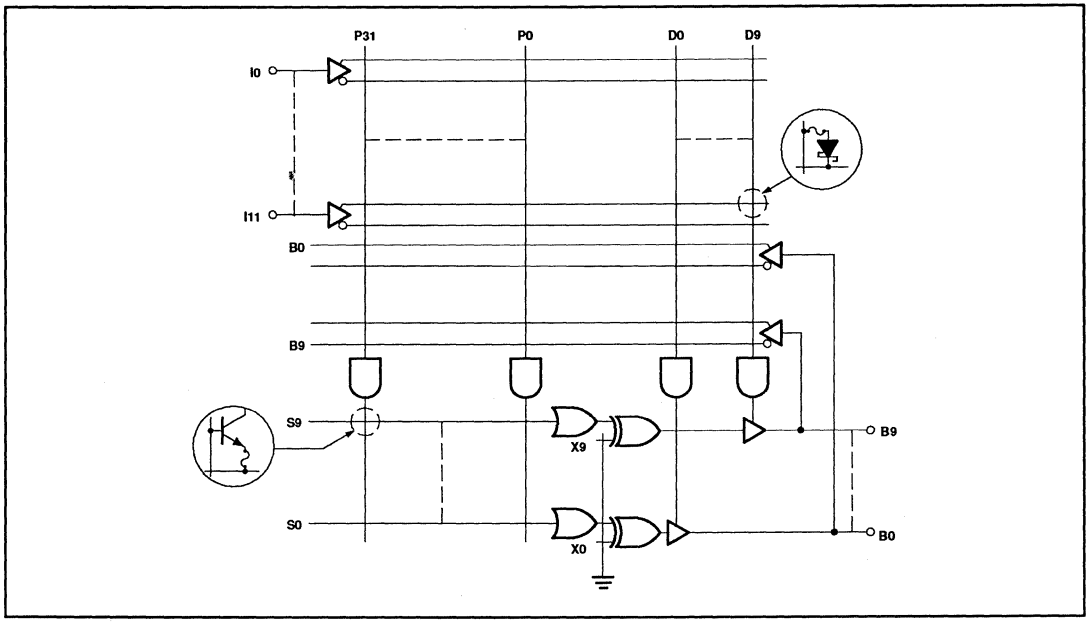


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ⁘ Programmable connection.

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING MIN	RATING MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA			0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4			V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V			–100	µA
I_{IH}	High	V_{IN} = V_{CC}			40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V			80	µA
		V_{OUT} = 0.45V			–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		150	200	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I4 = 0V, inputs I5 – I9 = 4.5V, I11 = 4.5V and I19 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I11 = 0V.
5. Same conditions as Note 4 except input I11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I11 and B0 – B9 = 0V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only (I0 – I11).

Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

AC ELECTRICAL CHARACTERISTICS

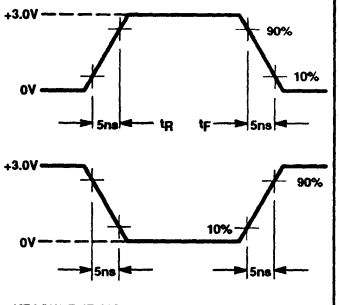
$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS						UNIT
					PLUS173B			PLUS173D			
					MIN	TYP	MAX	MIN	TYP	MAX	
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		11	15		10	12	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		11	15		10	12	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		11	15		10	12	ns

NOTES:

1. For 3-State outputs; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

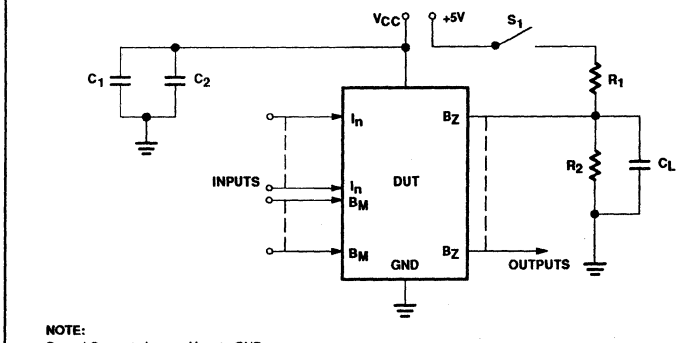
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT



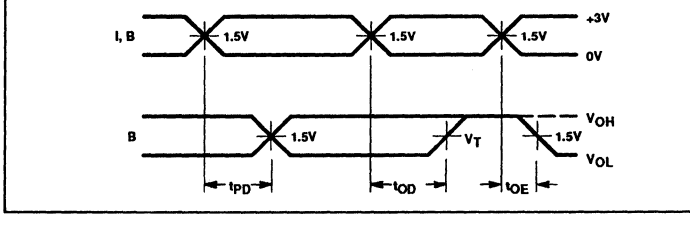
NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

LOGIC PROGRAMMING

The PLUS173 series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLUS173 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

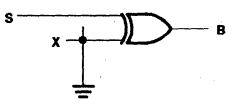
PLUS173 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

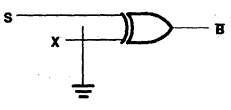
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-Party Programmer/ Software Support)* of this data handbook for additional information.

OUTPUT POLARITY – (B)

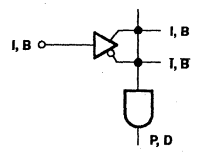


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

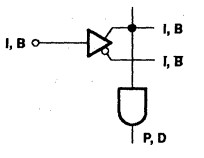


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

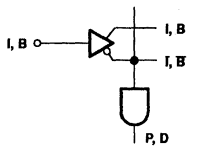
AND ARRAY – (I, B)



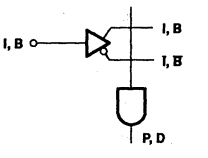
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

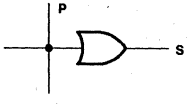


STATE	CODE
I, B̄	L

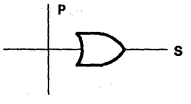


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the true and complement of any input (I, B) are left intact.

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

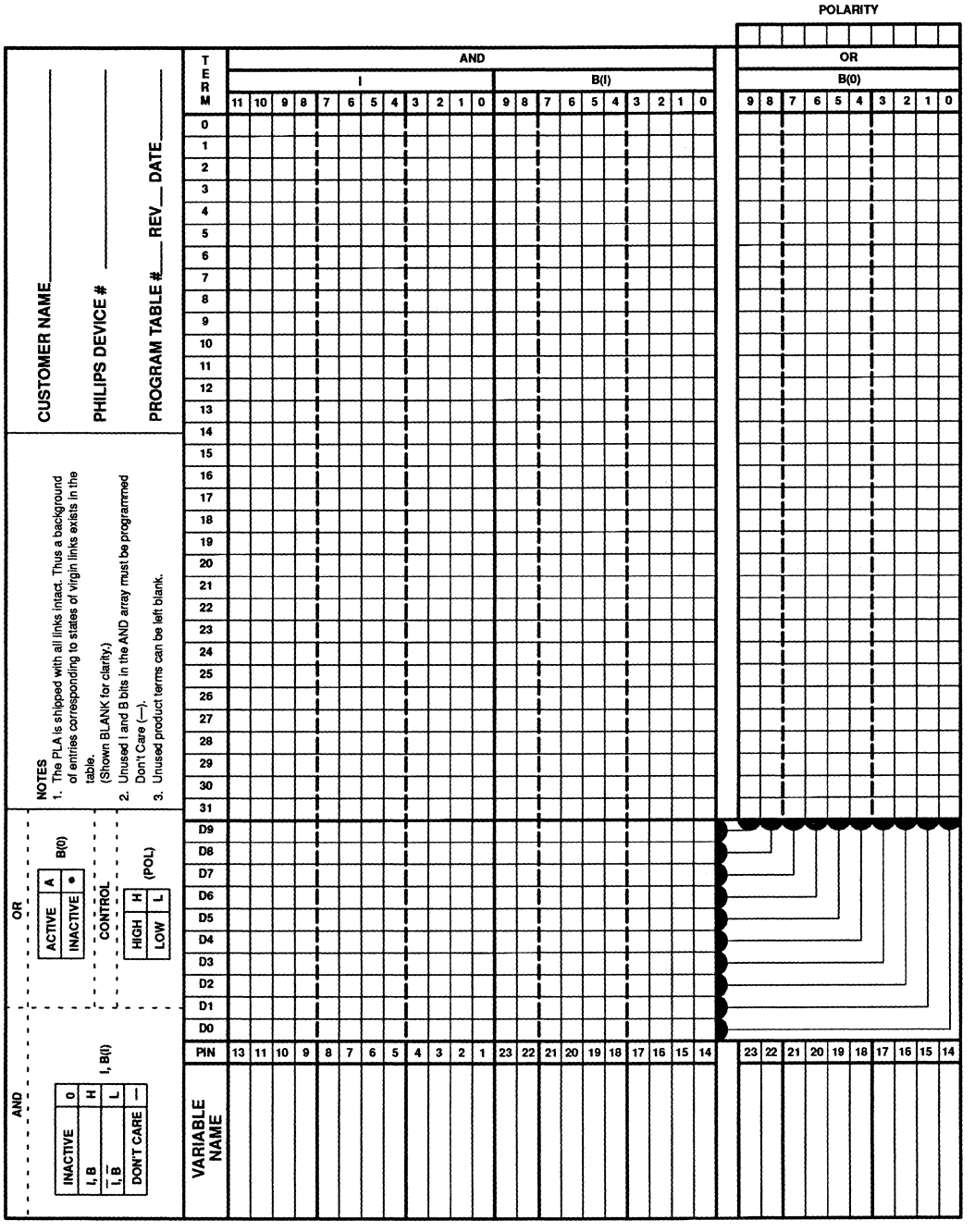
1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic arrays ($22 \times 42 \times 10$)

PLUS173B/D

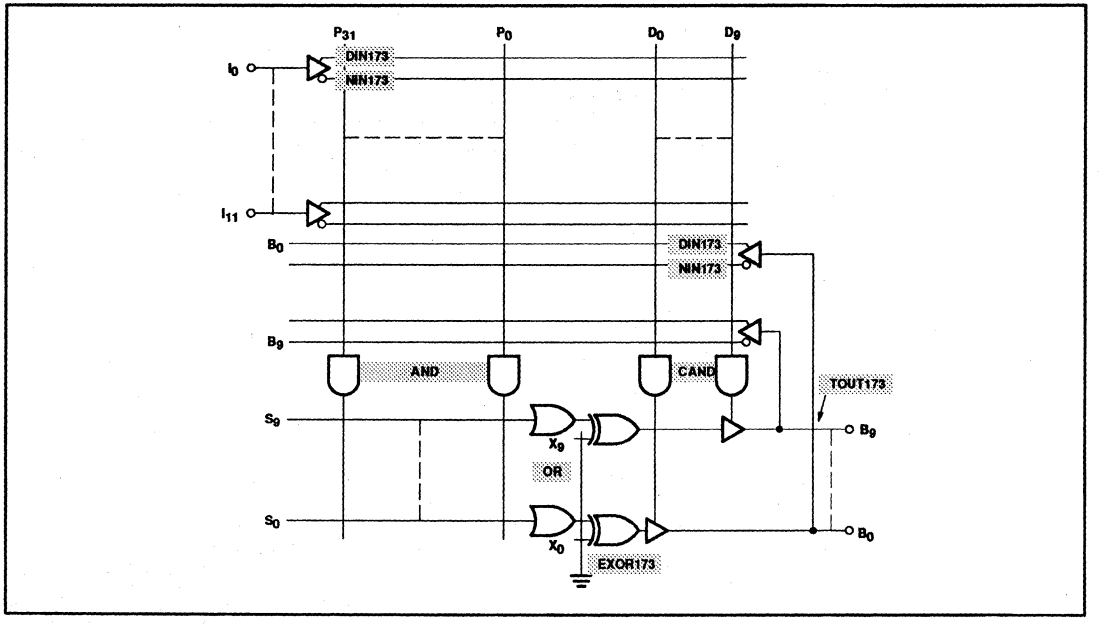
PROGRAM TABLE



Programmable logic arrays (22 × 42 × 10)

PLUS173B/D

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic array (22 × 42 × 10)

PLUS173–10

DESCRIPTION

The PLUS173–10 PLD is a high speed, combinatorial Programmable Logic Array. The Philips Semiconductors state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce maximum propagation delays of 10ns or less.

The 24-pin PLUS173–10 device has a programmable AND array and a programmable OR array. Unlike PAL® devices, 100% product term sharing is supported. Any of the 32 logic product terms can be connected to any or all of the 10 output OR gates. Most PAL ICs are limited to 7 AND terms per OR function; the PLUS173–10 device can support up to 32 input wide OR functions.

The polarity of each output is user-programmable as either Active-High or Active-Low, thus allowing AND-OR or AND-NOR logic implementation. This feature adds an element of design flexibility, particularly when implementing complex decoding functions.

The PLUS173–10 device is user-programmable using one of several commercially available, industry standard PLD programmers.

FEATURES

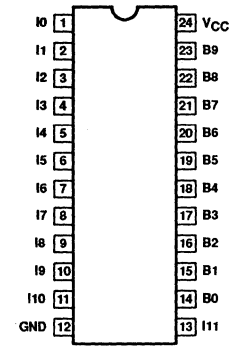
- I/O propagation delays
  - 10ns (worst case)
- Functional superset of 20L10 and most other 24-pin combinatorial PAL devices
- Two programmable arrays
  - Supports 32 input wide OR functions
- 12 inputs
- 10 bi-directional I/O
- 42 AND gates
  - 32 logic product terms
  - 10 direction control terms
- Programmable output polarity
  - Active-High or Active-Low
- Security fuse
- 3-State outputs
- Power dissipation: 850mW (typ.)
- TTL Compatible

APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

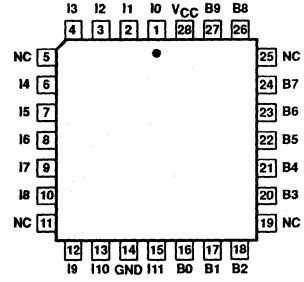
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

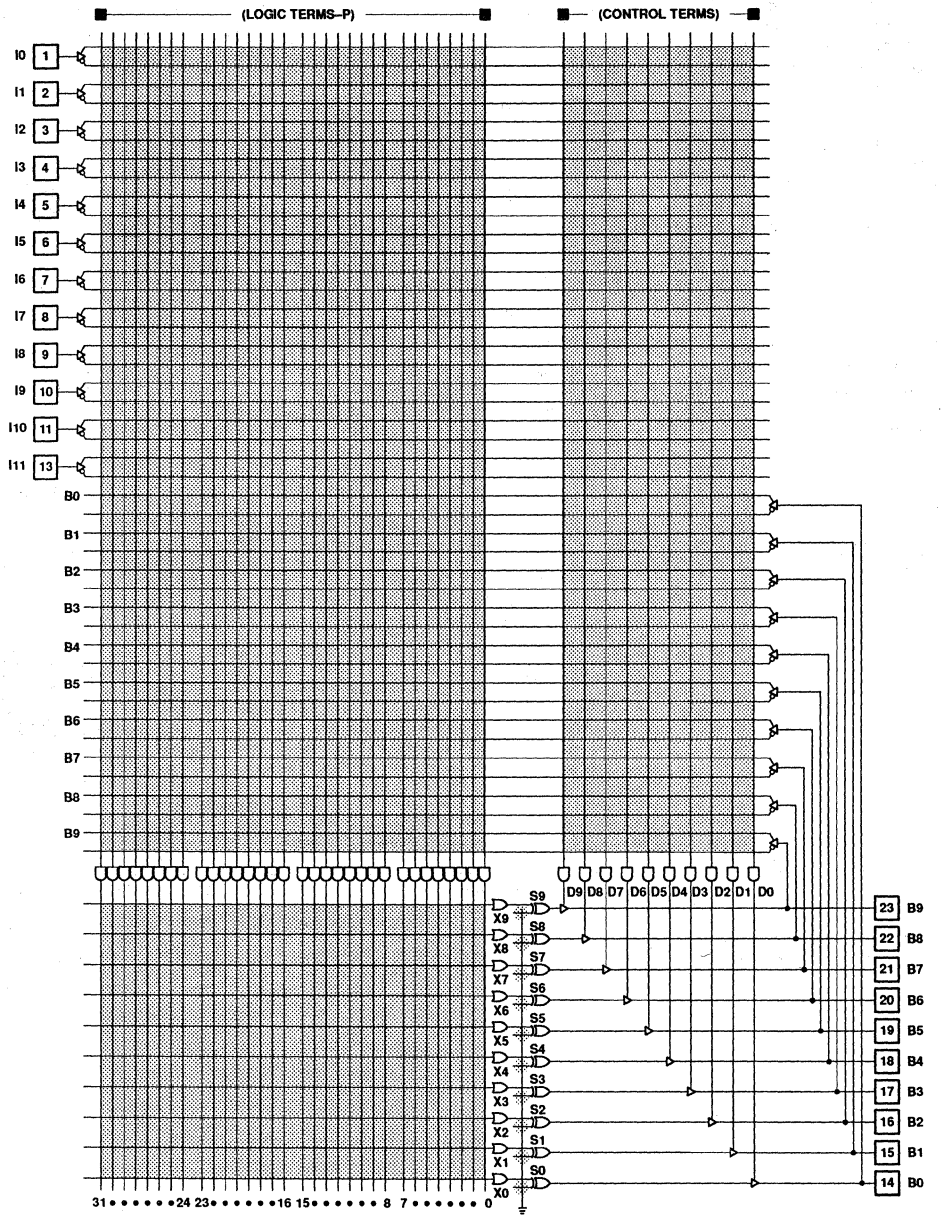
DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
24-Pin Plastic Dual In-Line 300mil-wide	10ns	PLUS173–10N	0410D
28-Pin Plastic Leaded Chip Carrier	10ns	PLUS173–10A	0401F

®PAL is a registered trademark of Advanced Micro Devices Corporation.

Programmable logic array (22 × 42 × 10)

PLUS173–10

LOGIC DIAGRAM

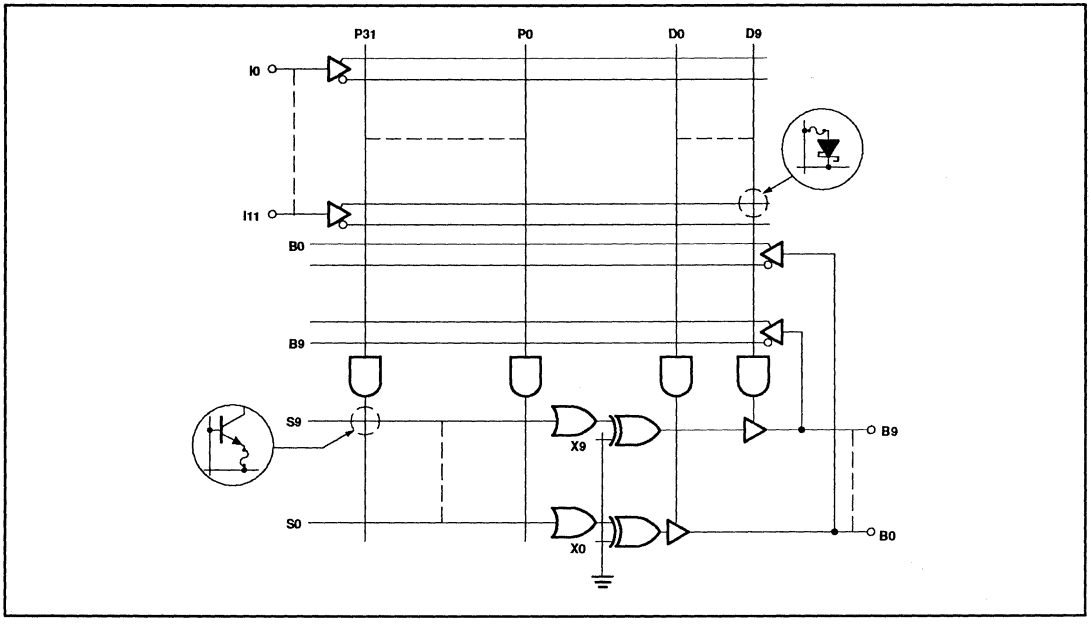


NOTES:
1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic array (22 × 42 × 10)

PLUS173–10

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATING Min	RATING Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100.0	mA
T_{amb}	Operating free-air temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

Programmable logic array (22 × 42 × 10)

PLUS173–10

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OL}	Low[4]	I_{OL} = 15mA		0.4	0.5	V
V_{OH}	High[5]	I_{OH} = –2mA	2.4	2.9		V
Input current[9]						
		V_{CC} = MAX				
I_{IL}	Low	V_{IN} = 0.45V		–20	–100	µA
I_{IH}	High	V_{IN} = V_{CC}		1	40	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[8]	V_{OUT} = 2.7V		0	80	µA
		V_{OUT} = 0.45V		–15	–140	
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	–15	–30	–70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		170	210	mA
Capacitance						
		V_{CC} = 5V				
I_{IN}	Input	V_{IN} = 2.0V		8		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with inputs I0 – I4 = 0V, inputs I5 – I9 = 4.5V, I11 = 4.5V and I10 = 10V. For outputs B0 – B4 and for outputs B5 – B9 apply the same conditions except I11 = 0V.
5. Same conditions as Note 4 except input I11 = +10V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with inputs I0 – I11 and B0 – B9 = 0V. Part in Virgin State.
8. Leakage values are a combination of input and output leakage.
9. I_{IL} and I_{IH} limits are for dedicated inputs only (I0 – I11).

Programmable logic array (22 × 42 × 10)

PLUS173–10

AC ELECTRICAL CHARACTERISTICS

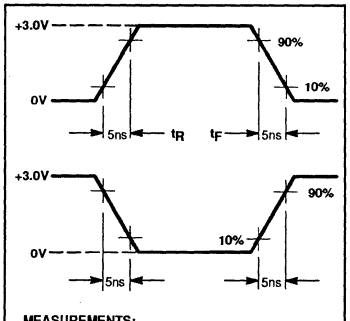
$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 300\Omega$, $R_2 = 390\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
t_{PD}	Propagation Delay[2]	Input +/–	Output +/–	C_L = 30pF		8	10	ns
t_{OE}	Output Enable[1]	Input +/–	Output –	C_L = 30pF		8	10	ns
t_{OD}	Output Disable[1]	Input +/–	Output +	C_L = 5pF		8	10	ns

NOTES:

1. For 3-State outputs; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
2. All propagation delays are measured and specified under worst case conditions.

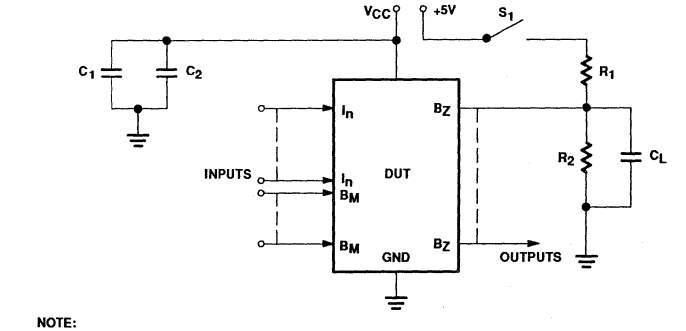
VOLTAGE WAVEFORM



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT



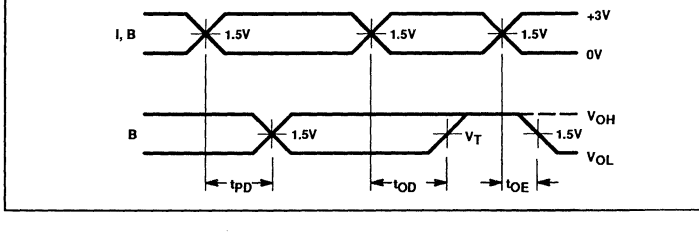
NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{PD}	Propagation delay between input and output.
t_{OD}	Delay between input change and when output is off (Hi-Z or High).
t_{OE}	Delay between input change and when output reflects specified output level.

TIMING DIAGRAM



Programmable logic array (22 × 42 × 10)

PLUS173–10

LOGIC PROGRAMMING

The PLUS173–10 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLUS173–10 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

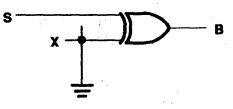
PLUS173–10 logic designs can also be generated using the program table entry format, which is detailed on the following page. This program table entry format is supported by SNAP only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

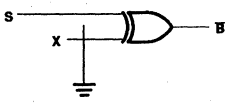
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

OUTPUT POLARITY – (B)

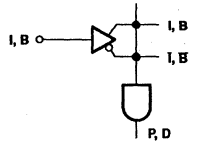


ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

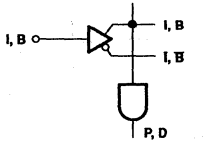


ACTIVE LEVEL	CODE
LOW (INVERTING)	L

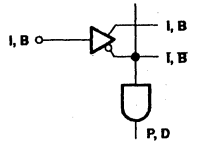
AND ARRAY – (I, B)



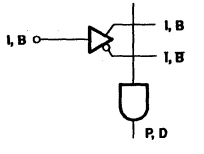
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, B	H

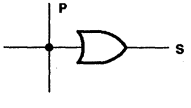


STATE	CODE
I, B̄	L

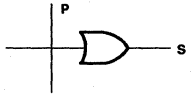


STATE	CODE
DON'T CARE	–

OR ARRAY – (B)



P_n STATUS	CODE
ACTIVE[1]	A



P_n STATUS	CODE
INACTIVE	•

VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs are at "H" polarity.
2. All P_n terms are disabled.
3. All P_n terms are active on all outputs.

NOTES:

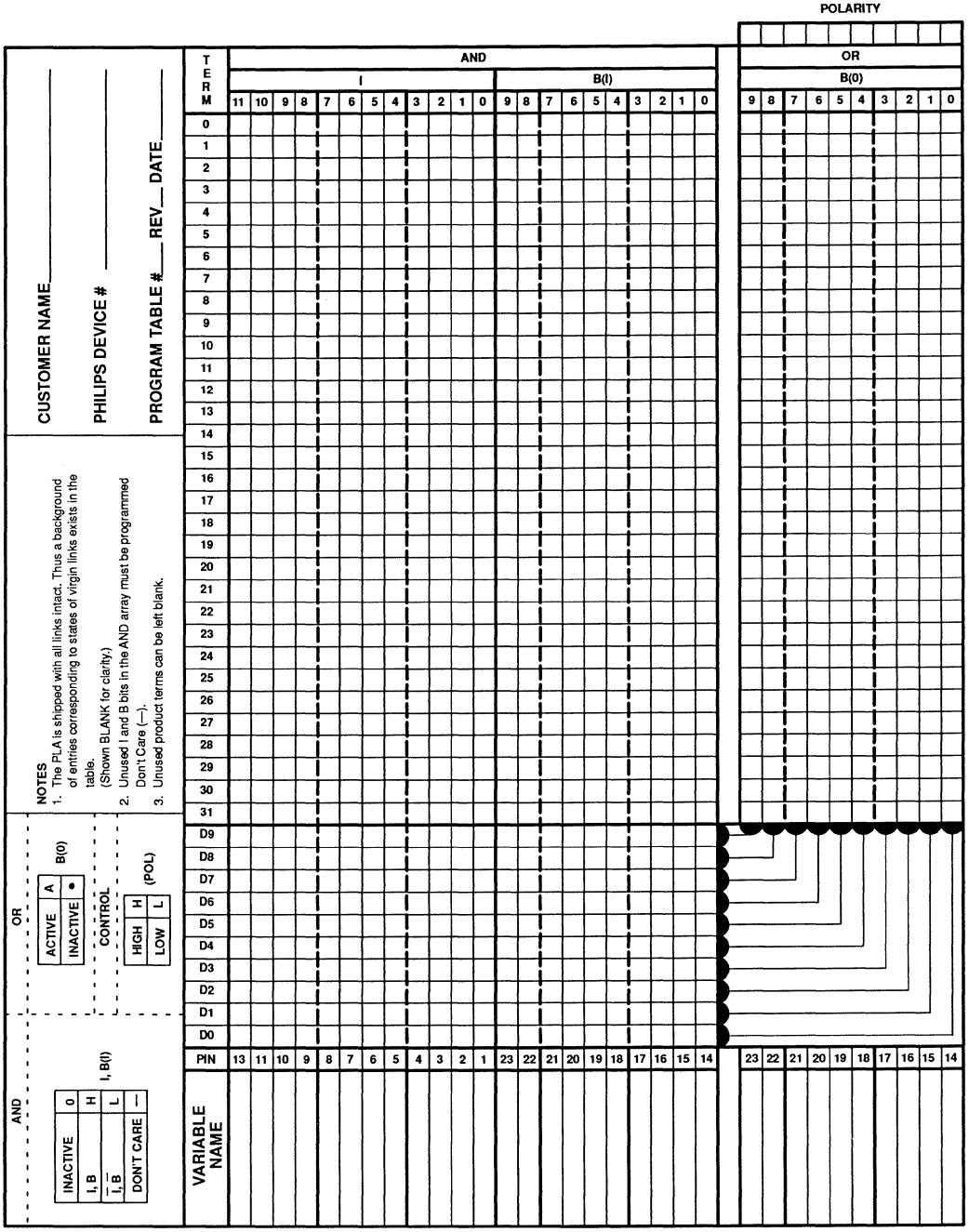
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P_n, D_n.
2. Any gate P_n, D_n will be unconditionally inhibited if both the true and complement of any input (I, B) are left intact.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

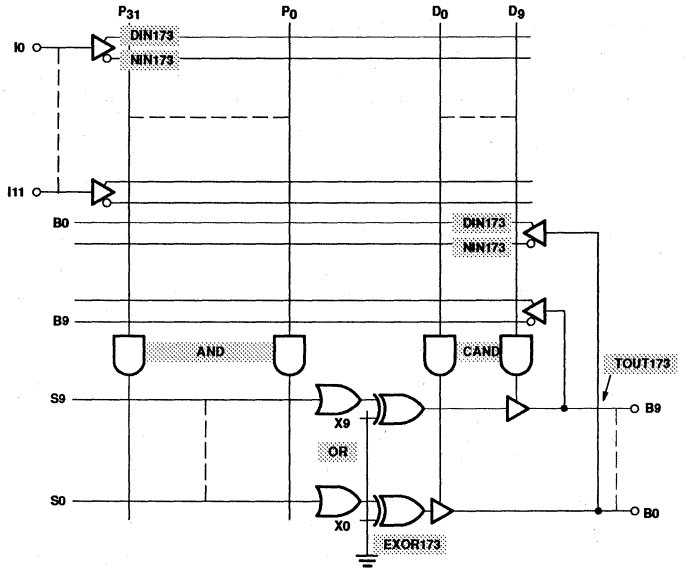
Programmable logic array (22 × 42 × 10)

PLUS173–10

PLA PROGRAM TABLE



SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

DESCRIPTION

The PLS100 (3-State) and PLS101 (Open Collector) are bipolar, fuse Programmable Logic Arrays (PLAs). Each device utilizes the standard AND/OR/Invert architecture to directly implement custom sum of product equations.

Each device consists of 16 dedicated inputs and 8 dedicated outputs. Each output is capable of being actively controlled by any or all of the 48 product terms. The True, Complement, or Don't Care condition of each of the 16 inputs and be ANDed together to comprise one P-term. All 48 P-terms can be selectively ORed to each output.

The PLS100 and PLS101 are fully TTL compatible, and chip enable control for expansion of input variables and output inhibit. They feature either Open Collector or 3-State outputs for ease of expansion of product terms and application in bus-organized systems.

Order codes are listed in the Ordering Information Table.

FEATURES

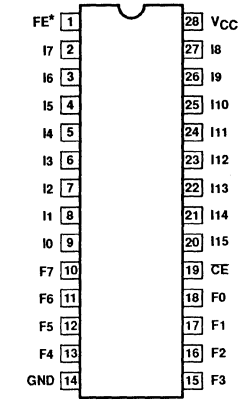
- Field-programmable (Ni-Cr link)
- Input variables: 16
- Output functions: 8
- Product terms: 48
- I/O propagation delay: 50ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading: –100μA (max.)
- Chip Enable input
- Output option:
  - PLS100: 3-State
  - PLS101: Open-Collector
- Output disable function:
  - 3-State: Hi-Z
  - Open-Collector: High

APPLICATIONS

- CRT display systems
- Code conversion
- Peripheral controllers
- Function generators
- Look-up and decision tables
- Microprogramming
- Address mapping
- Character generators
- Data security encoders
- Fault detectors
- Frequency synthesizers
- 16-bit to 8-bit bus interface
- Random logic replacement

PIN CONFIGURATIONS

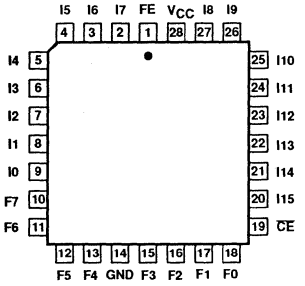
N Package



* Fuse Enable Pin: It is recommended that this pin be left open or connected to ground during normal operation.

N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

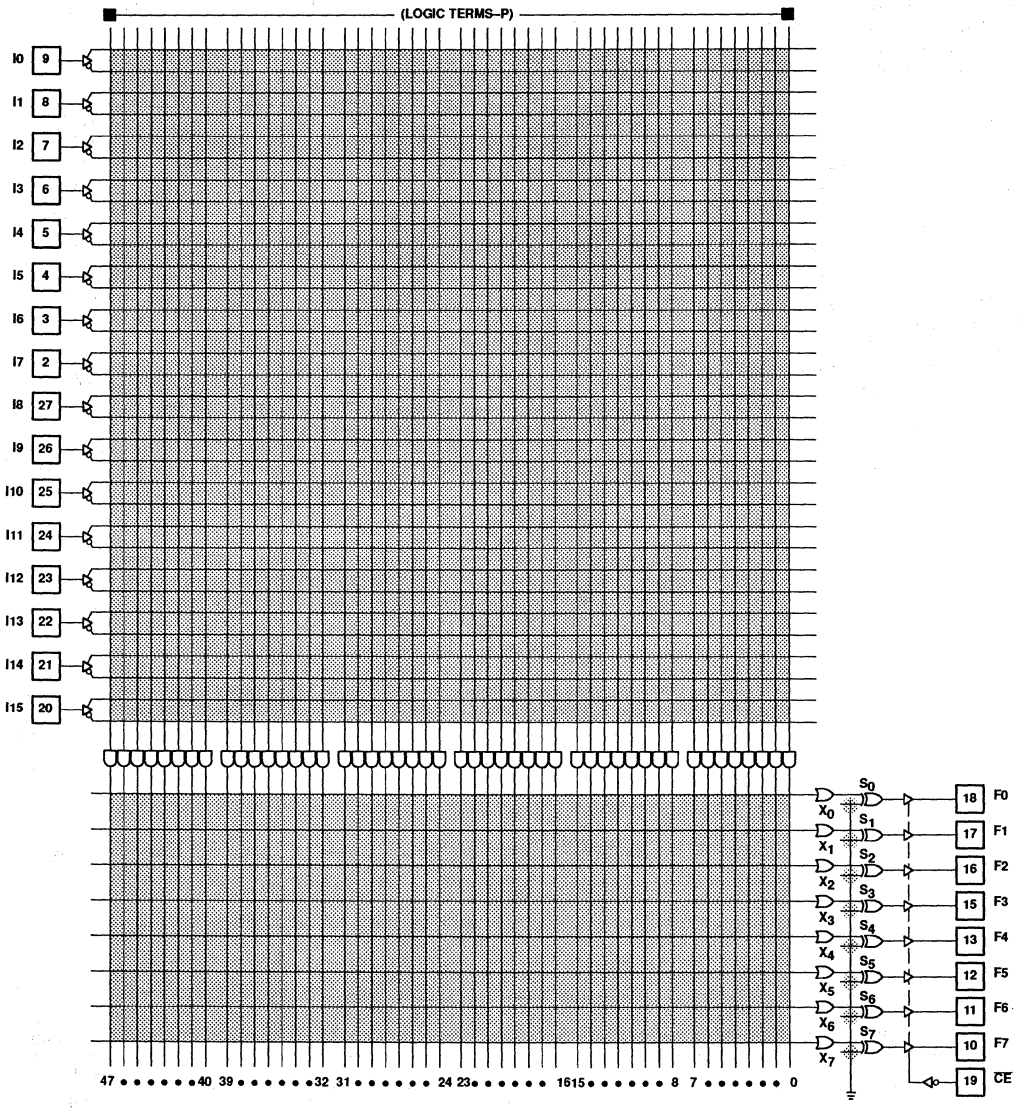
ORDERING INFORMATION

DESCRIPTION	3-STATE	OPEN COLLECTOR	DRAWING NUMBER
28-Pin Plastic Dual In-Line 600mil-wide	PLS100N	PLS101N	0413D
28-Pin Plastic Leaded Chip Carrier	PLS100A	PLS101A	0401F

Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

LOGIC DIAGRAM

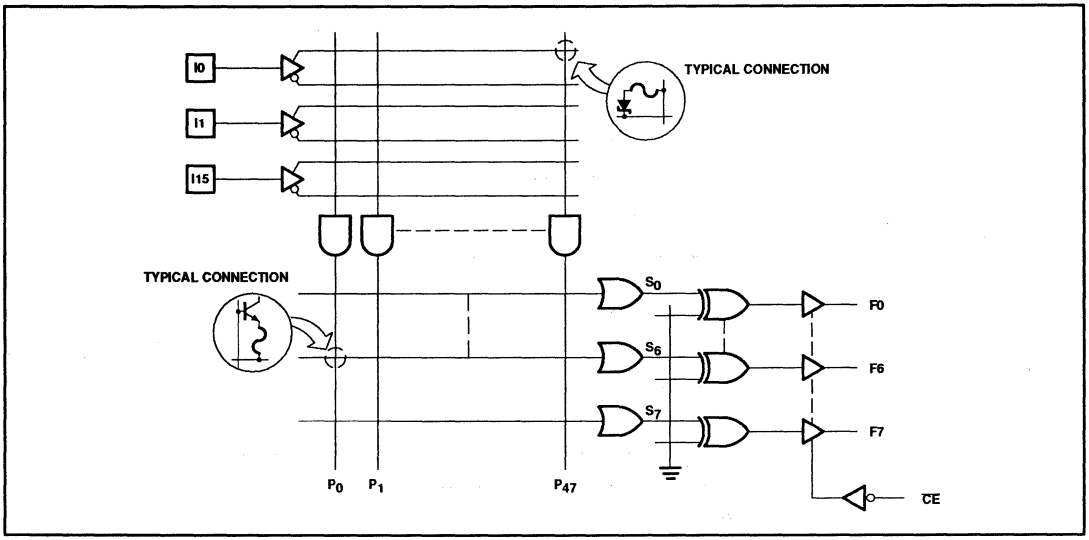


NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ⁘ Programmable connection.

Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

FUNCTIONAL DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7.0	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Output voltage	+5.5	V_{DC}
I_{IN}	Input current	±30	mA
I_{OUT}	Output current	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS100 device is also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Philips Semiconductors Military Data Handbook.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High (PLS100)[4]	I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		< 1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
$I_{O(OFF)}$	Hi-Z state (PLS100)	$\overline{CE}$ = High, V_{CC} = MAX				
		V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit (PLS100) [3, 6]	$\overline{CE}$ = Low, V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = MAX		120	170	mA
Capacitance						
		$\overline{CE}$ = High, V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		17		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one pin at a time.
4. Measured with V_{IL} applied to $\overline{CE}$ and a logic high stored.
5. Measured with a programmed logic condition for which the output test is at a low logic level. Output sink current is applied through a resistor to V_{CC}.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with the Chip Enable input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

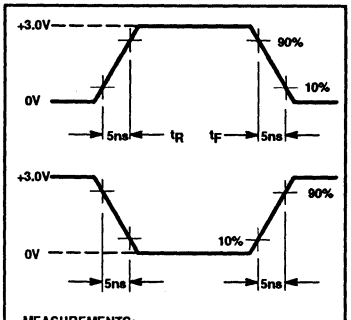
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				MIN	TYP[1]	MAX	
Propagation delay[2]							
t_{PD}	Input	Output	Input		35	50	ns
t_{CE}	Chip Enable[3]	Output	Chip Enable		15	30	ns
Disable time							
t_{CD}	Chip Disable[3]	Output	Chip Enable		15	30	ns

NOTES:
1. All typical values are at $V_{CC} = 5V$. $T_{amb} = +25°C$.
2. All propagation delays are measured and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

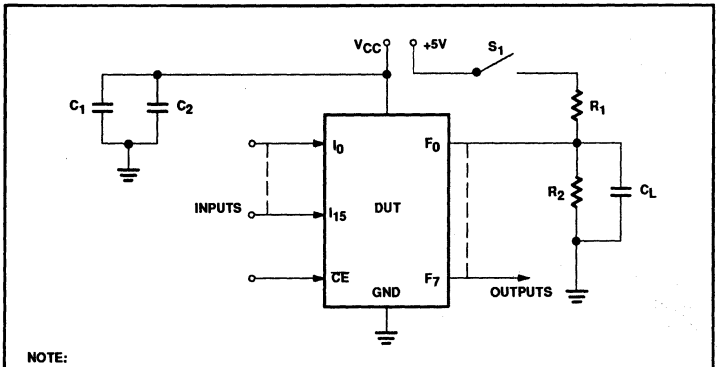
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

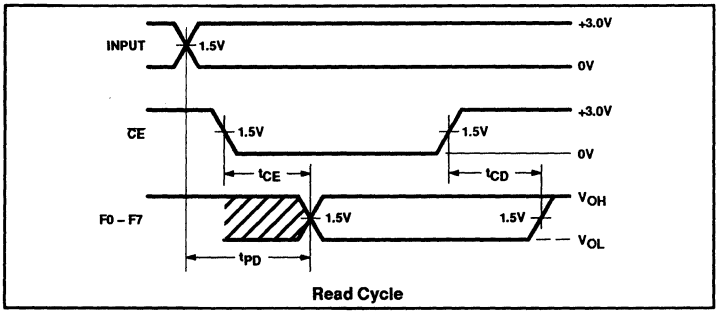


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CE}	Delay between beginning of Chip Enable Low (with Input valid) and when Data Output becomes valid.
t_{CD}	Delay between when Chip Enable becomes High and Data Output is in off state (Hi-Z or High).
t_{PD}	Delay between beginning of valid Input (with Chip Enable Low) and when Data Output becomes valid.

TIMING DIAGRAM



Read Cycle

Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

LOGIC PROGRAMMING

PLS100/PLS101 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

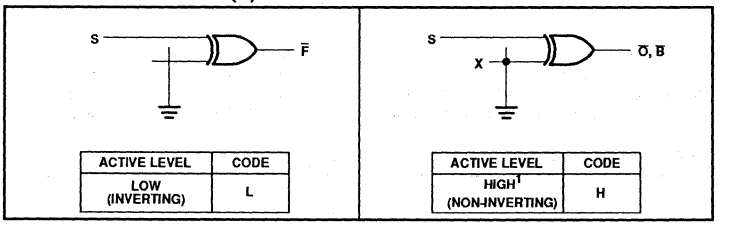
PLS100/PLS101 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors' SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The sumbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this dat handbook for additional informational.

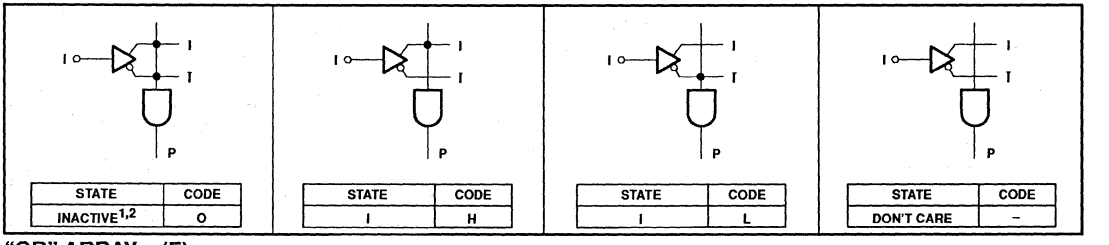
OUTPUT POLARITY – (F)



ACTIVE LEVEL	CODE
LOW (INVERTING)	L

ACTIVE LEVEL	CODE
HIGH[1] (NON-INVERTING)	H

"AND" ARRAY – (I)



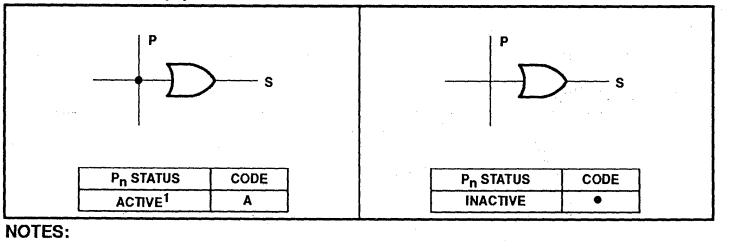
STATE	CODE
INACTIVE[1,2]	O

STATE	CODE
I	H

STATE	CODE
I	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (F)



P_n STATUS	CODE
ACTIVE[1]	A

P_n STATUS	CODE
INACTIVE	•

NOTES:

1. This is the initial unprogrammed state of all links. It is normally associated with all unused (inactive) AND gates P_n.
2. Any gate P_n will be unconditionally inhibited if any one of its (I) link pairs is left intact.

VIRGIN STATE

The PLS100/101 virgin devices are factory shipped in an unprogrammed state, with all fuses intact, such that:

1. All P_n terms are disabled (inactive) in the AND array.
2. All P_n terms are active in the OR array.
3. All outputs are Active-High.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic arrays (16 × 48 × 8)

PLS100/PLS101

PROGRAM TABLE

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	Im	Don't Care	Prod. Term Present in Fp	Prod. Term Not Present in Fp	Active High	Active Low
H	L	– (dash)	A	• (period)	H	L
NOTE Enter (–) for unused inputs of used P-terms.			NOTES 1. Entries independent of output polarity. 2. Enter (A) for unused outputs of used P-terms.		NOTES 1. Polarity programmed once only. 2. Enter (H) for all unused outputs.	

CUSTOMER NAME _____

PURCHASE ORDER # _____

PHILIPS DEVICE # _____ CF (XXXX)

CUSTOMER SYMBOLIZED PART # _____

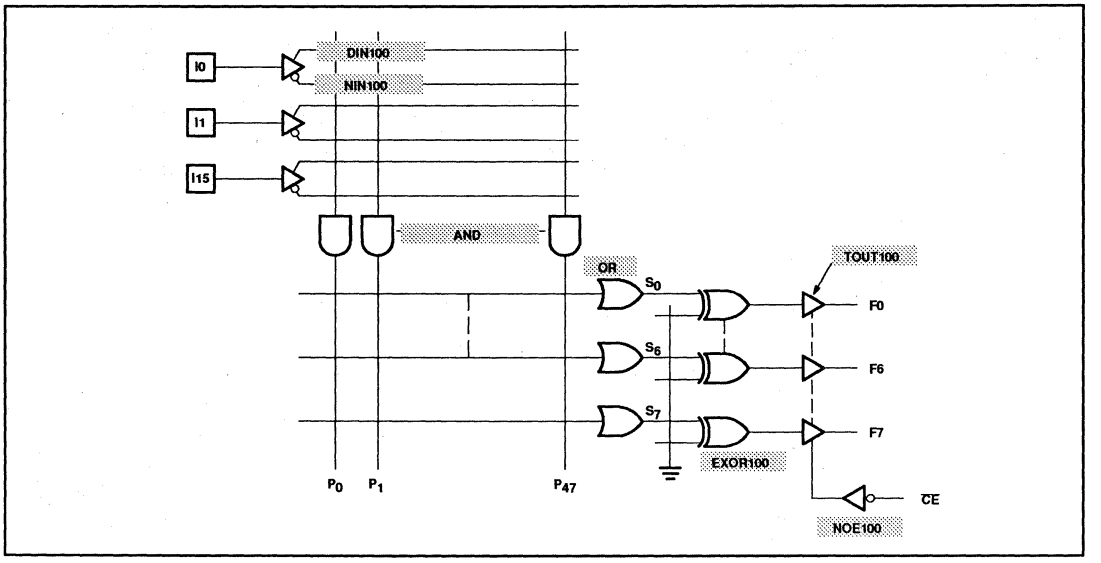
TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____ REV ____ DATE ____

	AND															
TERM	INPUT (I_m)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
PIN NO.	20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9
VARIABLE NAME																

POLARITY							
OR OUTPUT (F_P)							
7	6	5	4	3	2	1	0
10	11	12	13	15	16	17	18

SNAP RESOURCE SUMMARY DESIGNATIONS



Section 5
Programmable Logic Sequencer Devices

CONTENTS

Programmable logic sequencer ($16 \times 45 \times 12$)

PLS155

DESCRIPTION

The PLS155 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 4 registered I/O outputs (F) in conjunction with 8 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the AND gates can drive the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS155 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed below.

FEATURES

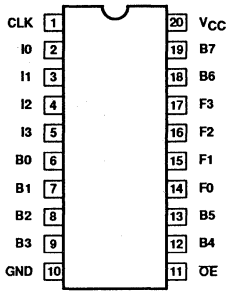
- f_{MAX} = 14MHz
  - 18.2MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 8 bidirectional I/O lines
- 4 bidirectional registers
- J-K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: –100µA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

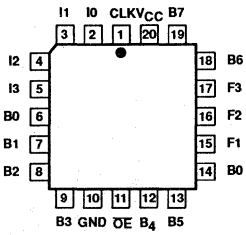
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

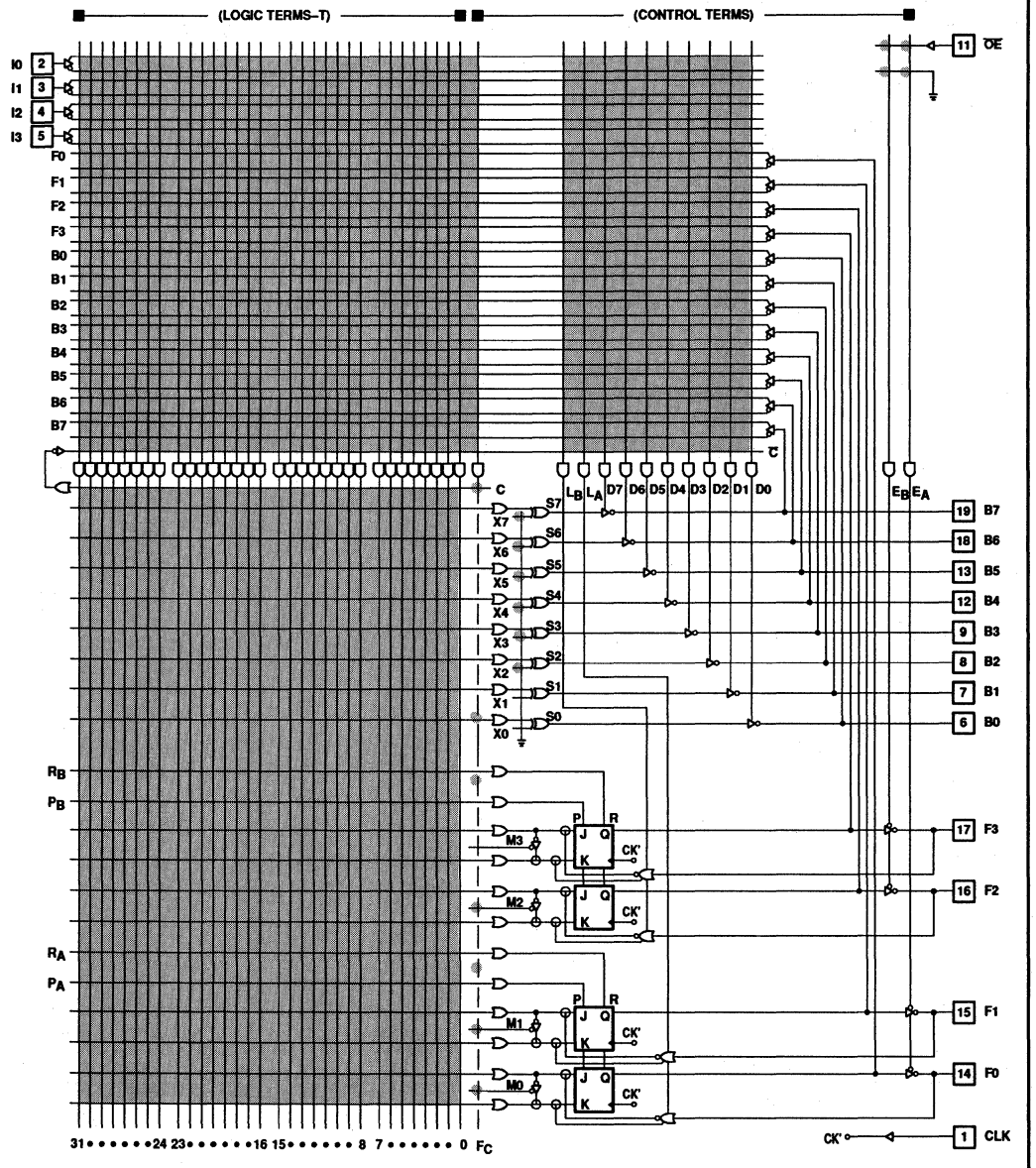
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual In-Line Package (300mil-wide)	PLS155N	0408D
20-Pin Plastic Leaded Chip Carrier	PLS155A	0400E

Programmable logic sequencer (16 × 45 × 12)

PLS155

LOGIC DIAGRAM

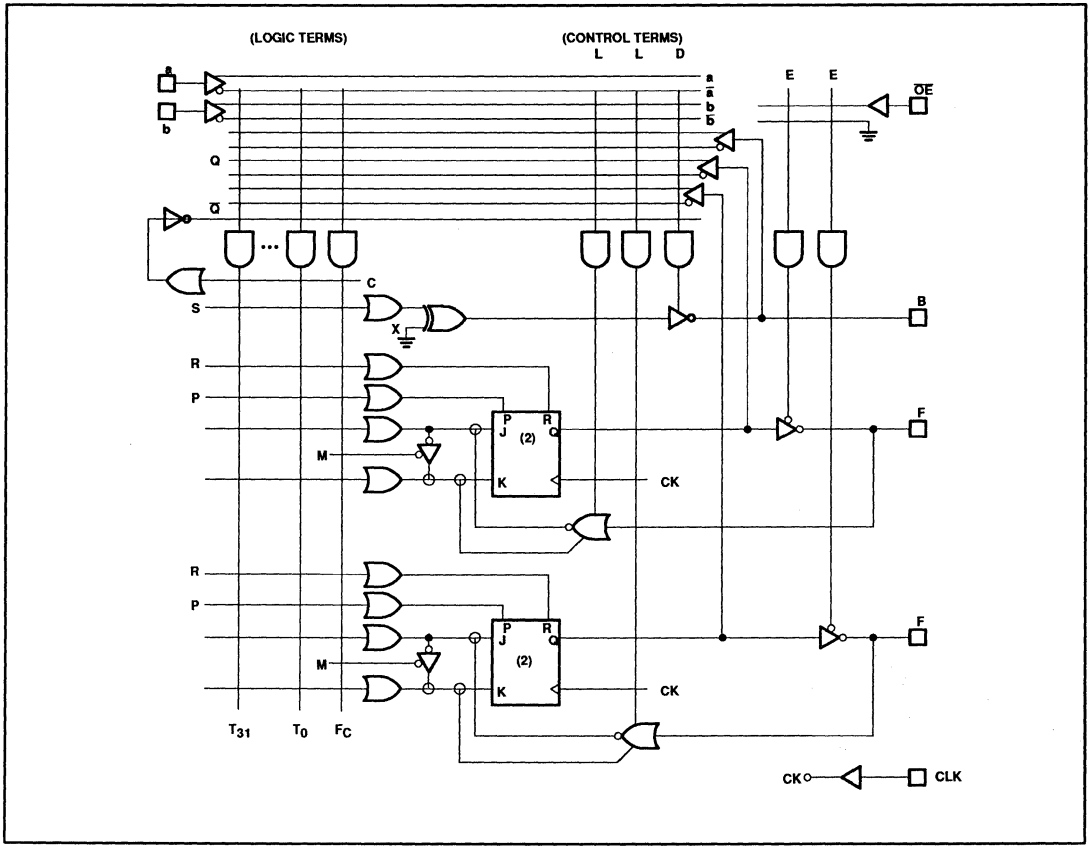


NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ● Programmable connection.

Programmable logic sequencer (16 × 45 × 12)

PLS155

FUNCTIONAL DIAGRAM



FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								Hi-Z
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

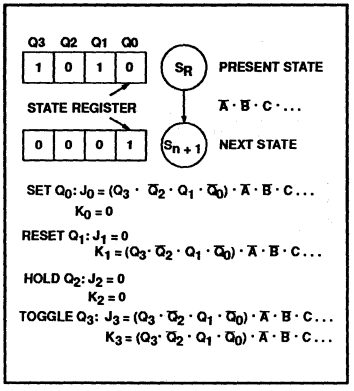
NOTES:

1. Positive Logic:
   $J\text{-}K = T_0 + T_1 + T_2 \ldots\ldots\ldots\ldots T_{31}$
   $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. * * = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

Programmable logic sequencer (16 × 45 × 12)

PLS155

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS Min	RATINGS Max	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current[5]						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		−1	−140	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[4]	V_{CC} = MAX		150	190	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
5. Leakage values are a combination of input and output leakage.
6. Measured with V_{IH} applied to $\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.

Programmable logic sequencer (16 × 45 × 12)

PLS155

AC ELECTRICAL CHARACTERISTICS

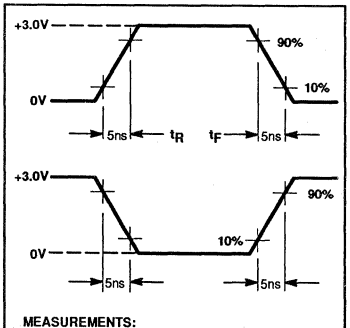
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	25	20		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	30	20		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	70	50		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	40	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	40	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	20	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	65	40		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–10		ns
t_{IH2}	Input	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delays								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		25	30	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		40	50	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		35	55	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		30	35	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		50	55	ns

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 95ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.
6. For test circuits, waveforms and timing diagrams see the following pages.

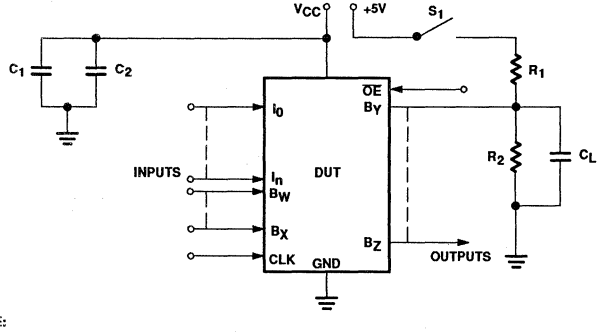
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

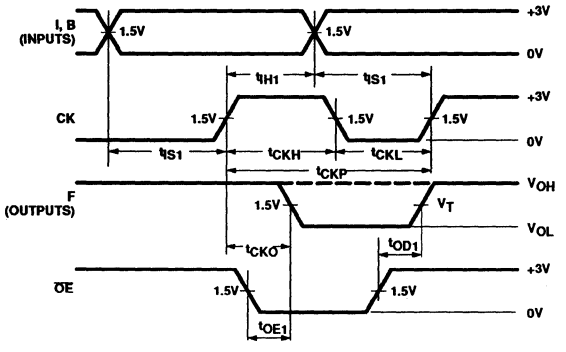
Input Pulses

TEST LOAD CIRCUIT

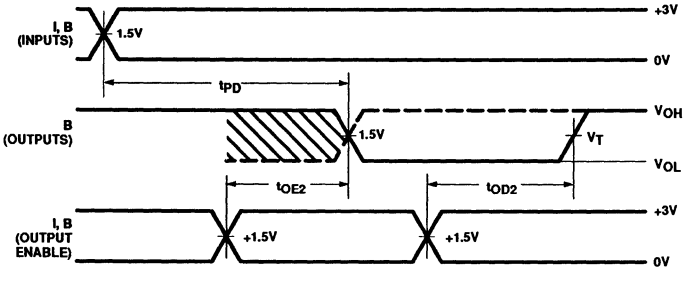


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

TIMING DIAGRAMS



Flip-Flop Outputs

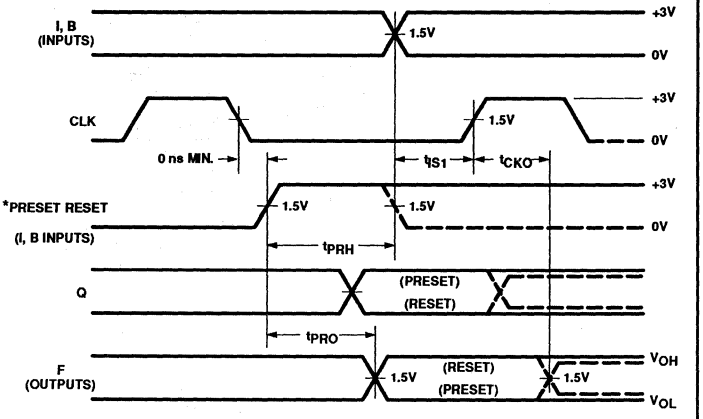


Gate Outputs

TIMING DEFINITIONS

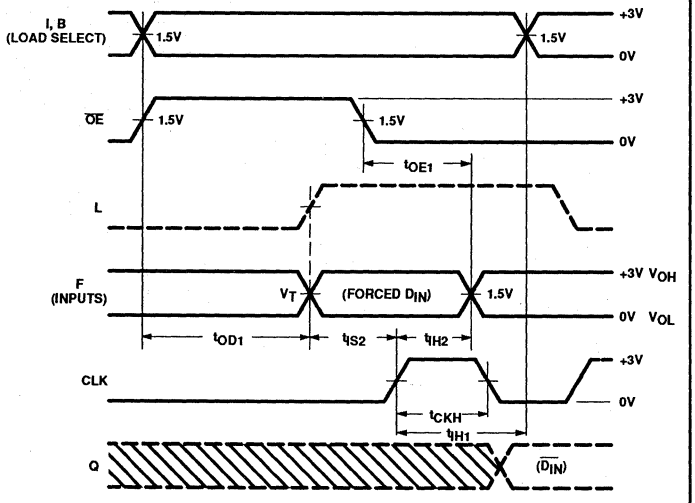
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



* The leading edge of preset/reset must occur only when the input clock is "low", and must remain "high" as long as required to override clock. The falling edge of preset/reset can never go "low" when the input clock is "high".

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (16 × 45 × 12)

PLS155

LOGIC PROGRAMMING

The PLS155 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips' Semiconductors SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

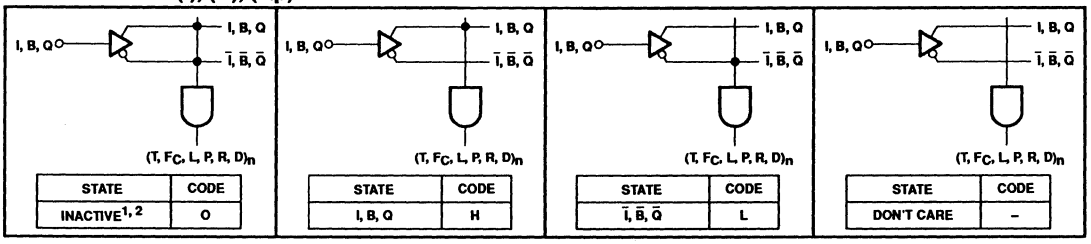
PLS155 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I), (B), (Qp)



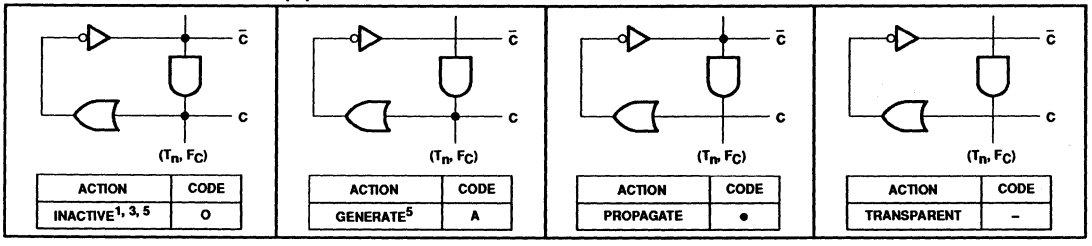
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



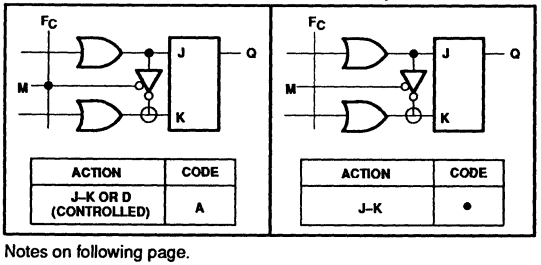
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

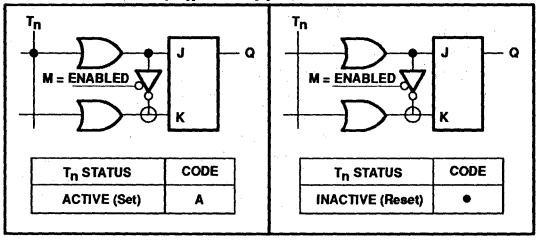
ACTION	CODE
J–K	•

Notes on following page.

Programmable logic sequencer (16 × 45 × 12)

PLS155

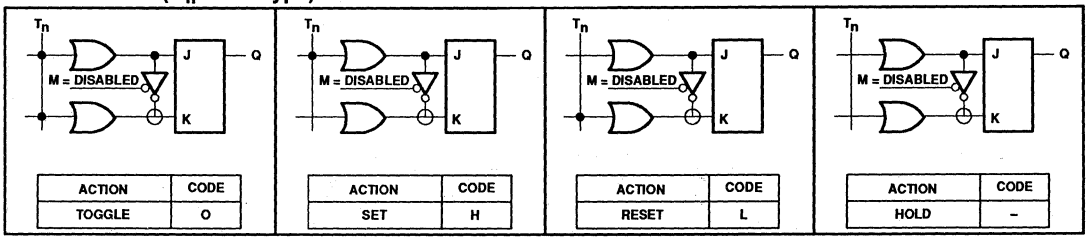
"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)	A

T_n STATUS	CODE
INACTIVE (Reset)	•

"OR" ARRAY – (Q_n = J-K Type)



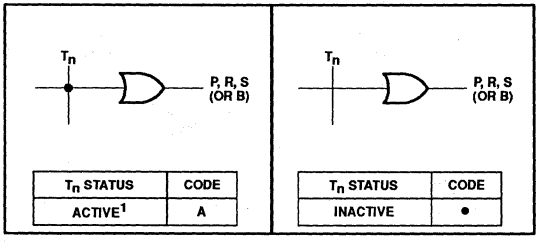
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

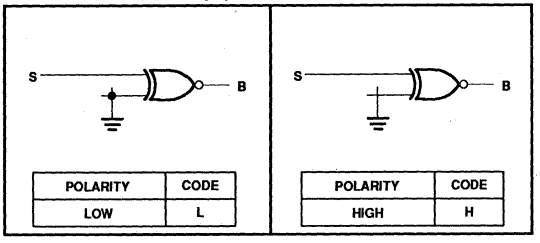
"OR" ARRAY – (S or B), (P), (R)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

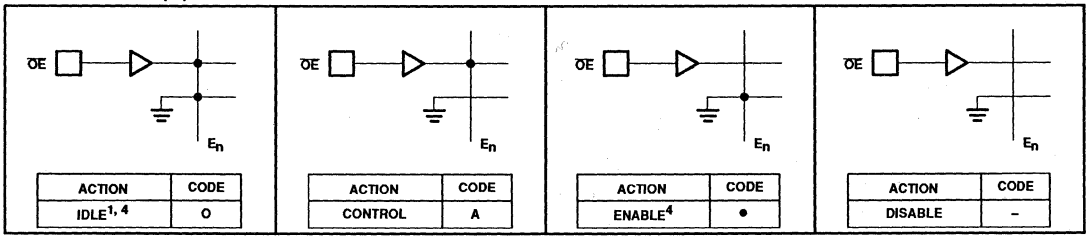
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

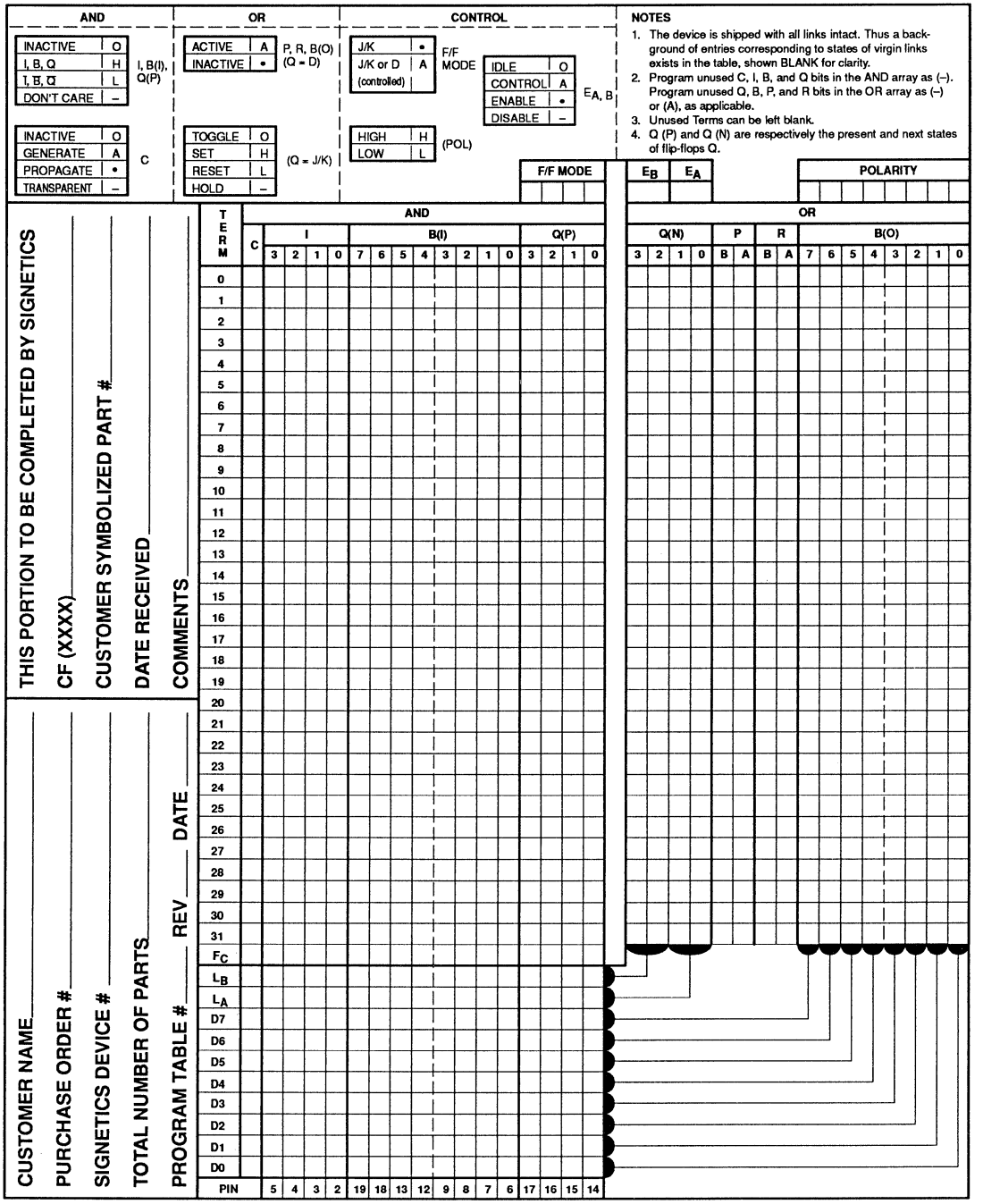
ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate (T, F_C, L, P, R, $D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates (L, P, R, $D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12)

PLS155

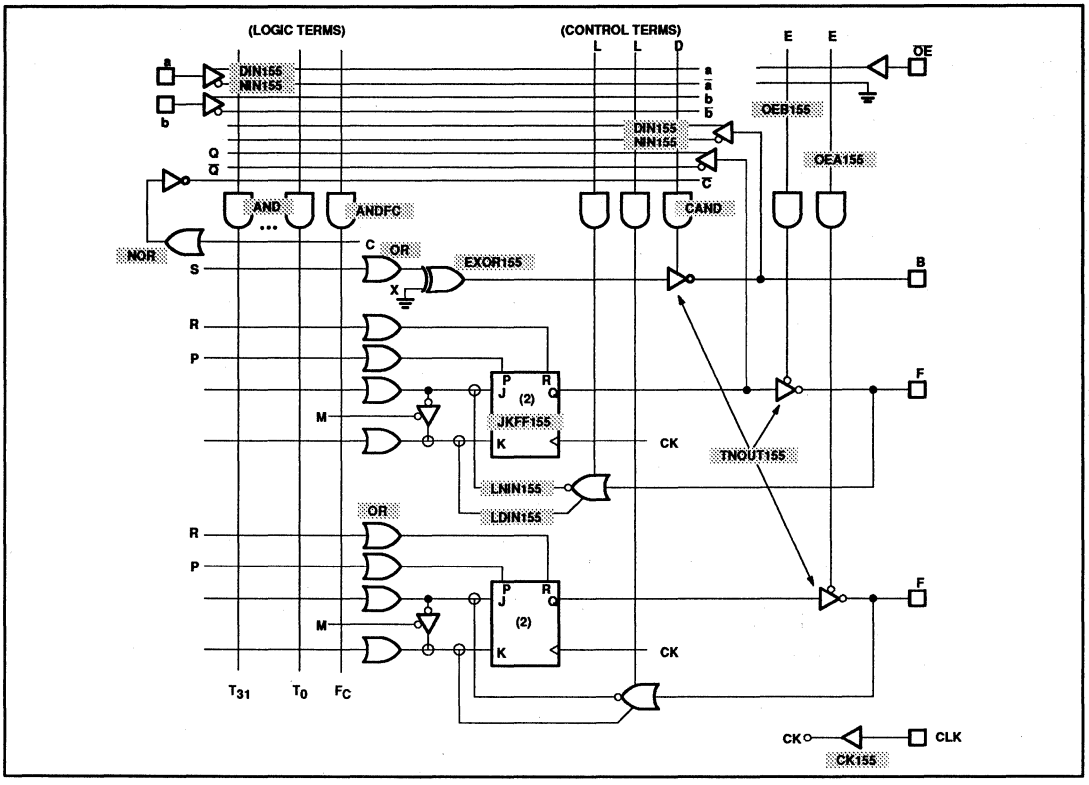
PROGRAM TABLE



Programmable logic sequencer (16 × 45 × 12)

PLS155

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 45 × 12)

PLS157

DESCRIPTION

The PLS157 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 6 registered I/O outputs (F) in conjunction with 6 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), are driven from the AND array for 4 of the 8 registers. The Preset and Reset lines (P, R) controlling the lower four registers are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS157 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed below.

FEATURES

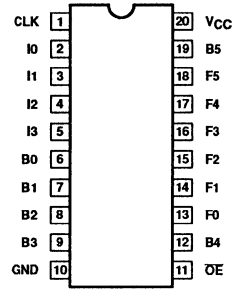
- f_{MAX} = 14MHz
  - 18.2MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 6 bidirectional I/O lines
- 6 bidirectional registers
- J-K, T, or D-type flip-flops
- 3-State outputs
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: –100µA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible

APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

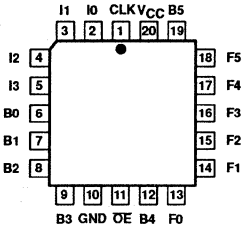
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

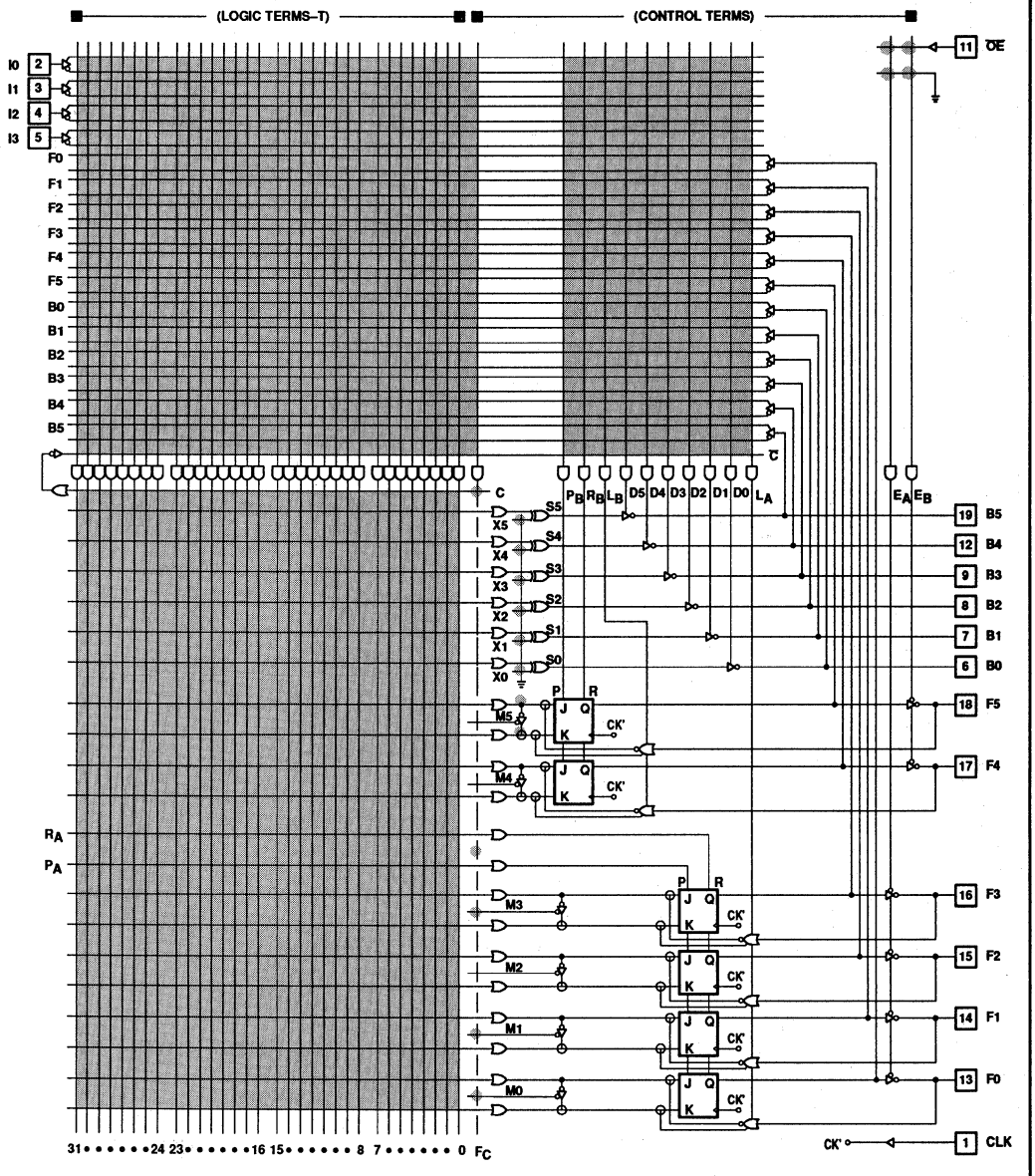
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual In-Line Package (300mil-wide)	PLS157N	0408B
20-Pin Plastic Leaded Chip Carrier	PLS157A	0400E

Programmable logic sequencer (16 × 45 × 12)

PLS157

LOGIC DIAGRAM

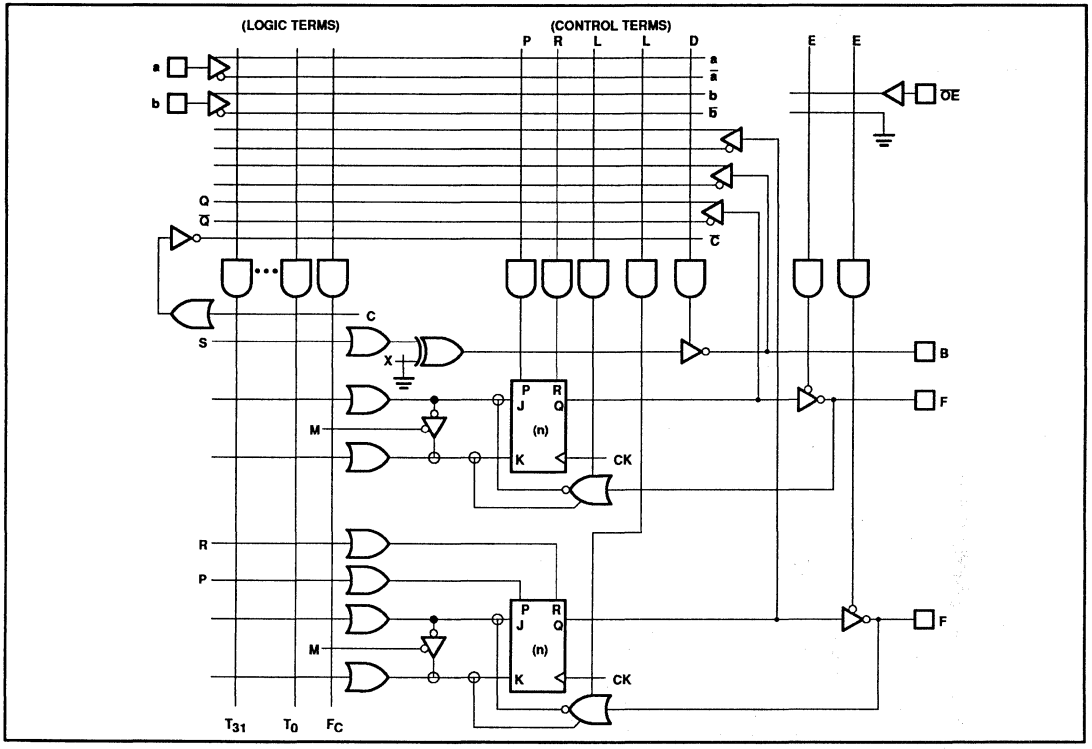


NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ▒ Programmable connection.

Programmable logic sequencer (16 x 45 x 12)

PLS157

FUNCTIONAL DIAGRAM

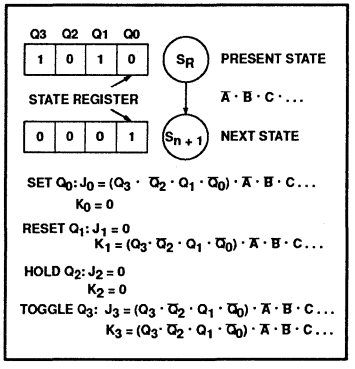


VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

Programmable logic sequencer (16 × 45 × 12)

PLS157

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								HI-Z
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H* *
	X	↑	X	X	H	L	H	L* *

NOTES:
1. Positive Logic: $\overline{J\text{-}K} = \overline{T_0} + \overline{T_1} + \overline{T_2} \ldots\ldots\ldots\ldots \overline{T_{31}}$
   $\overline{T_n} = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B_0 \cdot B_1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. * * = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		−1	−140	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[4]	V_{CC} = MAX		150	190	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
5. Leakage values are a combination of input and output leakage.
6. Measured with V_{IH} applied to $\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.

Programmable logic sequencer (16 × 45 × 12)

PLS157

AC ELECTRICAL CHARACTERISTICS

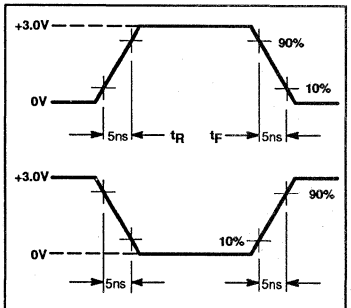
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	25	20		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	30	20		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	70	50		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	40	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	40	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	20	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	65	40		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–10		ns
t_{IH2}	Input	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delays								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		25	30	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		40	50	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		35	55	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		30	35	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		50	55	ns

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 95ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

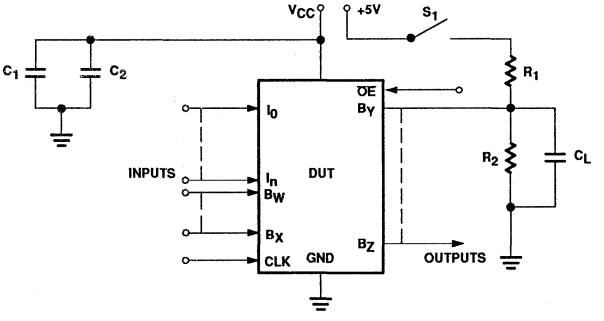
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

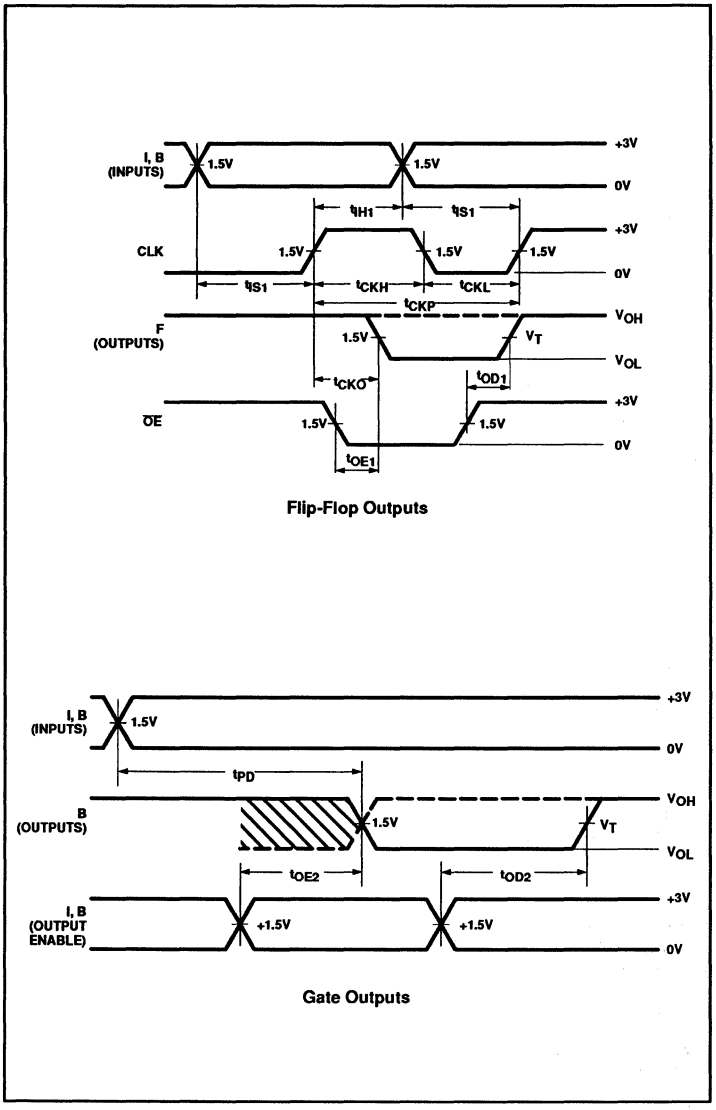
Input Pulses

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

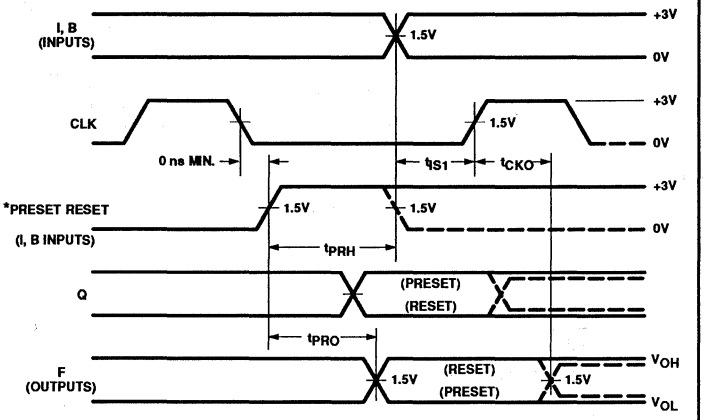
TIMING DIAGRAMS



TIMING DEFINITIONS

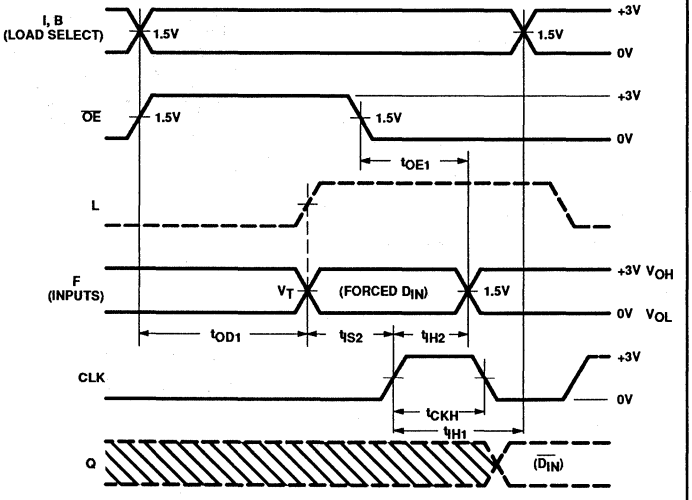
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



* The leading edge of preset/reset must occur only when the input clock is "low", and must remain "high" as long as required to override clock. The falling edge of preset/reset can never go "low" when the input clock is "high".

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (16 × 45 × 12)

PLS157

LOGIC PROGRAMMING

The PLS157 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

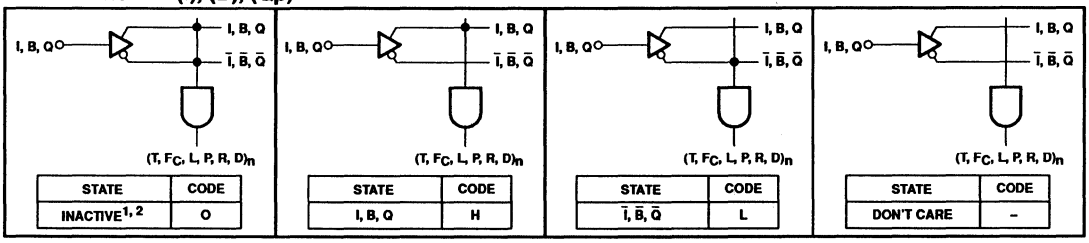
PLS157 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package only.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I), (B), (Qp)



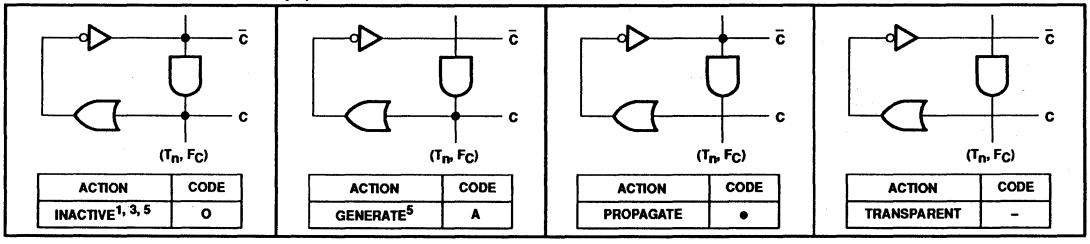
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
$\bar{I}, \bar{B}, \bar{Q}$	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



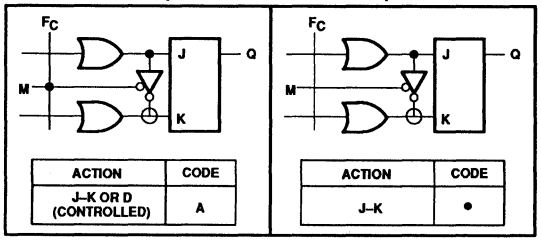
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

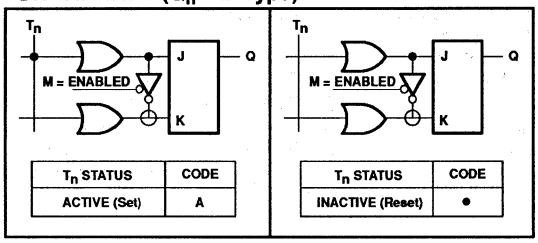
ACTION	CODE
J–K	•

Notes on following page.

Programmable logic sequencer (16 × 45 × 12)

PLS157

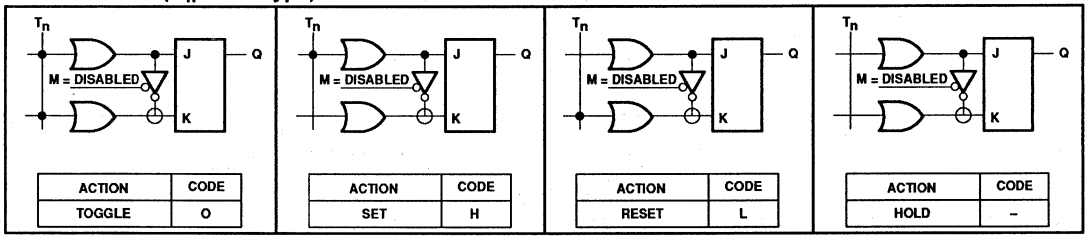
"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)	A

T_n STATUS	CODE
INACTIVE (Reset)	•

"OR" ARRAY – (Q_n = J-K Type)



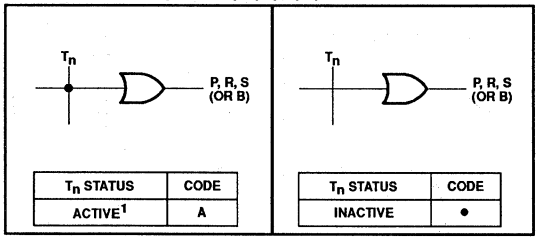
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

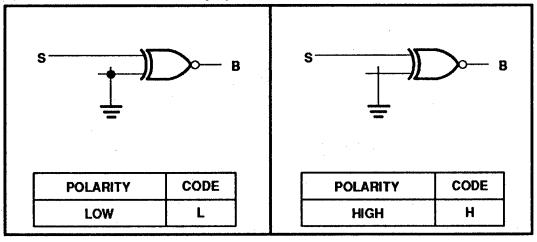
"OR" ARRAY – (S or B), (P), (R)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

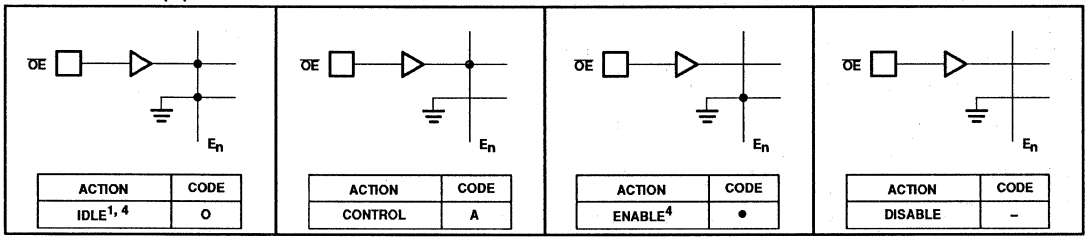
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

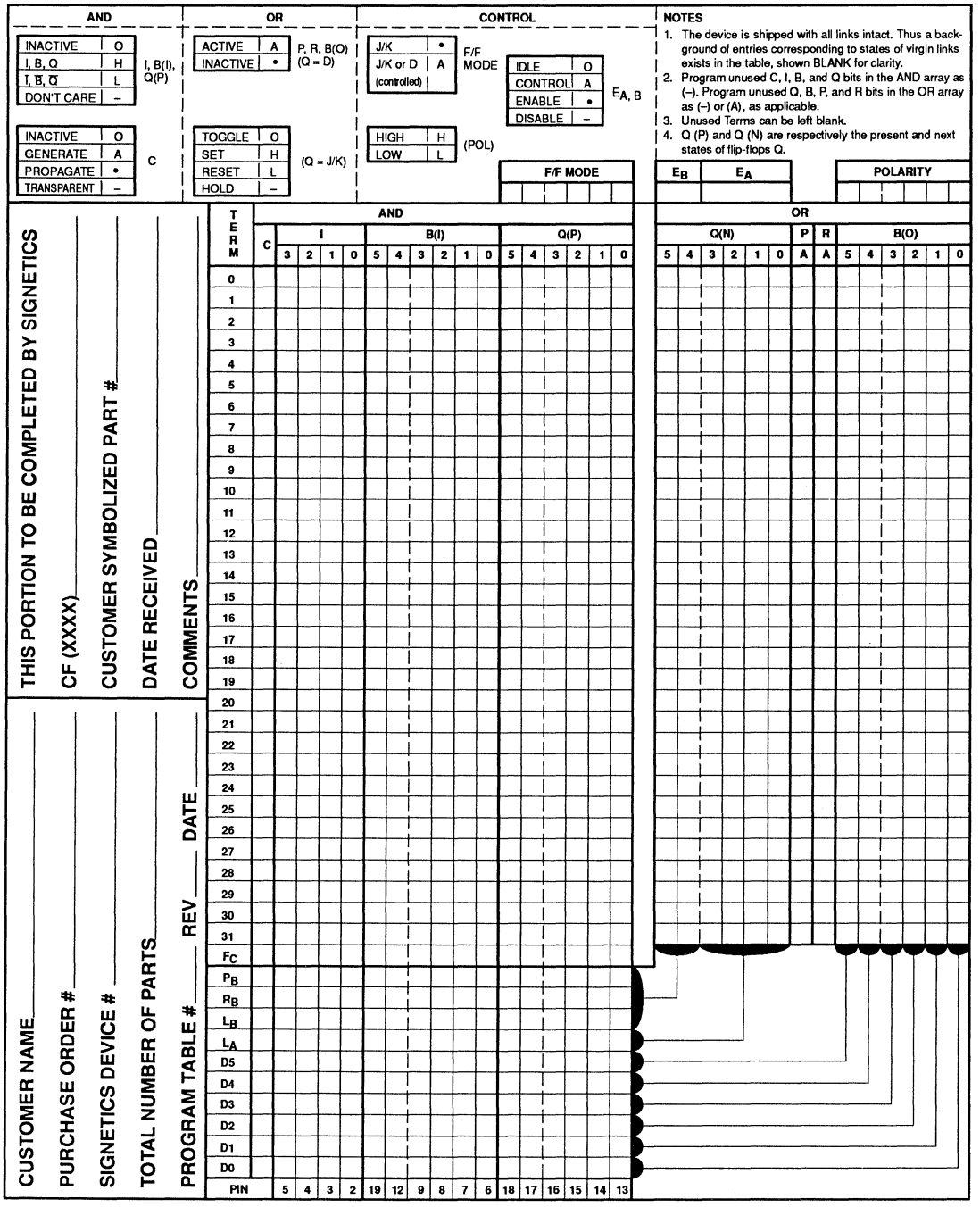
ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate (T, F_C, L, P, R, $D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates (L, P, R, $D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12)

PLS157

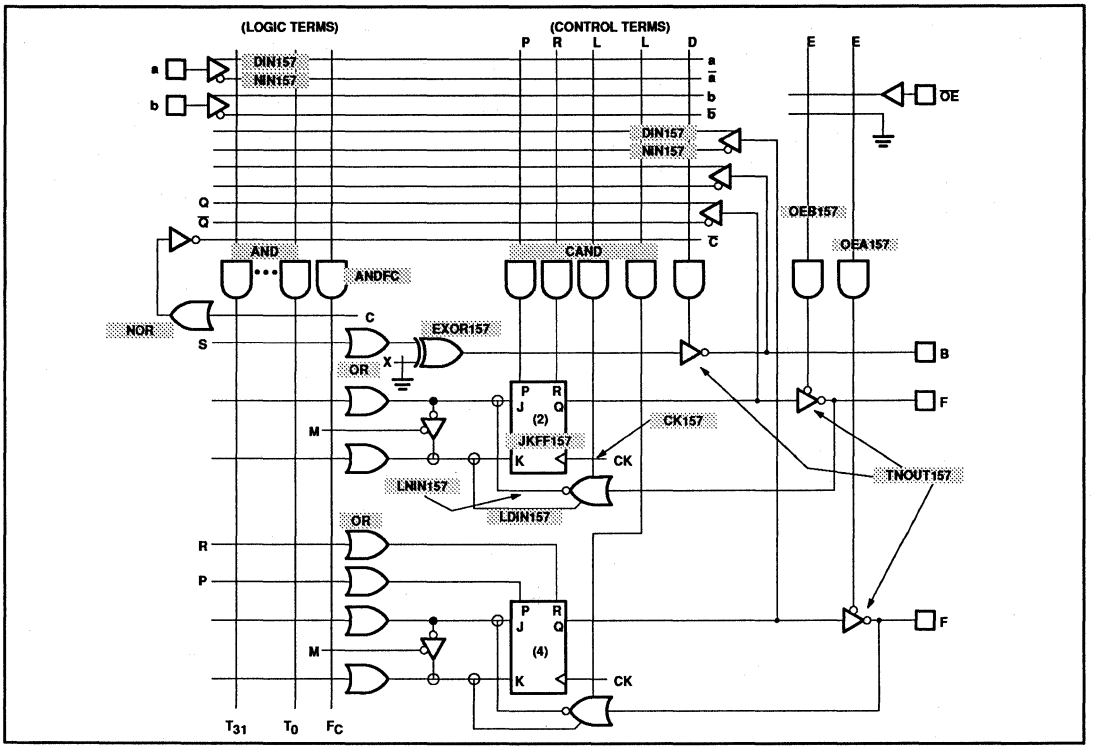
PROGRAM TABLE



Programmable logic sequencer (16 × 45 × 12)

PLS157

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 45 × 12)

PLS159A

DESCRIPTION

The PLS159A is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate F_C. It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of Ex-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. There are 4 AND gates for the Asynchronous Preset/Reset functions.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS159A is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

FEATURES

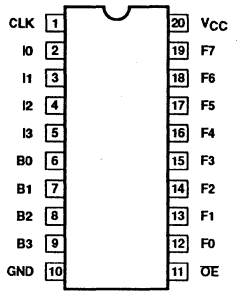
- High-speed version of PLS159
- f_{MAX} = 18MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J-K, T, or D-type flip-flops
- Power-on reset feature on all flip-flops (F_n = 1)
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Input loading: –100µA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

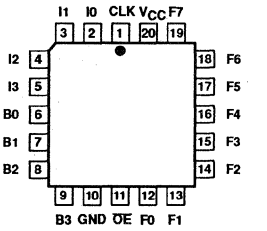
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

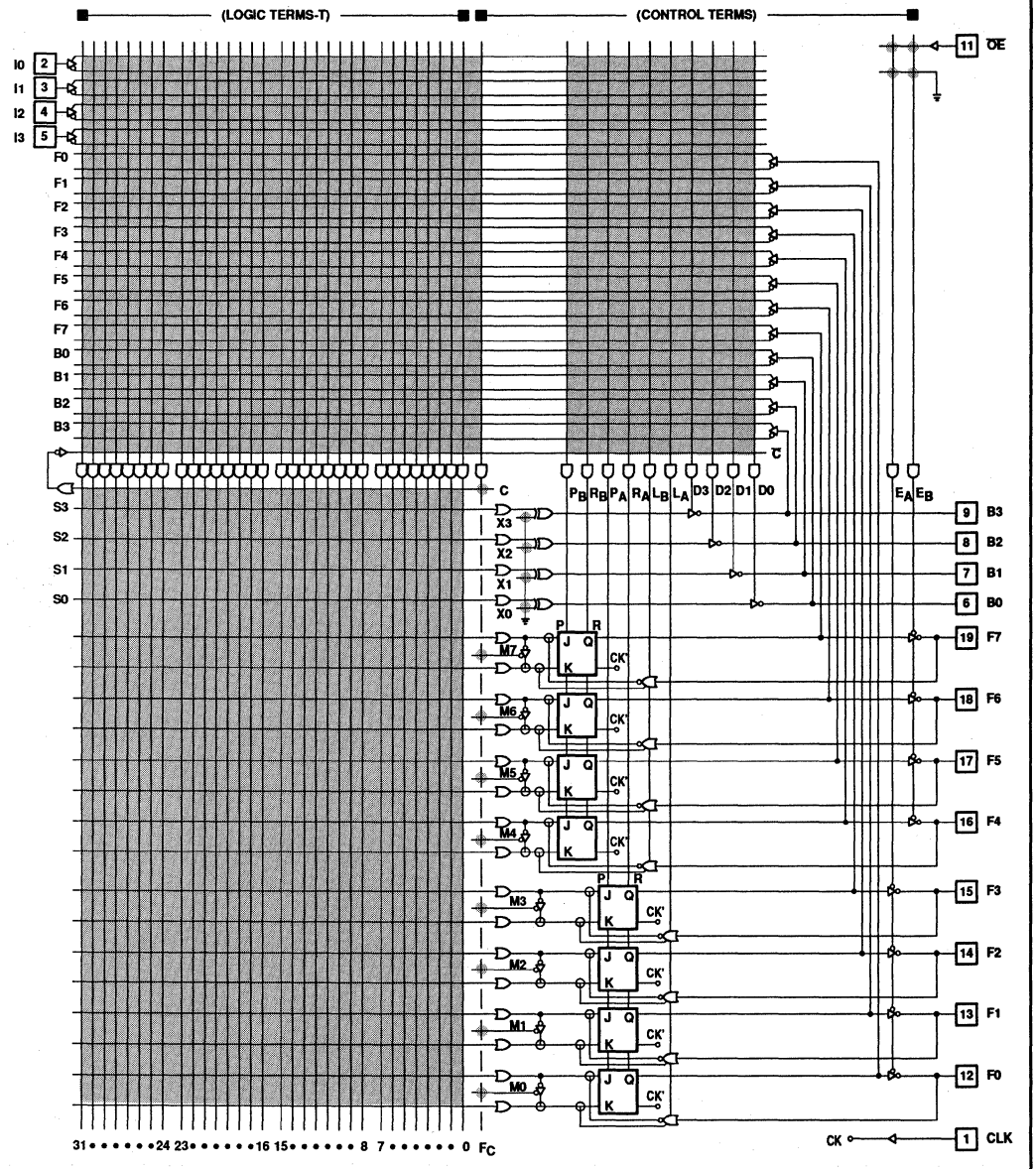
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20-Pin Plastic Dual In-Line Package (300mil-wide)	PLS159AN	0408D
20-Pin Plastic Leaded Chip Carrier	PLS159AA	0400E

Programmable logic sequencer (16 × 45 × 12)

PLS159A

LOGIC DIAGRAM

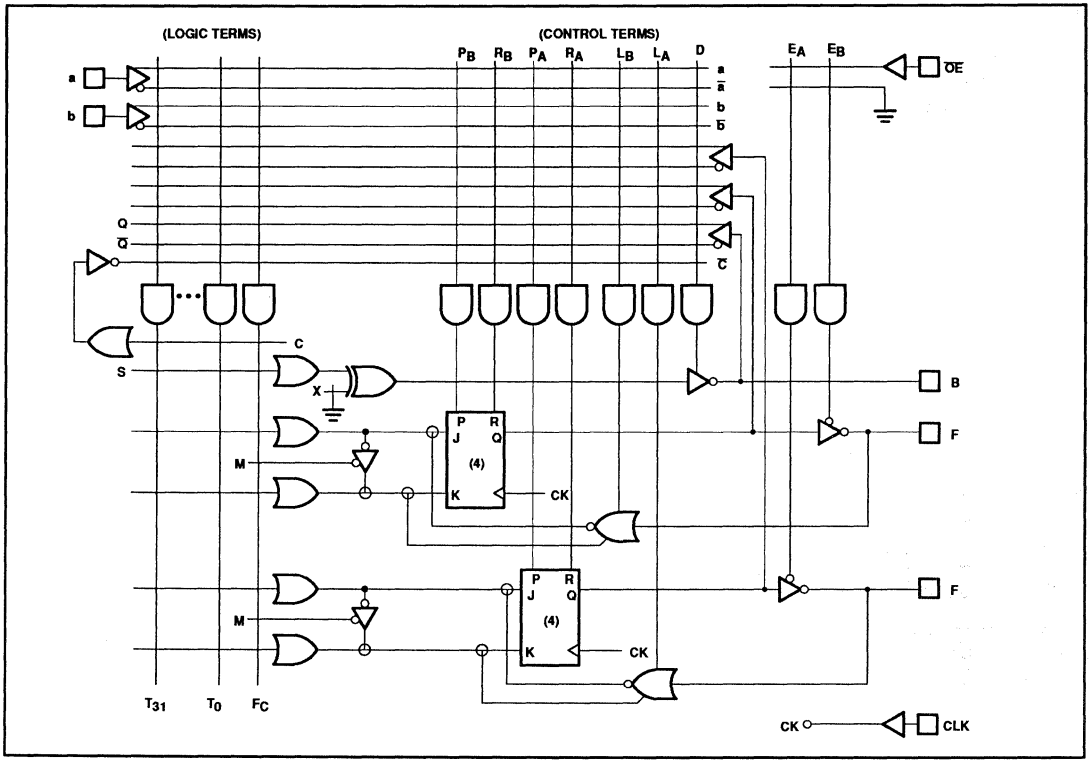


NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ● Programmable connection.

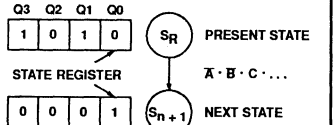
Programmable logic sequencer (16 × 45 × 12)

PLS159A

FUNCTIONAL DIAGRAM



LOGIC FUNCTION



SET Q_0: $J_0 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
$K_0 = 0$
RESET Q_1: $J_1 = 0$
$K_1 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
HOLD Q_2: $J_2 = 0$
$K_2 = 0$
TOGGLE Q_3: $J_3 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
$K_3 = (Q_3 \cdot \overline{Q}_2 \cdot Q_1 \cdot \overline{Q}_0) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$

NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								Hi-Z
L	X	X	L	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:
1. Positive Logic:
   $J\text{-}K = \overline{T}_0 + \overline{T}_1 + \overline{T}_2 \ldots\ldots\ldots\ldots \overline{T}_{31}$
   $T_n = \overline{C} \cdot (I0 \cdot \overline{I1} \cdot I2 \ldots) \cdot (Q_0 \cdot \overline{Q}_1 \ldots) \cdot (B0 \cdot B1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

Programmable logic sequencer (16 × 45 × 12)

PLS159A

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

CAUTION: PLS159A PROGRAMMING ALGORITHM

The programming voltage required to program the PLS159A is higher (17.5V) than that required to program the PLS159 (14.5V). Consequently, the PLS159 programming algorithm will not program the PLS159A. Please exercise caution when accessing programmer device codes to insure that the correct algorithm is used.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

Programmable logic sequencer (16 × 45 × 12) — PLS159A

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[4, 7]	V_{CC} = MAX, V_{OUT} = 5.5V		1	80	µA
		V_{OUT} = 0.45V		−1	−140	µA
I_{OS}	Short circuit[3, 5]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[6]	V_{CC} = MAX		150	190	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IH} applied to $\overline{OE}$.
5. Duration of short circuit should not exceed 1 second.
6. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
7. Leakage values are a combination of input and output leakage.

Programmable logic sequencer (16 × 45 × 12)

PLS159A

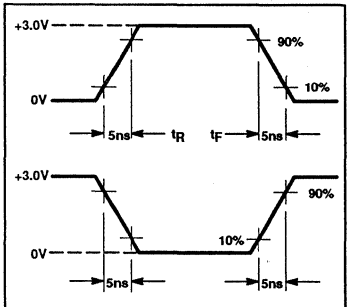
AC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	20	15		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	20	15		ns
t_{CKP}	Period	CK +	CK +	C_L = 30pF	55	45		ns
t_{PRH}	Preset/Reset pulse	(I,B) –	(I,B) +	C_L = 30pF	35	30		ns
Setup time[5]								
t_{IS1}	Input	(I,B) ±	CK +	C_L = 30pF	35	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I,B) ±	CK +	C_L = 30pF	55	45		ns
Hold time								
t_{IH1}	Input	(I,B) ±	CK +	C_L = 30pF	0	–5		ns
t_{IH2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delay								
t_{CKO}	Clock	CK +	F ±	C_L = 30pF		15	20	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I,B) ±	B ±	C_L = 30pF		25	35	ns
t_{OE2}	Output enable[3]	(I,B) +	B ±	C_L = 30pF		20	30	ns
t_{OD2}	Output disable[3]	(I,B) –	B +	C_L = 5pF		20	30	ns
t_{PRO}	Preset/Reset	(I,B) +	F ±	C_L = 30pF		35	45	ns
t_{PPR}	Power-on/preset	V_{CC} +	F –	C_L = 30pF		0	10	ns

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 75ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

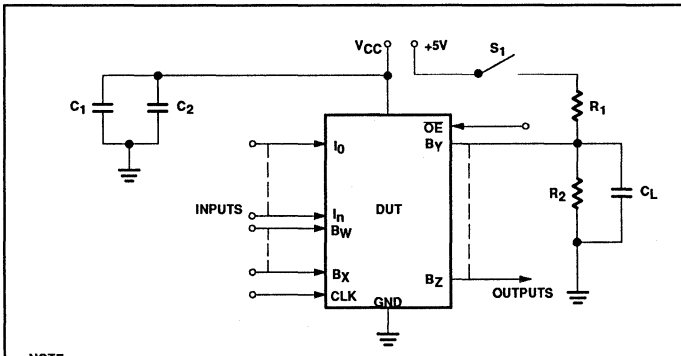
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

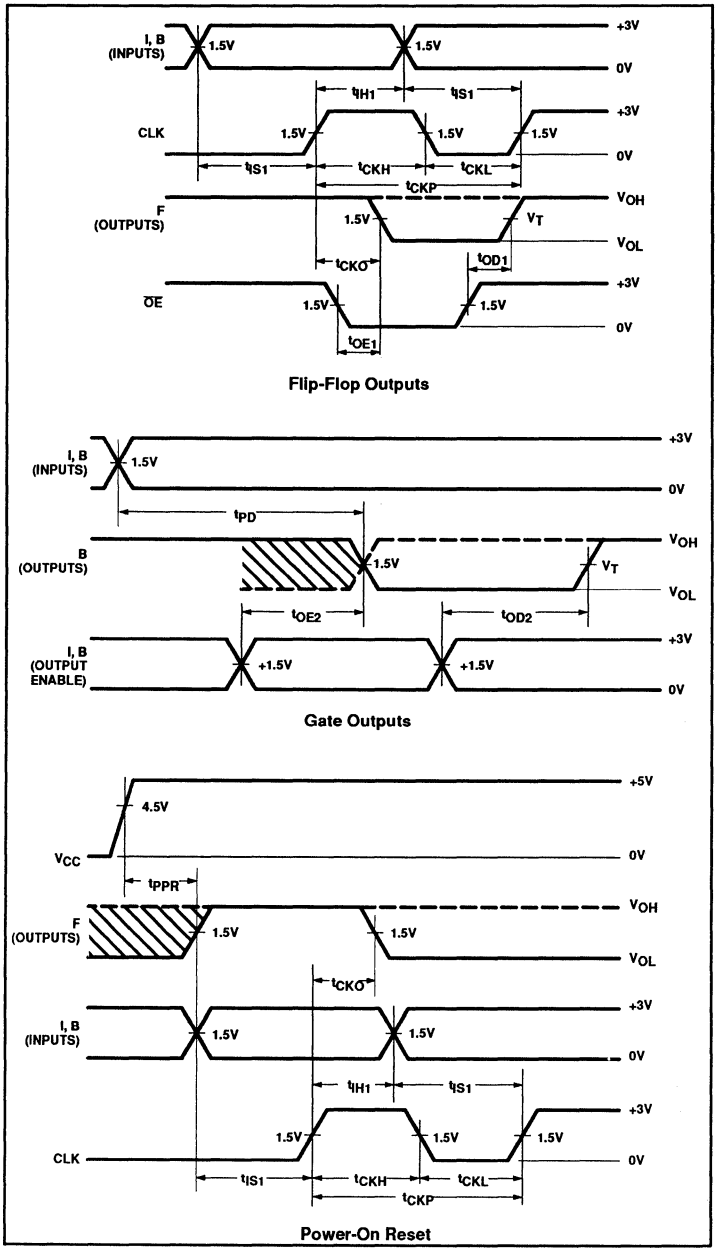


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (16 × 45 × 12)

PLS159A

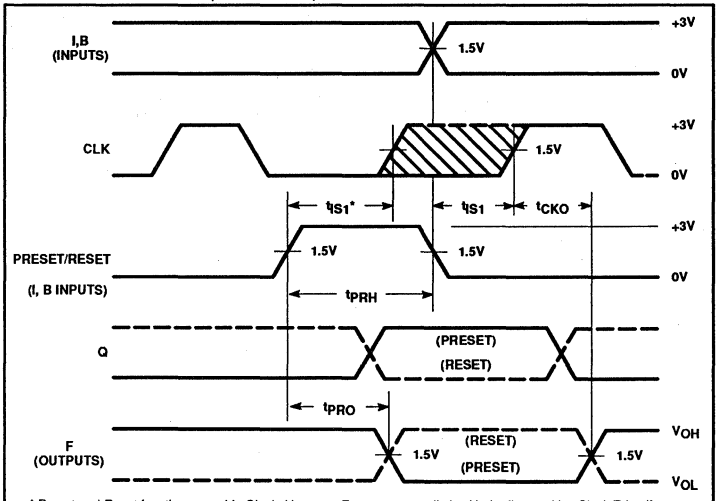
TIMING DIAGRAMS



TIMING DEFINITIONS

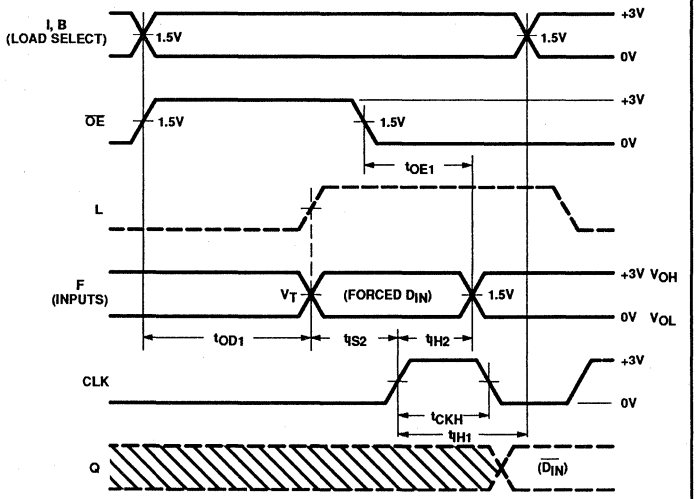
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



* Preset and Reset functions override Clock. However, F outputs may glitch with the first positive Clock Edge if t_{IS1} cannot be guaranteed by the user.

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (16 × 45 × 12)

PLS159A

LOGIC PROGRAMMING

The PLS159A is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

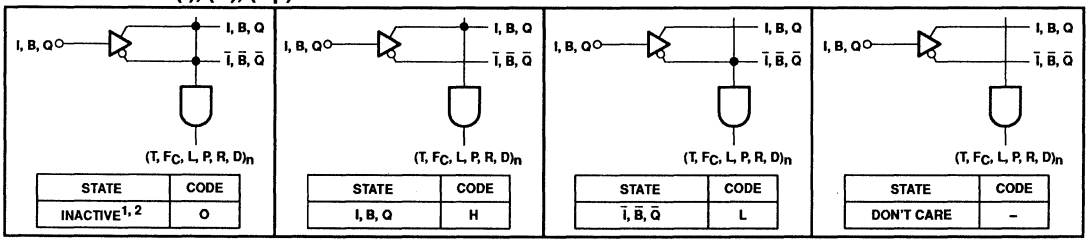
PLS159A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I), (B), (Qp)



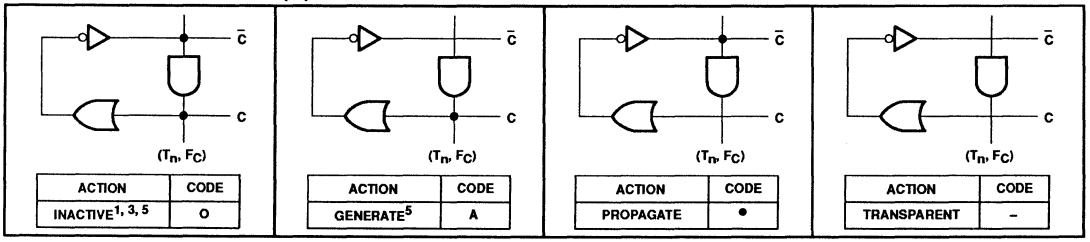
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
$\bar{I}, \bar{B}, \bar{Q}$	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



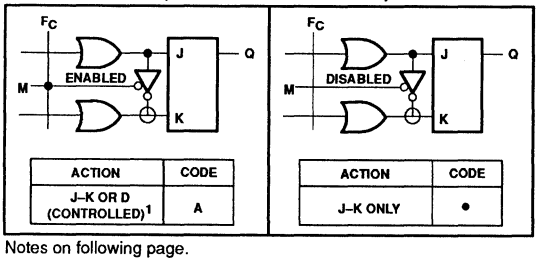
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

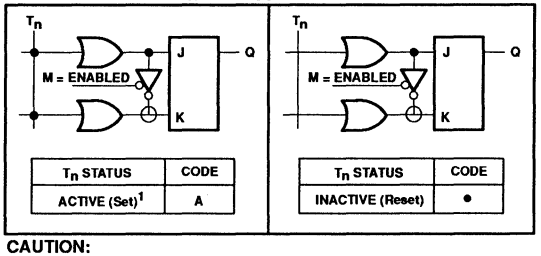
"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)[1]	A

ACTION	CODE
J–K ONLY	•

"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)[1]	A

T_n STATUS	CODE
INACTIVE (Reset)	•

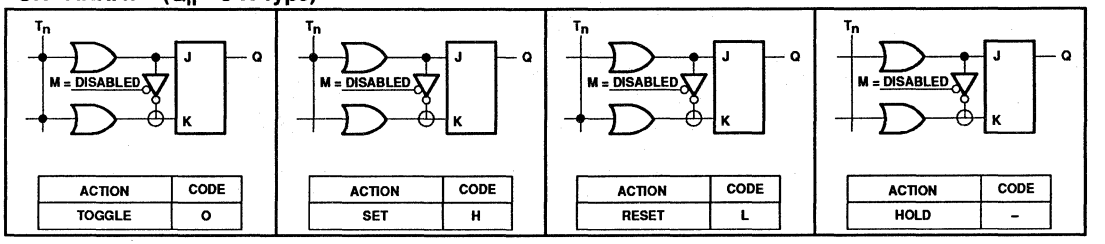
Notes on following page.

CAUTION:
THE PLS159A Programming Algorithm is different from the PLS159.

Programmable logic sequencer (16 × 45 × 12)

PLS159A

"OR" ARRAY – (Q_n = J-K Type)



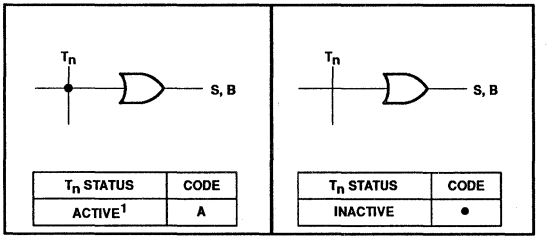
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

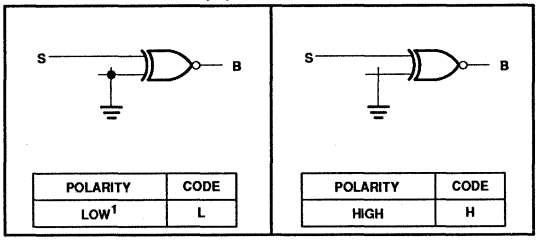
"OR" ARRAY – (S or B)



T_n STATUS	CODE
ACTIVE[1]	A

T_n STATUS	CODE
INACTIVE	•

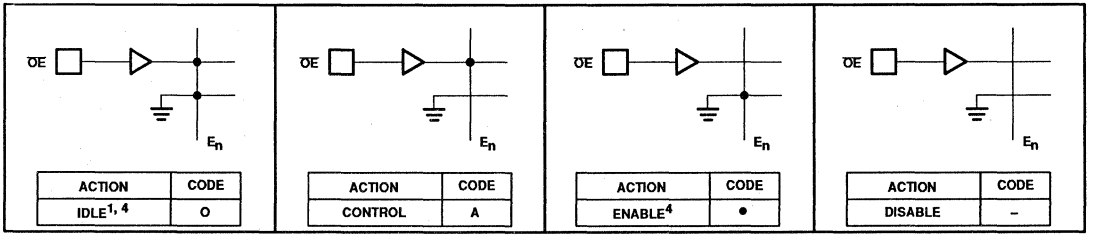
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW[1]	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate $(T, F_C, L, P, R, D)_n$ will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates $(L, P, R, D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (16 × 45 × 12)

PLS159A

PROGRAM TABLE

AND			OR			CONTROL				
INACTIVE	O	I, B(I), Q(P)	ACTIVE	A	P, R, B(O) (Q = D)	J/K	•	F/F MODE	IDLE	O
I, B, Q	H		INACTIVE	•		J/K or D (controlled)	A		CONTROL	A
I, B, Q̄	L								ENABLE	•
DON'T CARE	–								DISABLE	–
INACTIVE	O	C	TOGGLE	O	(Q = J/K)	HIGH	H	(POL)		
GENERATE	A		SET	H		LOW	L			
PROPAGATE	•		RESET	L						
TRANSPARENT	–		HOLD	–						

NOTES

1. The device is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity.
2. Program unused C, I, B, and Q bits in the AND array as (–). Program unused Q, B, P, and R bits in the OR array as (–) or (A), as applicable.
3. Unused Terms can be left blank.
4. Q (P) and Q (N) are respectively the present and next states of flip-flops Q.

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX)_____

CUSTOMER SYMBOLIZED PART #_____

DATE RECEIVED_____

COMMENTS_____

CUSTOMER NAME_____

PURCHASE ORDER #_____

PHILIPS DEVICE #_____

TOTAL NUMBER OF PARTS_____

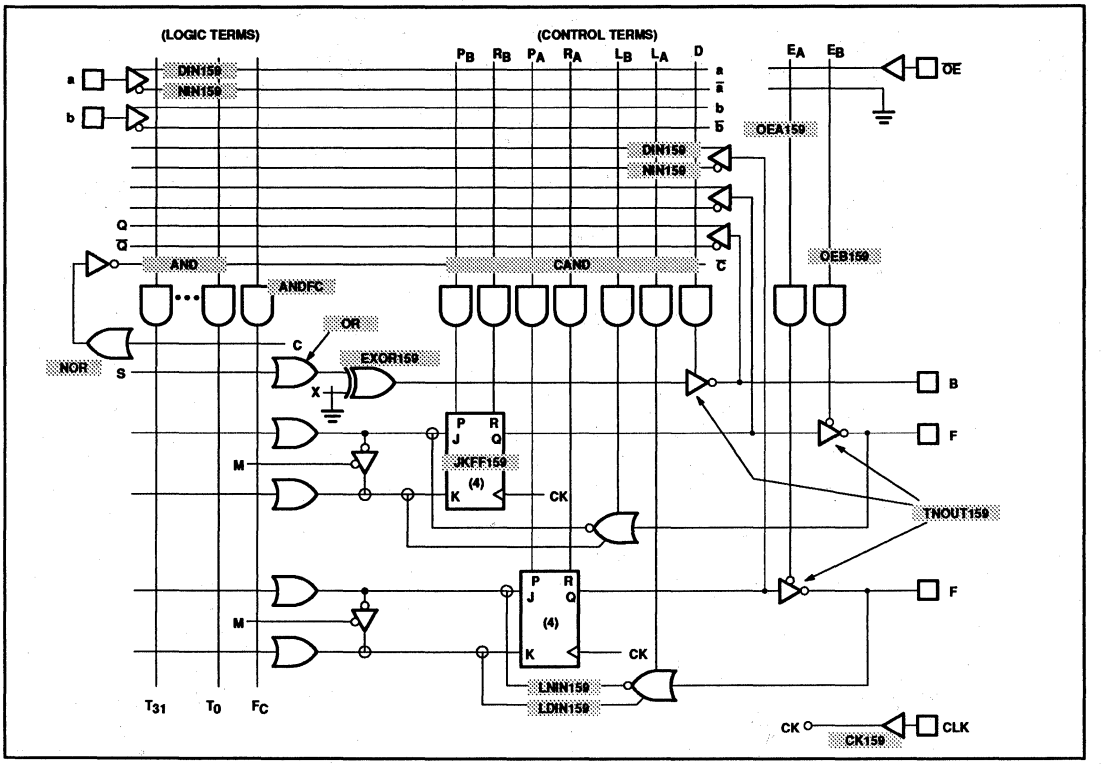
PROGRAM TABLE #_____ REV____ DATE_____

TERM	C	AND I				AND B(I)				AND Q(P)								F/F MODE	E_B	E_A	POLARITY	(OR) Q(N)								(OR) B(O)			
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0					7	6	5	4	3	2	1	0	3	2	1	0
0																																	
1																																	
2																																	
3																																	
4																																	
5																																	
6																																	
7																																	
8																																	
9																																	
10																																	
11																																	
12																																	
13																																	
14																																	
15																																	
16																																	
17																																	
18																																	
19																																	
20																																	
21																																	
22																																	
23																																	
24																																	
25																																	
26																																	
27																																	
28																																	
29																																	
30																																	
31																																	
F_C																																	
P_B																																	
R_B																																	
L_B																																	
P_A																																	
R_A																																	
L_A																																	
D3																																	
D2																																	
D1																																	
D0																																	
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12																

Programmable logic sequencer (16 × 45 × 12)

PLS159A

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencers (14 × 48 × 6)

PLS167/A

DESCRIPTION

The PLS167 and PLS167A are bipolar, Programmable Logic State machines of the Mealy type. The Programmable Logic Sequencers (PLS) contain logic AND/OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 8 Q_P, and 4 Q_F edge-triggered, clocked S/R flip-flops, with an asynchronous Preset Option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 14 external inputs, I0-13, with 8 internal inputs, P0-7, fed back from the State Register to form up to 48 transition terms (AND terms). In addition, P0 and P1 of the internal State Register are brought off-chip to allow extending the Output Register to 6 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-enable function, as an additional user programmable option.

Order codes are listed in the Ordering Information Table.

FEATURES

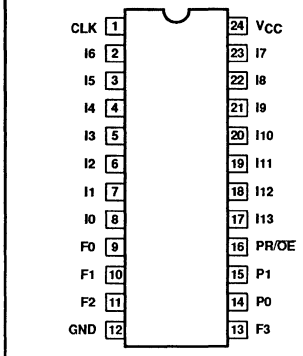
- PLS167
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS167A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 14 True/Complement buffered inputs
- 48 programmable AND gates
- 25 programmable OR gates
- 8-bit State Register
- 2-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- Power: 600mW (typ.)
- TTL compatible
- 3-State outputs
- Single +5V supply

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems

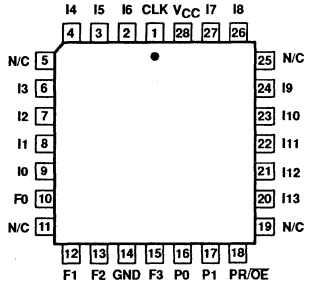
PIN CONFIGURATIONS

N Package



N = Plastic Dual In-Line Package (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

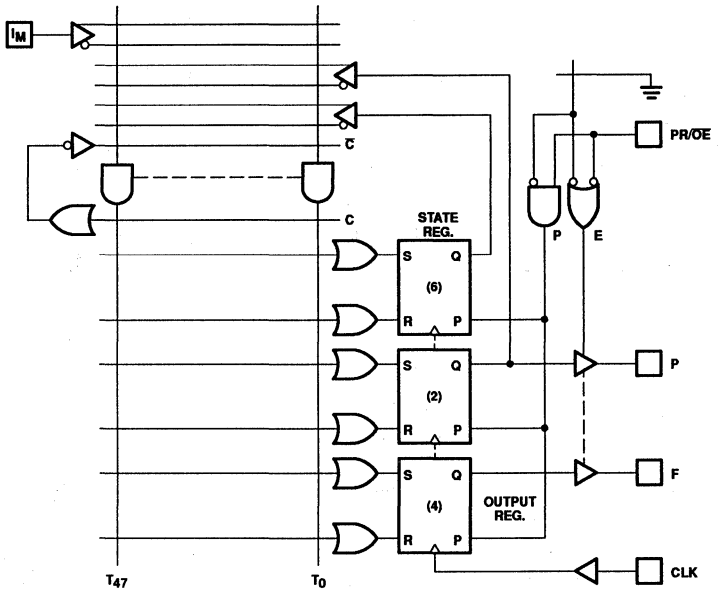
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Plastic Dual In-Line Package (300mil-wide)	PLS167N, PLS167AN	0410D
28-Pin Plastic Leaded Chip Carrier	PLS167A, PLS167AA	0401F

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

FUNCTIONAL DIAGRAM



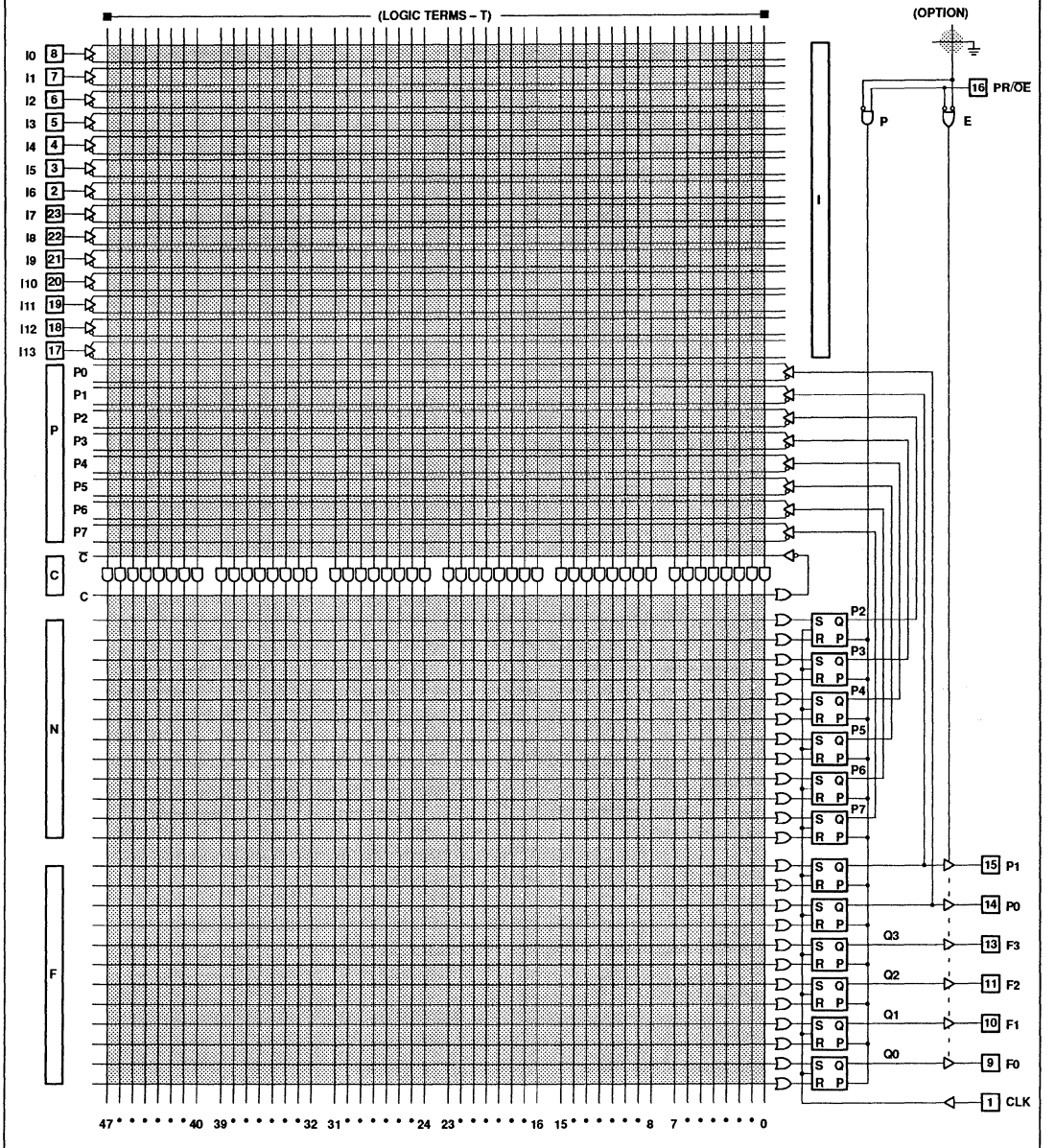
PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 7 17 – 23	I1 – I13	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
8	I0	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercised with standard TTL levels. When I0 is held at +10V, device outputs F0 – 3 and P0 – 1 reflect the contents of State Register bits P2 – 7 (see Diagnostic Output Mode diagram). The contents of flip-flops P0 – 1 and F0 – 3 remain unaltered.	Active-High/Low
9 – 11 13	F0 – 3	**Logic/Diagnostic Outputs:** Four device outputs which normally reflect the contents of Output Register bits Q_{0-3}, when enabled. When I0 is held at +10V, F0 – 3 = (P2 – 5).	Active-High
14 – 15	P0 – 1	**Logic/Diagnostic Outputs:** Two register bits with shared function as least Significant State Register bits, or most significant Output Register bits. When I_0 is held at +10V, P0 – 1 = (P6 – 7).	Active-High
16	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P0 – 7 and F0 – 3 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an Output Enable function to all output buffers.	Active-Low (L)

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

LOGIC DIAGRAM



NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ▒ Programmable connection.

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CK	S	R	$Q_{P/F}$	F
	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi-Z
+5V		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots T_{47}$
   $T_n = C(I0\ I1\ I2 \ldots)\ (P0\ P1 \ldots P7)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V)

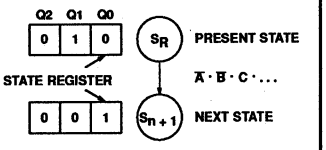
ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION

Typical State Transition:



SET Q_0: $S_0 = (\overline{Q_2} \cdot Q_1 \cdot \overline{Q_0}) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$
$R_0 = 0$

RESET Q_1: $S_1 = 0$
$R_1 = (\overline{Q_2} \cdot Q_1 \cdot \overline{Q_0}) \cdot \overline{A} \cdot \overline{B} \cdot C \ldots$

HOLD Q_2: $S_2 = 0$
$R_2 = 0$

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array pre-programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Philips Semiconductors qualified programming equipment.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
V_{OH}	High[4]	V_{CC} = MIN I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	80	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[5, 6]	V_{CC} = MAX V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5.0V V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

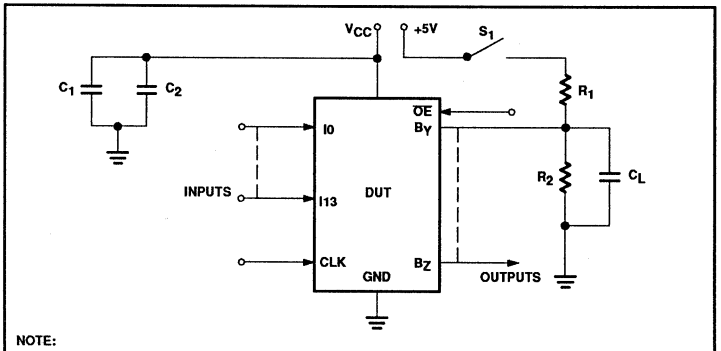
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75°CV \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS167			PLS167A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width[3]										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock Period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS_1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS_1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS_1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS_2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS_2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS_2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		5	–5		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

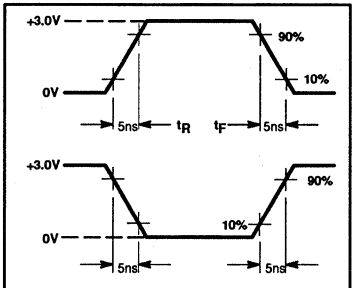
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



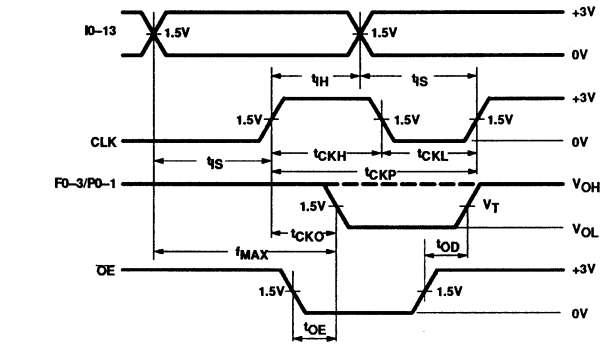
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

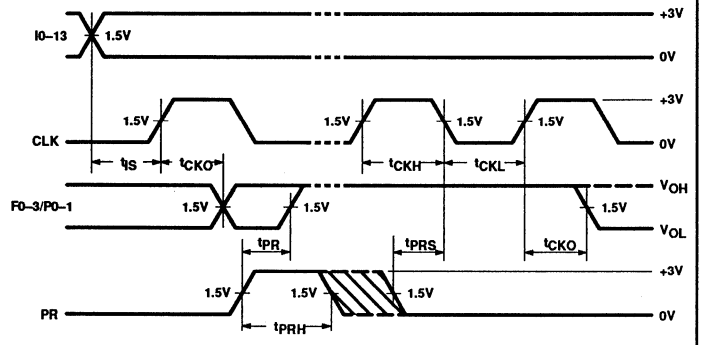
Programmable logic sequencers (14 × 48 × 6)

PLS167/A

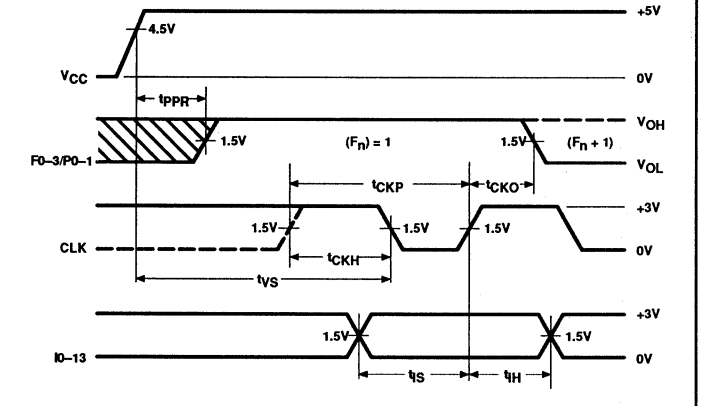
TIMING DIAGRAMS



Flip-Flop Outputs



Asynchronous Preset



Power-On Preset

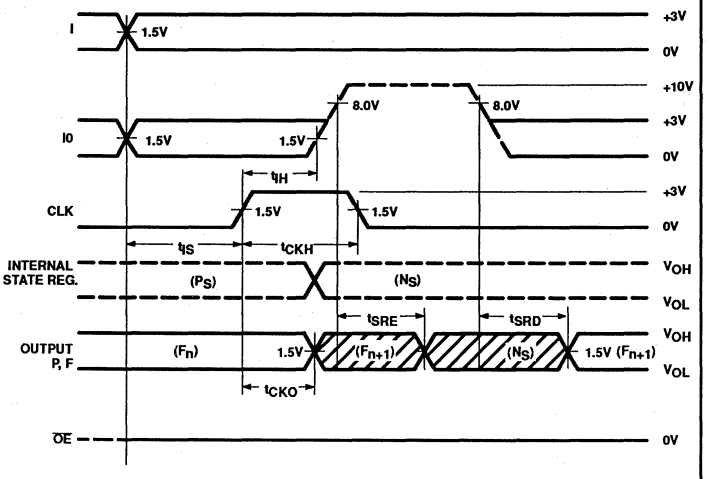
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/$\overline{OE}$ Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

TIMING DIAGRAMS (Continued)



Diagnostic Output Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS167 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The PLS167A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

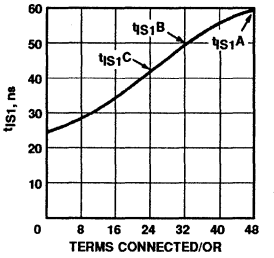


Figure 1. PLS167 t_{IS1} vs. Terms/OR Connected

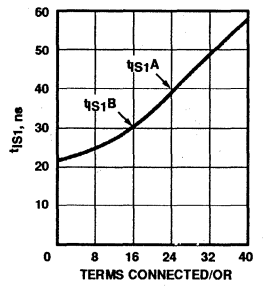


Figure 2. PLS167A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or Figure 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

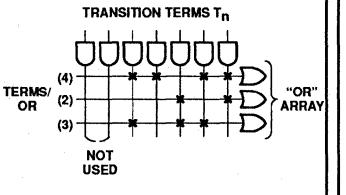


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

LOGIC PROGRAMMING

The PLS167/A devices are fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™ CUPL™ and PALASM® 90 design software packages also support the PLS167/A architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

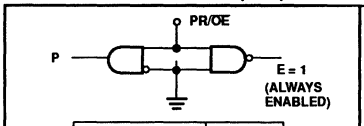
PLS167/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

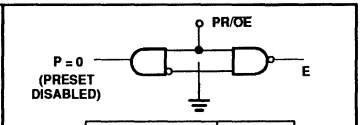
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

PRESET/$\overline{OE}$ OPTION – (P/E)



OPTION	CODE
PRESET[1]	H

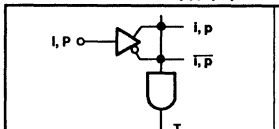


OPTION	CODE
$\overline{OE}$	L

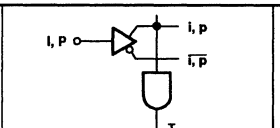
PROGRAMMING:

The PS167/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

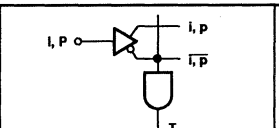
"AND" ARRAY – (I), (P)



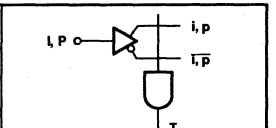
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, P	H

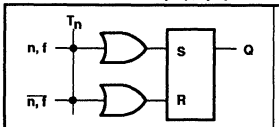


STATE	CODE
$\overline{I}$, $\overline{P}$	L

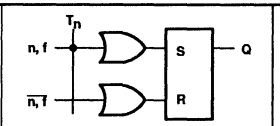


STATE	CODE
DON'T CARE	–

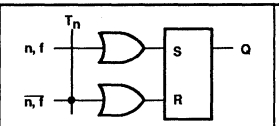
"OR" ARRAY – (N), (F)



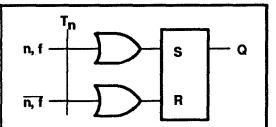
ACTION	CODE
INACTIVE[1, 3]	O



ACTION	CODE
SET	H

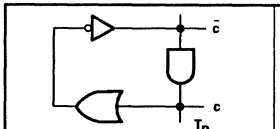


ACTION	CODE
RESET	L

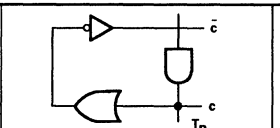


ACTION	CODE
NO CHANGE	–

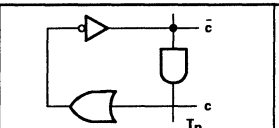
"COMPLEMENT" ARRAY – (C)



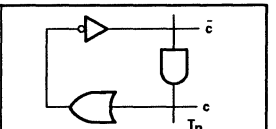
ACTION	CODE
INACTIVE[1, 4]	O



ACTION	CODE
GENERATE	A



ACTION	CODE
PROPAGATE	•



ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencers (14 × 48 × 6)

PLS167/A

PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
PHILIPS DEVICE #_____ CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

INACTIVE	0	C_n
GENERATE	A	
PROPAGATE	•	
TRANSPARENT	—	

INACTIVE	0	I_m, P_s
I, P	H	
$\overline{I}$, $\overline{P}$	L	
DON'T CARE	—	

OR

INACTIVE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	—	

OPTION

PRESET	H	P/E
$\overline{OE}$	L	

TERM	C_n	AND: INPUT (I_m) 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 7 6 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (Ns) 7 6 5 4 3 2 1 0	OUTPUT (Fr) 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		17 18 19 20 21 22 23 2 3 4 5 6 7 8			15 14 13 11 10 9	
VARIABLE NAME						

NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencers (14 × 48 × 6)

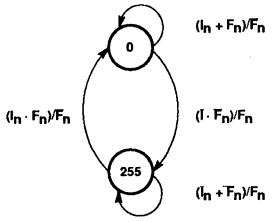
PLS167/A

TEST ARRAY

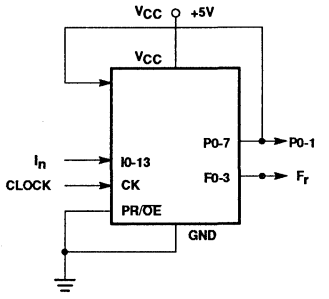
The PLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the PLS and applying the proper input sequence to $I_{0\text{-}13}$ as shown in the test circuit timing diagram.



State Diagram

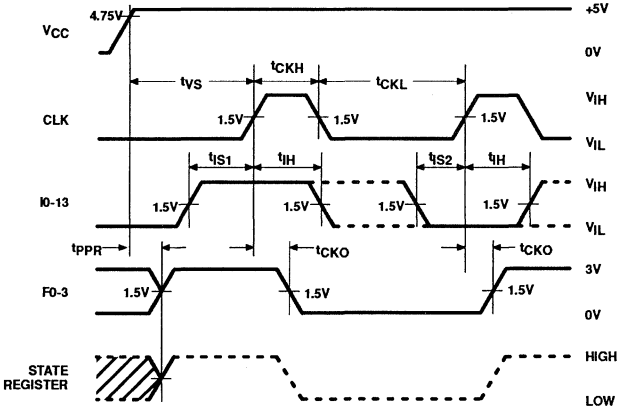


PLS Under Test

TERM	C	AND: INPUT (Im) 13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 7	6	5	4	3	2	1	0	OPTION (P/E) H — OR: NEXT STATE (Ns) 7	6	5	4	3	2	1	0	OUTPUT (Fr) 3	2	1	0
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Signetic's qualified programming equipment.

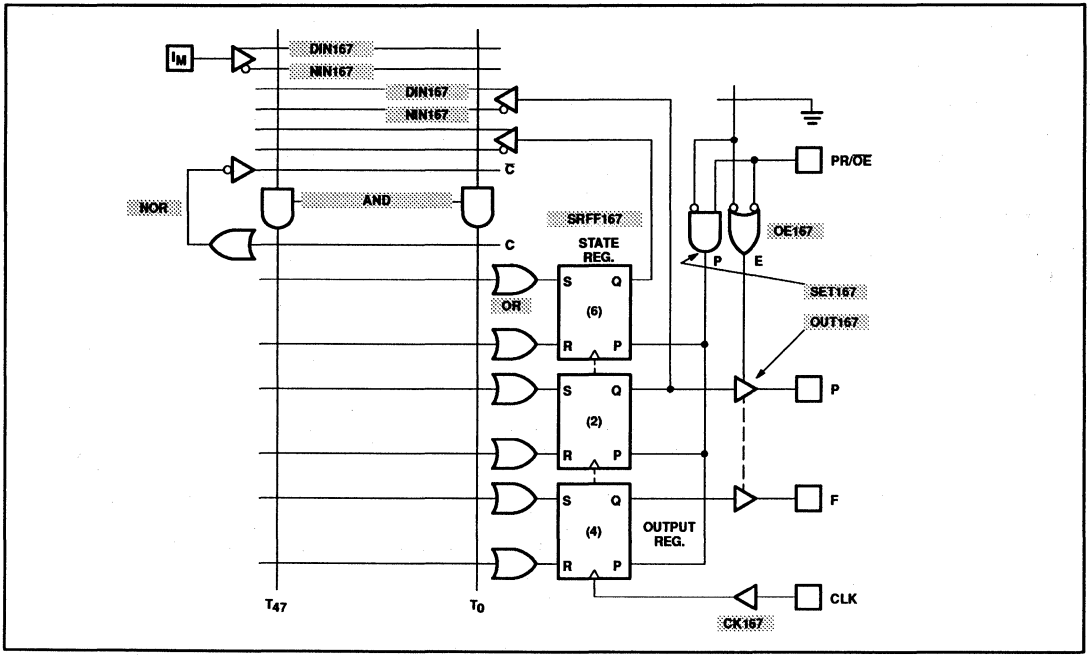


Test Circuit Timing Diagram

TERM	C	AND: INPUT (Im) 13	12	11	10	9	8	7	6	5	4	3	2	1	0	PRESENT STATE (Ps) 7	6	5	4	3	2	1	0	OPTION (P/E) H — OR: NEXT STATE (Ns) 7	6	5	4	3	2	1	0	OUTPUT (Fr) 5	4	3	2	1	0
48	–	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	–	–	–	–	–	–	–	–	–	–	–	–
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	–	–	–	–	–	–	–	–	–	–	–	–

Test Array Deleted

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencers (12 × 48 × 8)

PLS168/A

DESCRIPTION

The PLS168 and the PLS168A are bipolar, Programmable Logic State machines of the Mealy type. They contain logic AND/OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 10 Q_P, and 4 Q_F edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset Option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 12 external inputs, I0-11, with 10 internal inputs, P0-9, fed back from the State Register to form up to 48 transition terms (AND terms). In addition, P0–P3 of the internal State Register are brought off-chip to allow extending the Output Register to 8 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

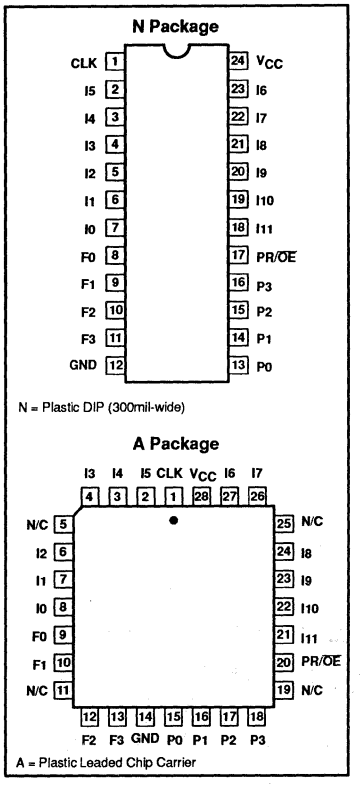
Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-enable function, as an additional user programmable option.

Order codes are listed in the Ordering Information table below.

FEATURES

- PLS168
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS168A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 12 True/Complement buffered inputs
- 48 programmable AND gates
- 29 programmable OR gates
- 10-bit State Register
- 4-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- Power: 600mW (typ.)
- TTL compatible
- 3-State outputs
- Single +5V supply

PIN CONFIGURATIONS



APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems
- Counters
- Shift registers

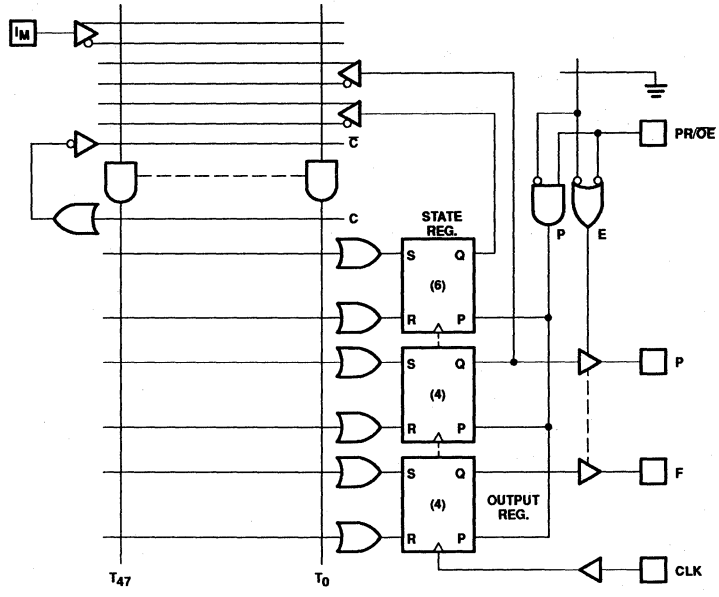
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Plastic DIP (300mil-wide)	PLS168N, PLS168AN	0410D
28-Pin Plastic Leaded Chip Carrier	PLS168A, PLS168AA	0401F

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

FUNCTIONAL DIAGRAM



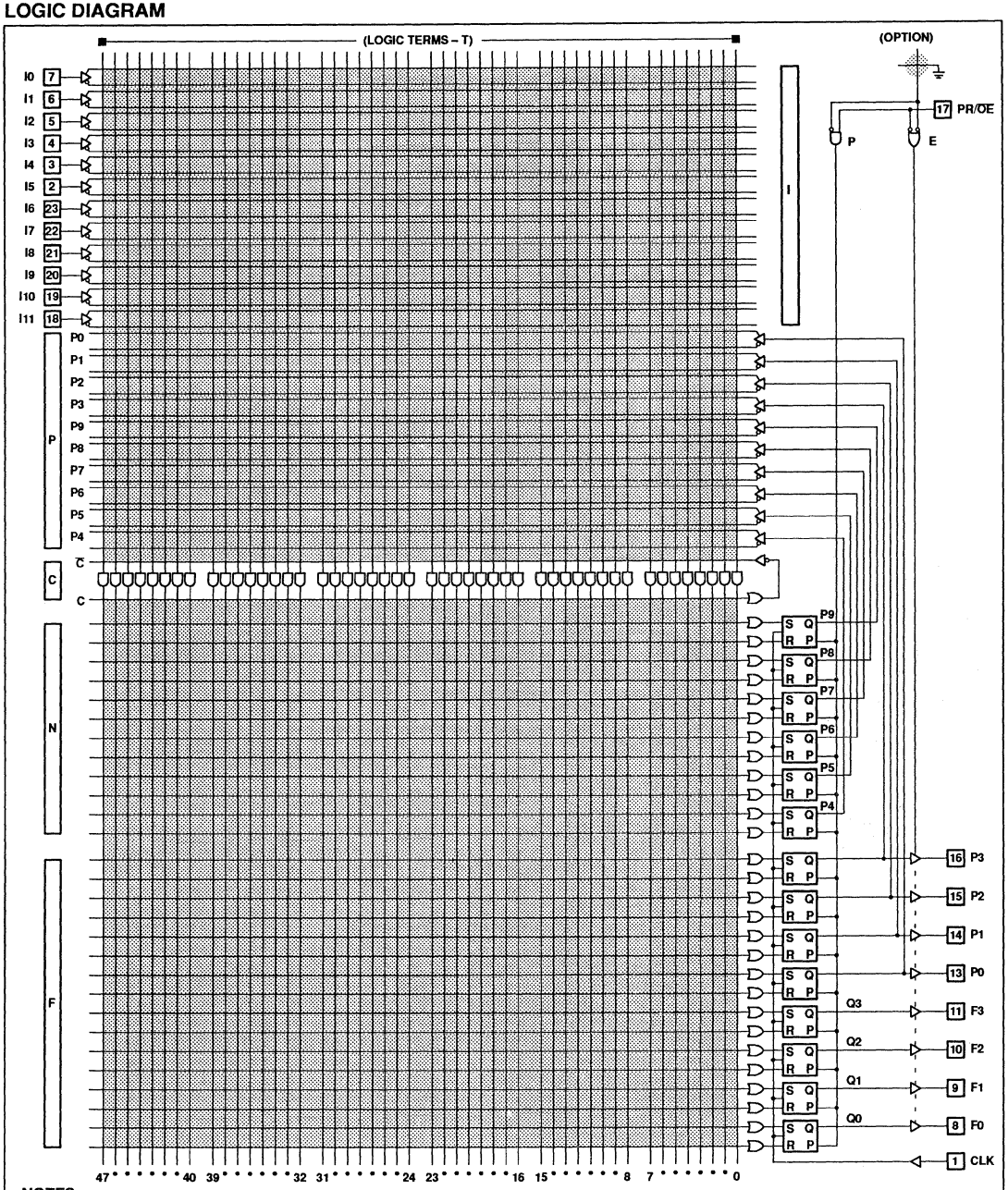
PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 6 18 – 23	I1 – I11	**Logic Inputs:** The 11 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
7	I0	**Logic/Diagnostic Input:** A 12th external logic input to the AND array, as above, when exercised with standard TTL levels. When I0 is held at +10V, device outputs F2 – F3 and P0 – P3 reflect the contents of State Register bits P4 – 9 (see Diagnostic Output Mode diagram). The contents of flip-flops P0 – 1 and F0 – 3 remain unaltered.	Active-High/Low
13 – 16	P0 – 3	**Logic/Diagnostic Outputs:** Four device outputs which normally reflect the contents of State Register bits P0 – 3. When I0 is held at +10V these pins reflect (P6 – P9).	Active-High
10 – 11	F2 – F3	**Logic/Diagnostic Outputs:** Two register bits (F2 – F3) which reflect Output register bits (Q2 – Q3). When I0 is held at +10V, these pins reflect (P4 – P5).	Active-High
17	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P0 – 9 and F0 – 3 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an Output Enable function to all output buffers.	Active-Low (L)
8, 9	F0 – F1	**Logic Output:** Two device outputs which reflect Output Registers Q0 – Q1. When I0 is held at +10V, F0 – F1 = Logic "1".	

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

LOGIC DIAGRAM



NOTES:

1. All programmed "AND" gate locations are pulled to logic "1".
2. All programmed "OR" gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CLK	S	R	$Q_{P/F}$	F
	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi-Z
+5V		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

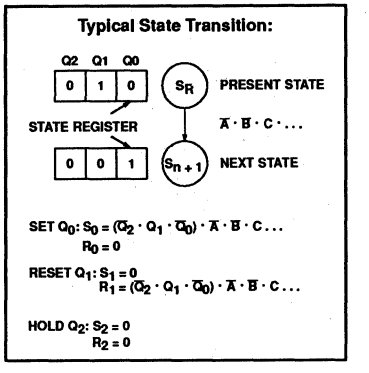
NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots + T_{47}$
   $T_n = C(I0\ I1\ I2 \ldots)\ (P0\ P1 \ldots P9)$
2. Either Preset (Active-High) or $\overline{\text{Output Enable}}$ (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V)

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:
1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array pre-programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Philips Semiconductors qualified programming equipment.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High[4]	I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CLK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state[6]	V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

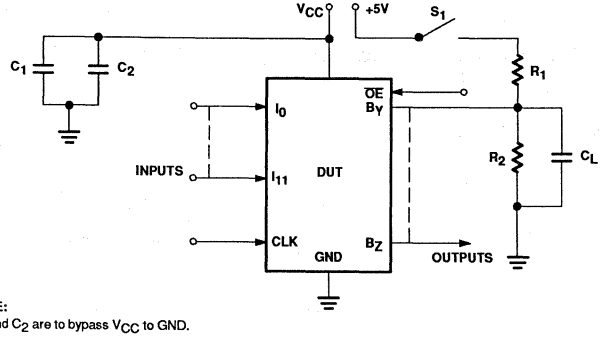
AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75°CV \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS168			PLS168A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width[3]										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock Period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		5	–10		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

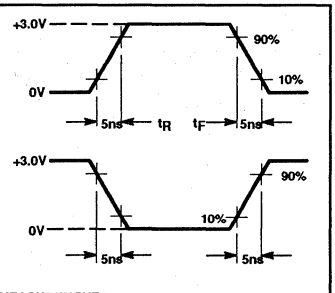
NOTES:
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



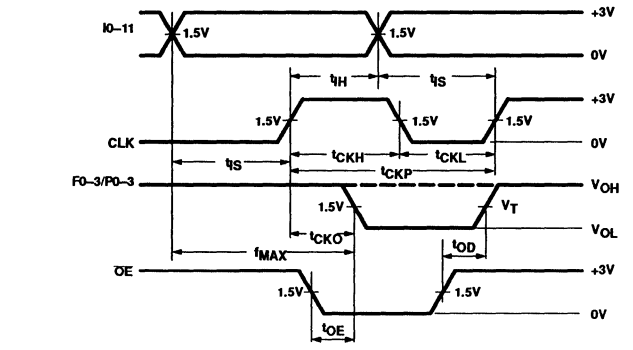
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

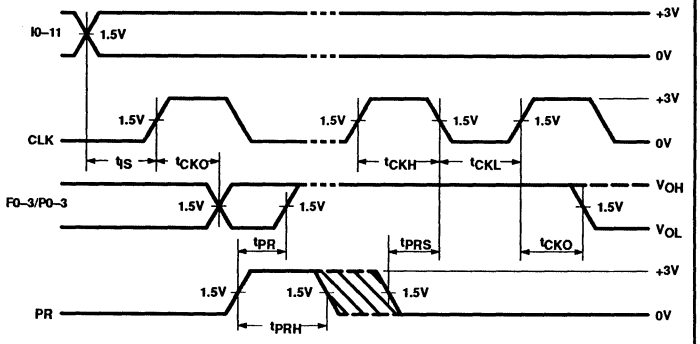
Programmable logic sequencers (12 × 48 × 8)

PLS168/A

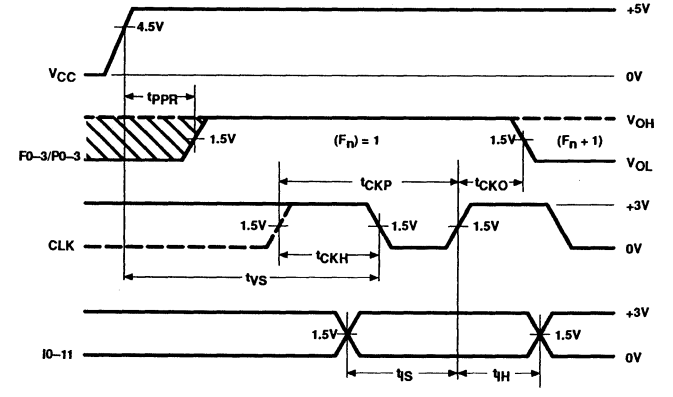
TIMING DIAGRAMS



Flip-Flop Outputs



Asynchronous Preset



Power-On Preset

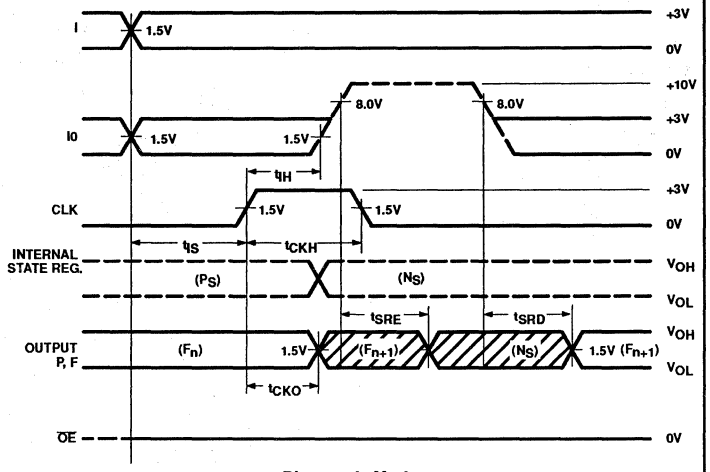
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/$\overline{OE}$ Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

TIMING DIAGRAMS (Continued)



Diagnostic Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS168 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The PLS168A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

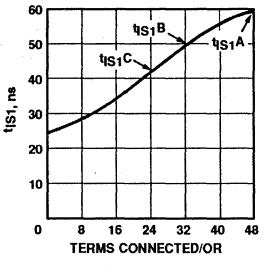


Figure 1. PLS168 t_{IS1} vs. Terms/OR Connected

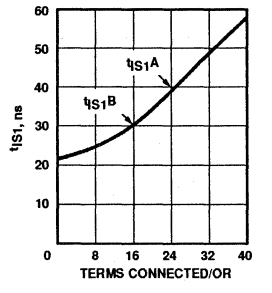


Figure 2. PLS168A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

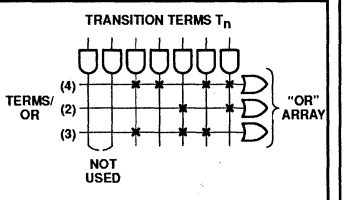


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

LOGIC PROGRAMMING

The PLS168/A devices are fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLS168/A architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

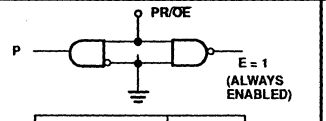
PLS168/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

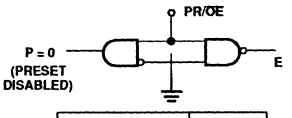
PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-partyl Programmer/ Software Support)* of this data handbook for additional information.

PRESET/$\overline{OE}$ OPTION – (P/E)



OPTION	CODE
PRESET[1]	H

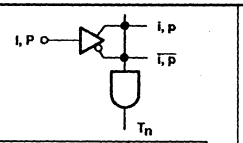


OPTION	CODE
$\overline{OE}$	L

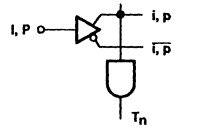
PROGRAMMING:

The PLS168/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

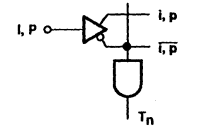
"AND" ARRAY – (I), (P)



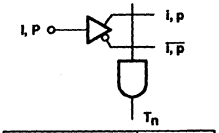
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, P	H

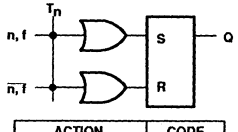


STATE	CODE
$\overline{I, P}$	L

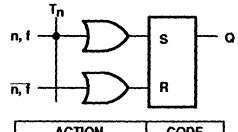


STATE	CODE
DON'T CARE	–

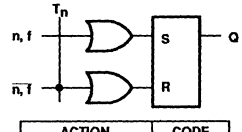
"OR" ARRAY – (N), (F)



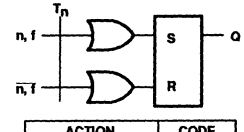
ACTION	CODE
INACTIVE[1, 3]	O



ACTION	CODE
SET	H

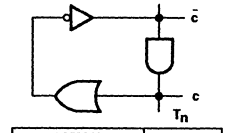


ACTION	CODE
RESET	L

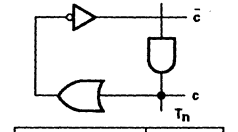


ACTION	CODE
NO CHANGE	–

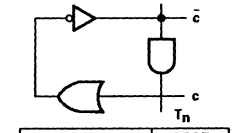
"COMPLEMENT" ARRAY – (C)



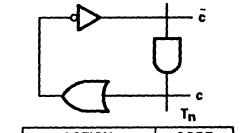
ACTION	CODE
INACTIVE[1, 4]	O



ACTION	CODE
GENERATE	A



ACTION	CODE
PROPAGATE	•



ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

PROGRAM TABLE

CUSTOMER NAME____________________

PURCHASE ORDER #____________________

PHILIPS DEVICE #__________ CF (XXXX)__________

CUSTOMER SYMBOLIZED PART #____________________

TOTAL NUMBER OF PARTS____________________

PROGRAM TABLE____________________

REV____________________

DATE____________________

PROGRAM TABLE ENTRIES

AND

INACTIVE	0	C_n
GENERATE	A	
PROPAGATE	•	
TRANSPARENT	—	

INACTIVE	0	I_m, P_s
I, P	H	
$\overline{I}$, $\overline{P}$	L	
DON'T CARE	—	

OR

INACTIVE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	—	

OPTION

PRESET	H	P/E
$\overline{OE}$	L	

AND | OPTION (P/E) | OR

TERM	C_n	INPUT (I_m)												PRESENT STATE (P_s)										REMARKS	NEXT STATE (N_s)										OUTPUT (F_r)			
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		9	8	7	6	5	4	3	2	1	0	3	2	1	0
0																																						
1																																						
2																																						
3																																						
4																																						
5																																						
6																																						
7																																						
8																																						
9																																						
10																																						
11																																						
12																																						
13																																						
14																																						
15																																						
16																																						
17																																						
18																																						
19																																						
20																																						
21																																						
22																																						
23																																						
24																																						
25																																						
26																																						
27																																						
28																																						
29																																						
30																																						
31																																						
32																																						
33																																						
34																																						
35																																						
36																																						
37																																						
38																																						
39																																						
40																																						
41																																						
42																																						
43																																						
44																																						
45																																						
46																																						
47																																						
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																		16	15	14	13	11	10	9	8
VARIABLE NAME																																						

NOTES:

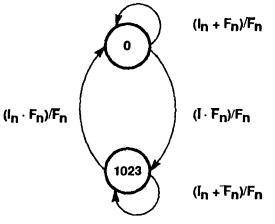
1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

TEST ARRAY

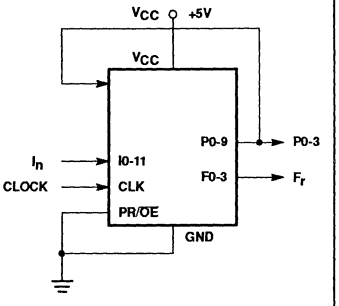
The PLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to I0-13 as shown in the test circuit timing diagram.



State Diagram

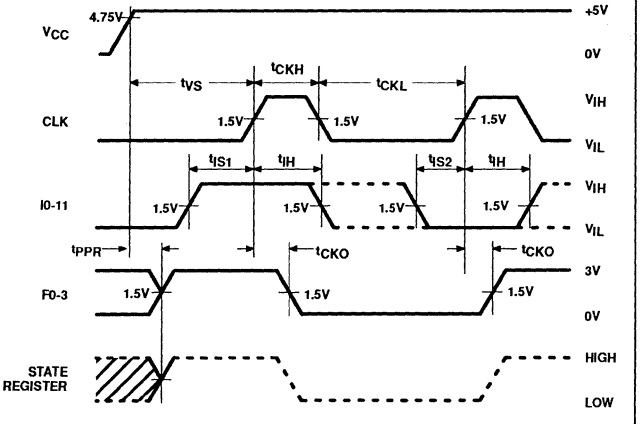


FPLS Under Test

TERM	C_n	AND: INPUT (Im) 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (Ps) 9 8 7 6 5 4 3 2 1 0	OPTION (P/E) H — OR: NEXT STATE (Ns) 9 8 7 6 5 4 3 2 1 0	OR: OUTPUT (Fr) 3 2 1 0
48	A	H H H H H H H H H H H H	H H H H H H H H H H	L L L L L L L L L L	L L L L
49	•	L L L L L L L L L L L L	L L L L L L L L L L	H H H H H H H H H H	H H H H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Signetic's qualified programming equipment.



Test Circuit Timing Diagram

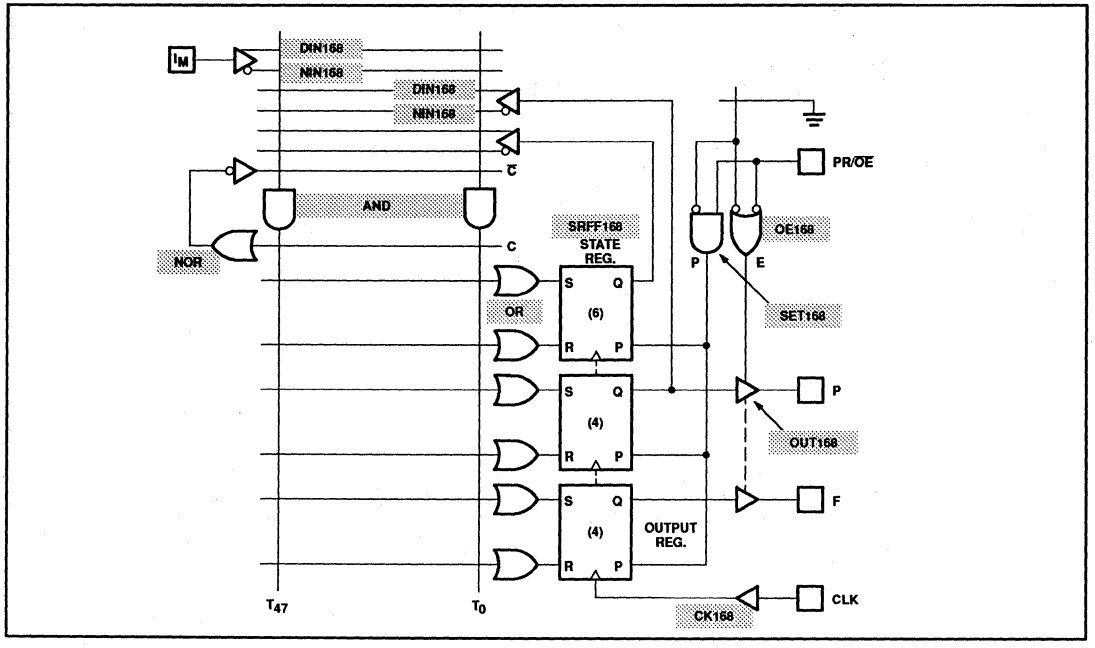
TERM	C_n	AND: INPUT (Im) 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (Ps) 9 8 7 6 5 4 3 2 1 0	OPTION (P/E) H — OR: NEXT STATE (Ns) 9 8 7 6 5 4 3 2 1 0	OR: OUTPUT (Fr) 3 2 1 0
48	–	H H H H H H H H H H H H	H H H H H H H H H H	– – – – – – – – – –	– – – –
49	•	L L L L L L L L L L L L	L L L L L L L L L L	– – – – – – – – – –	– – – –

Test Array Deleted

Programmable logic sequencers (12 × 48 × 8)

PLS168/A

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (20 × 45 × 12)

PLS179

DESCRIPTION

The PLS179 is a 3-State output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "foldback" inverting buffer and control gate, F_C. It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). There are 8 dedicated inputs. These yield variable I/O gate and register configurations via control gates (D, L) ranging from 20 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 8 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and the Complement Array output ($\overline{C}$). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

On-chip T/C buffers couple either True (I, B, Q) or Complement ($\overline{I}$, $\overline{B}$, $\overline{Q}$, $\overline{C}$) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Any of the 32 AND gates can drive bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Similarly, any of the 32 AND gates can drive the J-K inputs of all flip-flops. Four AND gates have been dedicated for the Asynchronous Preset/Reset functions.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

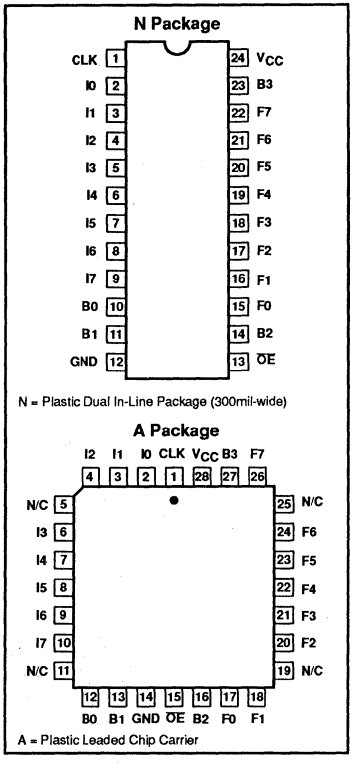
The PLS179 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are listed below.

FEATURES

- f_{MAX} = 18.2MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 8 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J/K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable $\overline{OE}$ control
- Positive edge-triggered clock
- Power-on reset on flip-flop (F_n = "1")
- Input loading: – 100µA (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- 3-State outputs

PIN CONFIGURATIONS



APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

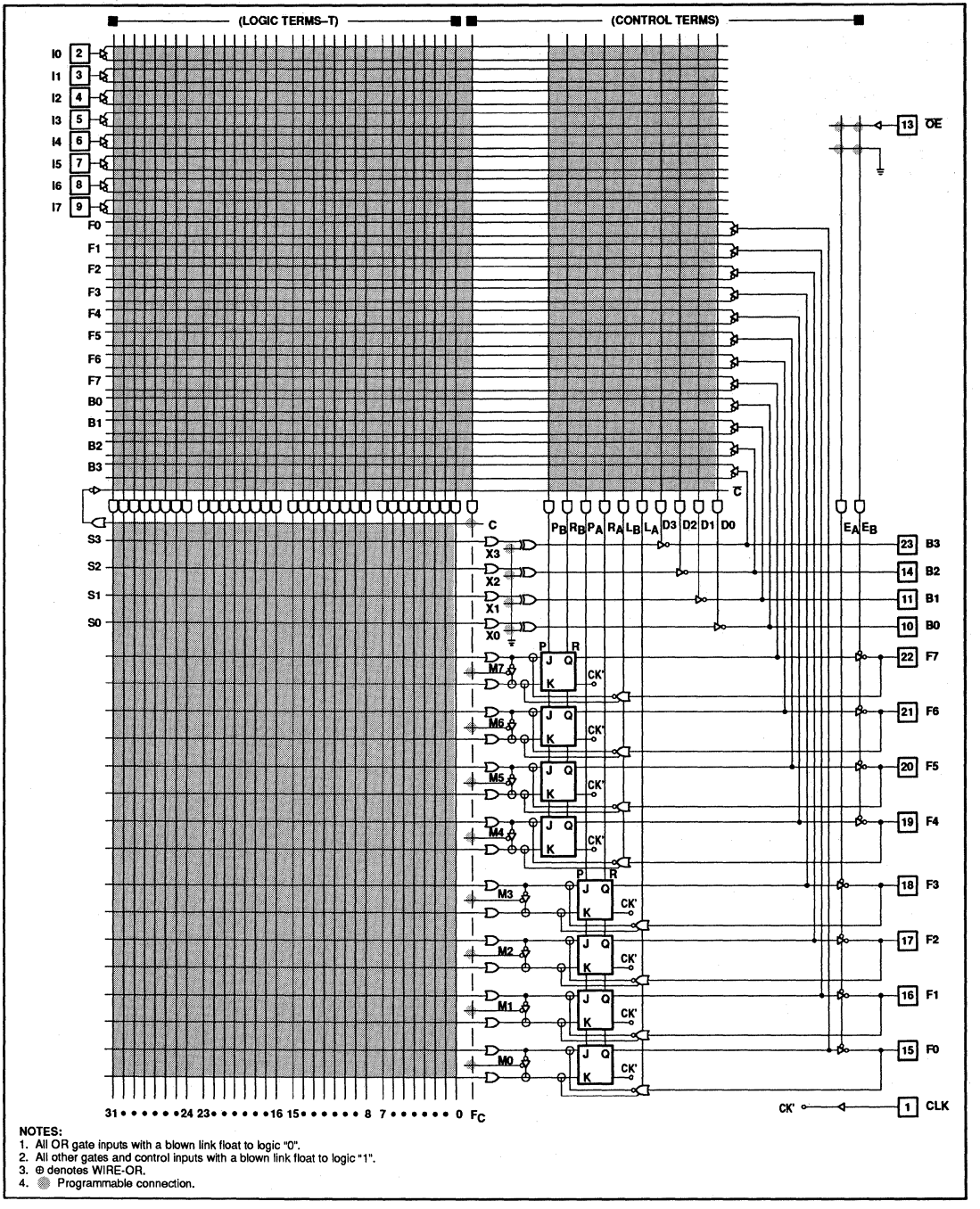
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Plastic Dual In-Line Package (300mil-wide)	PLS179N	0410D
28-Pin Plastic Leaded Chip Carrier	PLS179A	0401F

Programmable logic sequencer (20 × 45 × 12)

PLS179

LOGIC DIAGRAM

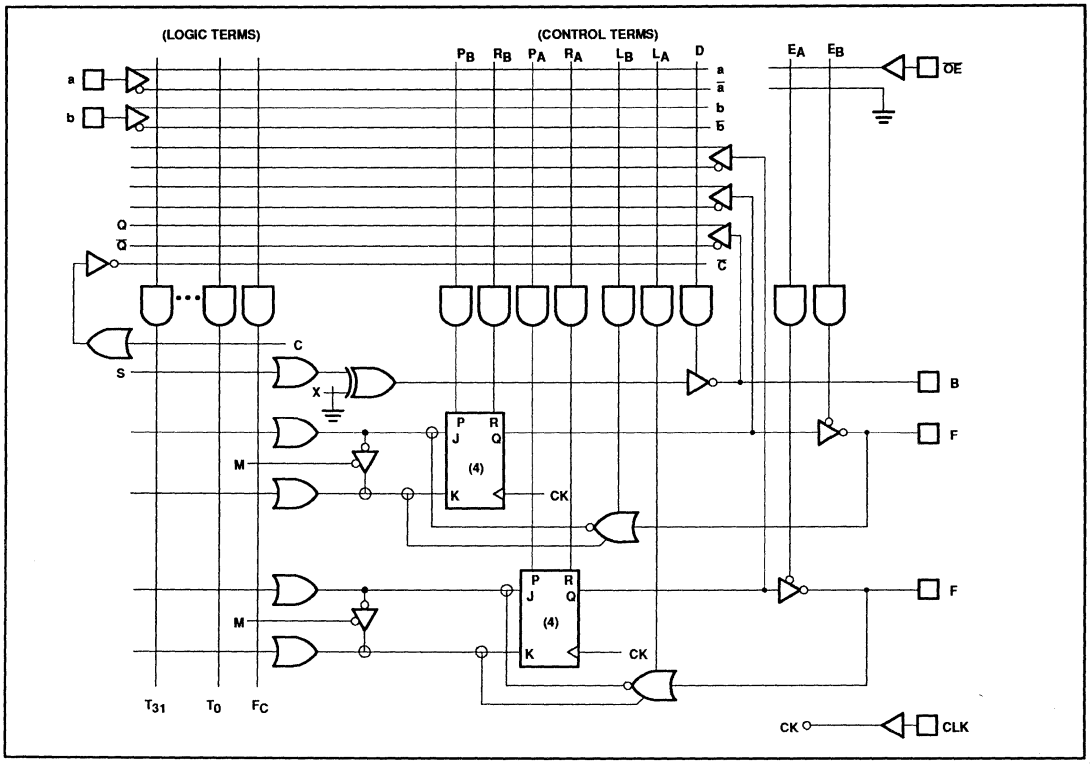


NOTES:
1. All OR gate inputs with a blown link float to logic "0".
2. All other gates and control inputs with a blown link float to logic "1".
3. ⊕ denotes WIRE-OR.
4. ▒ Programmable connection.

Programmable logic sequencer (20 × 45 × 12)

PLS179

FUNCTIONAL DIAGRAM



FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L	CK	P	R	J	K	Q	F
H								H/Hi-Z
L	X	X	X	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

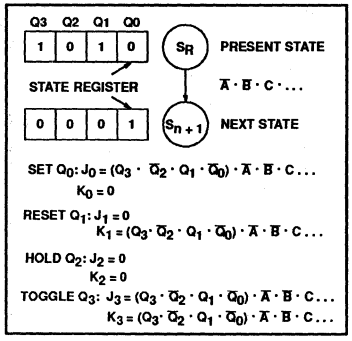
NOTES:

1. Positive Logic:
   $J\text{-}K = T_0 + T_1 + T_2 \ldots\ldots\ldots\ldots T_{31}$
   $T_n = \overline{C} \cdot (I0 \cdot I1 \cdot I2 \ldots) \cdot (Q_0 \cdot Q_1 \ldots) \cdot (B0 \cdot B1 \cdot \ldots)$
2. ↑ denotes transition from Low to High level.
3. X = Don't care
4. * = Forced at F_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J-K flip-flop independent of program code (Diagnostic mode), 3-State B outputs.

Programmable logic sequencer (20 × 45 × 12)

PLS179

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D and T mode flip-flops.

VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. $\overline{OE}$ is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J-K only or J-K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = Mln			0.8	V
V_{IC}	Clamp	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 10mA		0.35	0.5	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = 5.5V		<1	40	μA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	μA
Output current						
$I_{O(OFF)}$	Hi-Z state[4, 7]	V_{CC} = MAX, V_{OUT} = 5.5V		1	80	μA
		V_{OUT} = 0.45V			−140	μA
I_{OS}	Short circuit[3, 5]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[6]	V_{CC} = MAX		150	210	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		15		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IH} applied to $\overline{OE}$.
5. Duration of short circuit should not exceed 1 second.
6. I_{CC} is measured with the $\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.
7. Leakage values are a combination of input and output leakage.

Programmable logic sequencer (20 × 45 × 12) — PLS179

AC ELECTRICAL CHARACTERISTICS

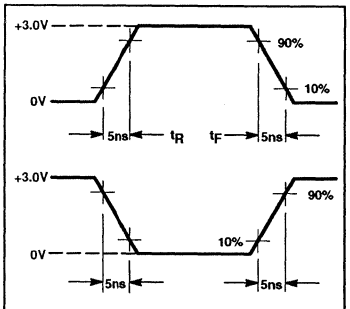
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75°CV \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN[5]	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse width[3]								
t_{CKH}	Clock[2] High	CK +	CK –	C_L = 30pF	20	15		ns
t_{CKL}	Clock Low	CK –	CK +	C_L = 30pF	20	15		ns
t_{CKP}	Clock period	CK +	CK +	C_L = 30pF	40	30		ns
t_{PRH}	Preset/Reset pulse	(I, B) –	(I, B) +	C_L = 30pF	35	30		ns
Setup time								
t_{IS1}	Input	(I, B) ±	CK +	C_L = 30pF	35	30		ns
t_{IS2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
t_{IS3}	Input (through Complement Array)[4]	(I, B) ±	CK +	C_L = 30pF	55	45		ns
Hold time								
t_{IH1}	Input	(I, B) ±	CK +	C_L = 30pF	0	–5		ns
t_{IH2}	Input (through F_n)	F ±	CK +	C_L = 30pF	15	10		ns
Propagation delay								
t_{CKO}	Clock	CK ±	F ±	C_L = 30pF		15	20	ns
t_{OE1}	Output enable[3]	$\overline{OE}$ –	F –	C_L = 30pF		20	30	ns
t_{OD1}	Output disable[3]	$\overline{OE}$ +	F +	C_L = 5pF		20	30	ns
t_{PD}	Output	(I, B) ±	B ±	C_L = 30pF		25	35	ns
t_{OE2}	Output enable[3]	(I, B) +	B ±	C_L = 30pF		20	30	ns
t_{OD2}	Output disable[3]	(I, B) –	B +	C_L = 5pF		20	30	ns
t_{PRO}	Preset/Reset	(I, B) +	F ±	C_L = 30pF		35	45	ns
t_{PPR}	Power-on preset	V_{CC} +	F –	C_L = 30pF		0	10	ns

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 10ns.
3. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
4. When using the Complement Array t_{CKP} = 75ns (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

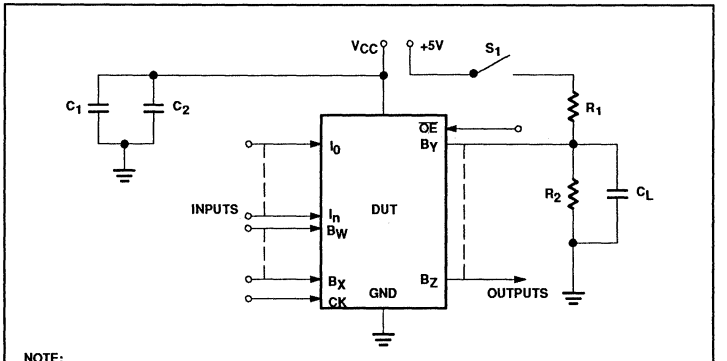
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

TEST LOAD CIRCUIT

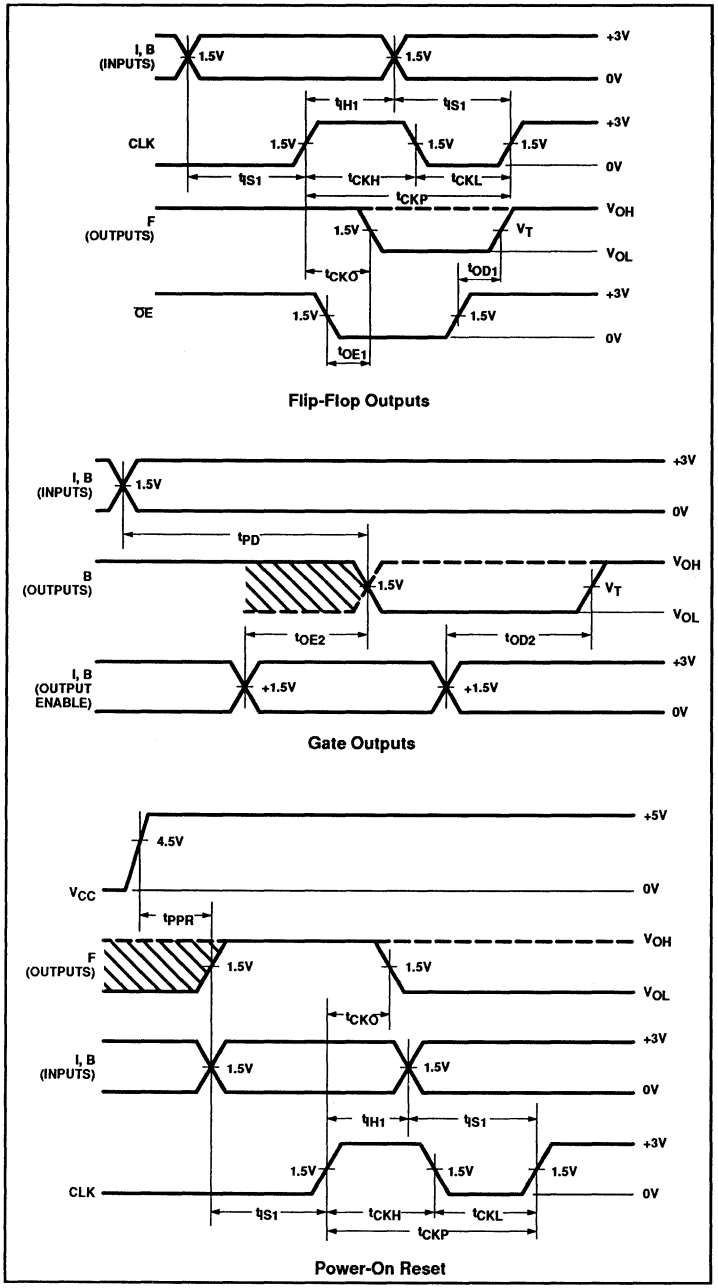


NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Programmable logic sequencer (20 × 45 × 12)

PLS179

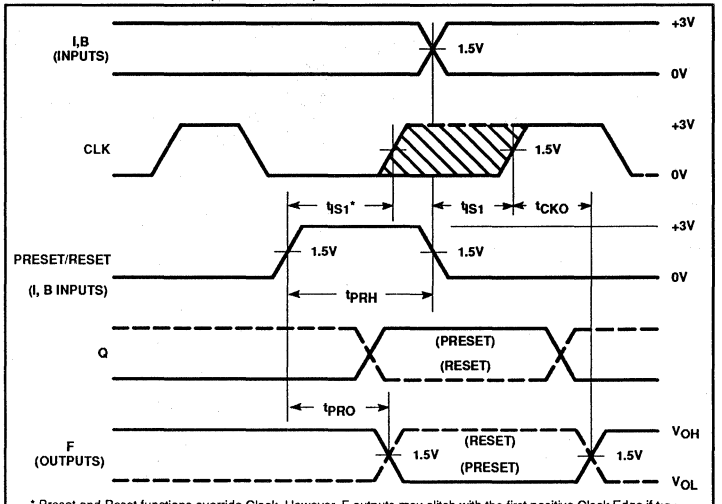
TIMING DIAGRAMS



TIMING DEFINITIONS

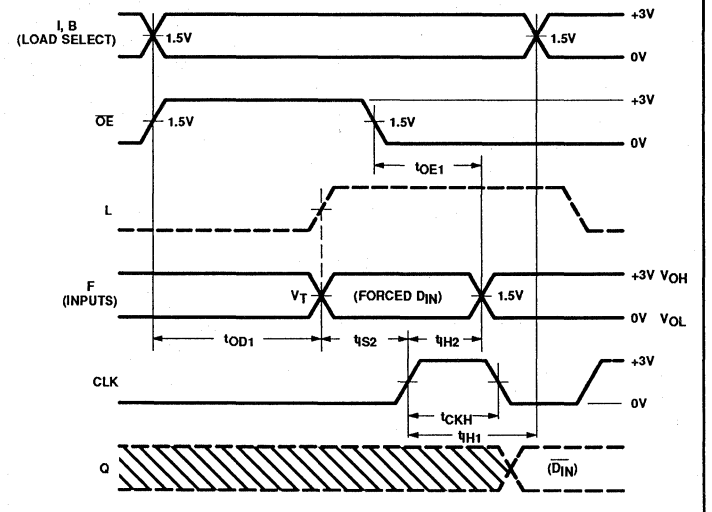
SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed Clock period.
t_{PRH}	Width of preset input pulse.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
t_{IH1}	Required delay between positive transition of clock and end of valid input data.
t_{IH2}	Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with $\overline{OE}$ Low).
t_{OE1}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD1}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{PPR}	Delay between V_{CC} (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
t_{PD}	Propagation delay between combinational inputs and outputs.
t_{OE2}	Delay between predefined Output Enable High, and when combinational Outputs become valid.
t_{OD2}	Delay between predefined Output Enable Low and when combinational Outputs are in the OFF-State.
t_{PRO}	Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

TIMING DIAGRAMS (Continued)



* Preset and Reset functions override Clock. However, F outputs may glitch with the first positive Clock Edge if t_{IS1} cannot be guaranteed by the user.

Asynchronous Preset/Reset



Flip-Flop Input Mode

Programmable logic sequencer (20 × 45 × 12)

PLS179

LOGIC PROGRAMMING

The PLS179 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

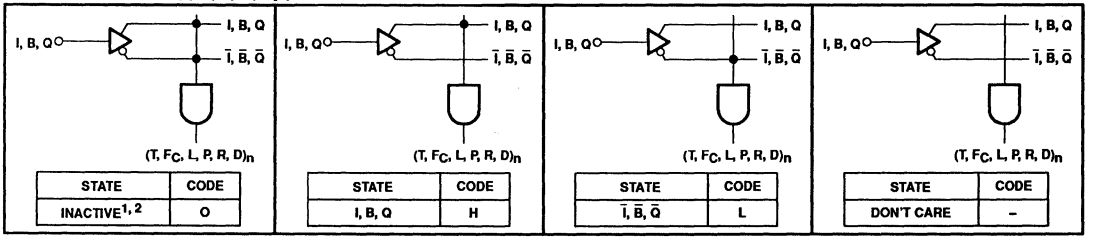
PLS179 logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I), (B), (Qp)



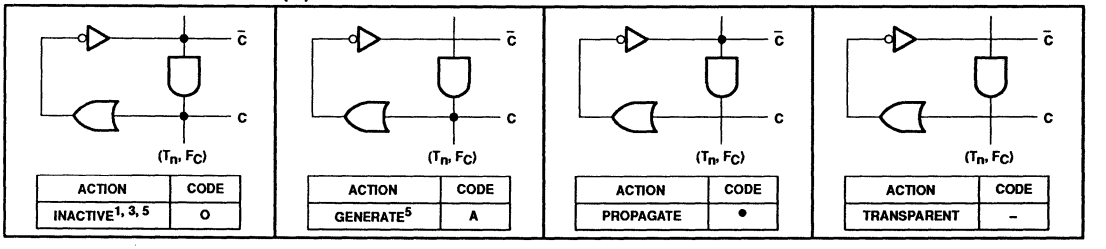
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, B, Q	H

STATE	CODE
$\bar{I}, \bar{B}, \bar{Q}$	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



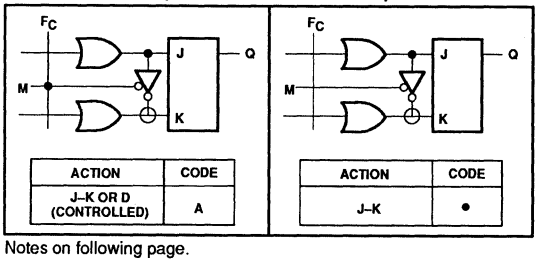
ACTION	CODE
INACTIVE[1, 3, 5]	O

ACTION	CODE
GENERATE[5]	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

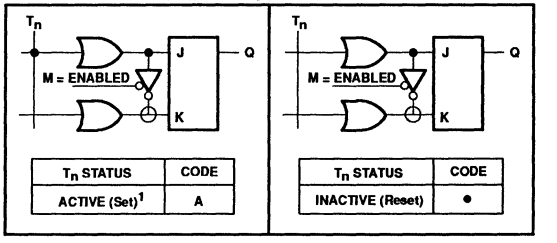
"OR" ARRAY – (F-F CONTROL MODE)



ACTION	CODE
J–K OR D (CONTROLLED)	A

ACTION	CODE
J–K	•

"OR" ARRAY – (Q_n = D-Type)



T_n STATUS	CODE
ACTIVE (Set)[1]	A

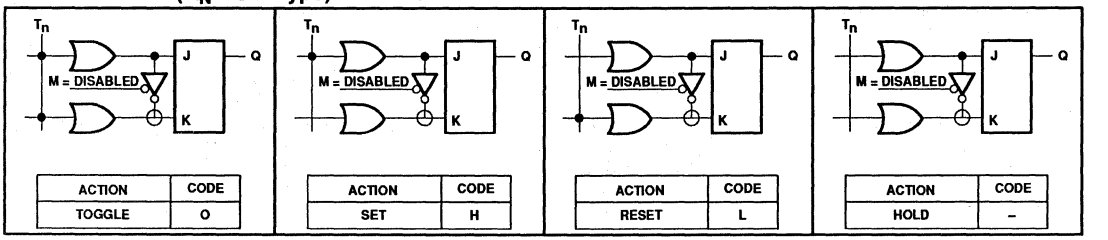
T_n STATUS	CODE
INACTIVE (Reset)	•

Notes on following page.

Programmable logic sequencer (20 × 45 × 12)

PLS179

"AND" ARRAY – (Q_N = J-K Type)



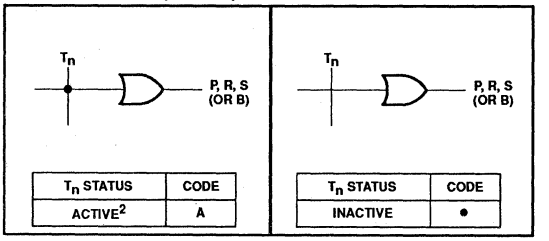
ACTION	CODE
TOGGLE	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

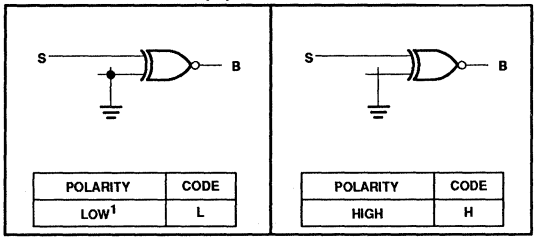
"OR" ARRAY – (S or B)



T_n STATUS	CODE
ACTIVE[2]	A

T_n STATUS	CODE
INACTIVE	•

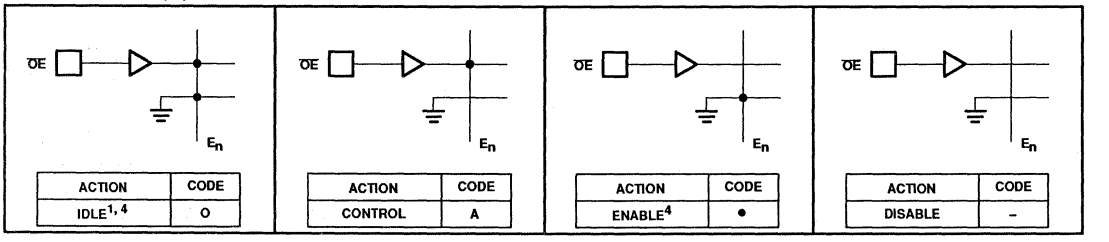
"EX-OR" ARRAY – (B)



POLARITY	CODE
LOW[1]	L

POLARITY	CODE
HIGH	H

"$\overline{OE}$" ARRAY – (E)



ACTION	CODE
IDLE[1, 4]	O

ACTION	CODE
CONTROL	A

ACTION	CODE
ENABLE[4]	•

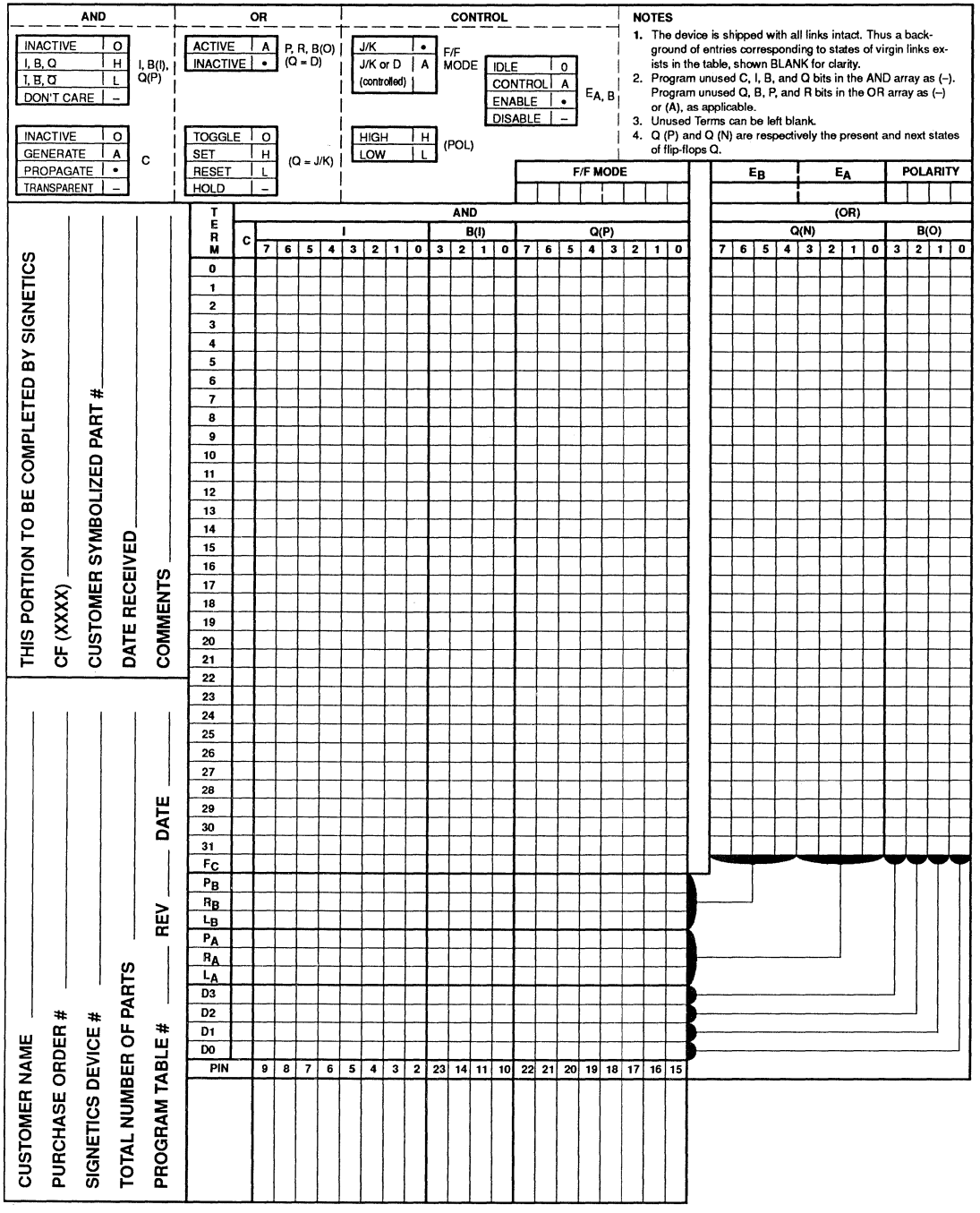
ACTION	CODE
DISABLE	–

NOTES:
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate (T, F_C, L, P, R, $D)_n$ will be unconditionally inhibited if any one of the I, B, or Q link pairs are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n, F_C.
4. E_n = O and E_n = • are logically equivalent states, since both cause F_n outputs to be unconditionally enabled.
5. These states are not allowed for control gates (L, P, R, $D)_n$ due to their lack of "OR" array links.

Programmable logic sequencer (20 × 45 × 12)

PLS179

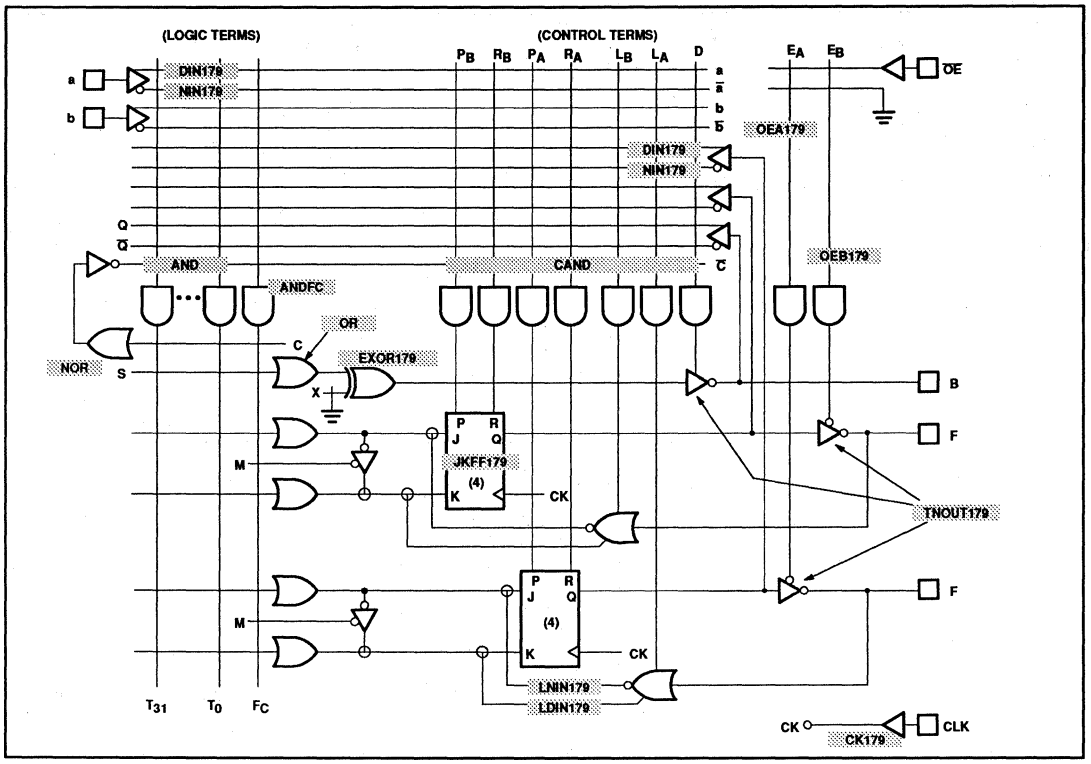
PROGRAM TABLE



Programmable logic sequencer (20 × 45 × 12)

PLS179

SNAP RESOURCE SUMMARY DESIGNATIONS



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

DESCRIPTION

The new PLC42VA12 CMOS PLD from Philips Semiconductors exhibits a unique combination of the two architectural concepts that revolutionized the PLD marketplace.

The Philips Semiconductors unique Output Macro Cell (OMC) embodies all the advantages and none of the disadvantages associated with the "V" type Output Macro Cell devices. This new design, combined with added functionality of two programmable arrays, represents a significant advancement in the configurability and efficiency of multi-function PLDs.

The most significant improvement in the Output Macro Cell structure is the implementation of the register bypass function. Any of the 10 J-K/D registers can be individually bypassed, thus creating a combinatorial I/O path from the AND array to the output pin. Unlike other "V" type devices, the register in the PLC42VA12 Macro Cell remains fully functional as a buried register. Both the combinatorial I/O and buried register have separate input paths (from the AND array). In most V-type architectures, the register is lost as a resource when the cell is configured as a combinatorial I/O. This feature provides the capability to operate the buried register independently from the combinatorial I/O.

The PLC42VA12 is an EPROM-based CMOS device. Designs can be generated using Philips Semiconductors SNAP PLD design software packages or one of several other commercially available JEDEC standard PLD design software packages.

FEATURES

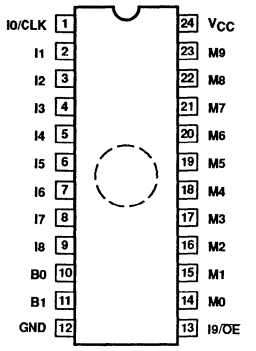
- High-speed EPROM-based CMOS Multi-Function PLD
  - Super set of 22V10, 32VX10 and 20RA10 PAL® ICs
- Two fully programmable arrays eliminate "P-term Depletion"
  - Up to 64 P-terms per OR function
- Improved Output Macro Cell Structure
  - Individually programmable as:
    - * Registered Output with feedback
    - * Registered Input
    - * Combinatorial I/O with Buried Register
    - * Dedicated I/O with feedback
    - * Dedicated Input (combinatorial)
  - Bypassed Registers are 100% functional with separate input and feedback paths
  - Individual Output Enable control functions
    - * From pin or AND array
- Reprogrammable – 100% tested for programmability
- Eleven clock sources
- Register Preload and Diagnostic Test Mode Features
- Security fuse

APPLICATIONS

- Mealy or Moore State Machines
  - Synchronous
  - Asynchronous
- Multiple, independent State Machines
- 10-bit ripple cascade
- Sequence recognition
- Bus Protocol generation
- Industrial control
- A/D Scanning

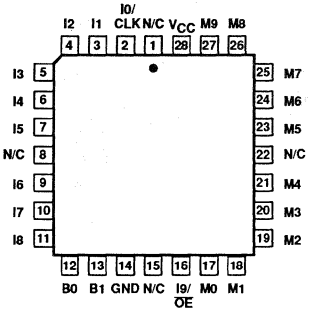
PIN CONFIGURATIONS

FA and N Packages



N = Plastic DIP (300mil-wide)
FA = Ceramic DIP with Quartz Window (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier (450mil-square)

ORDERING INFORMATION

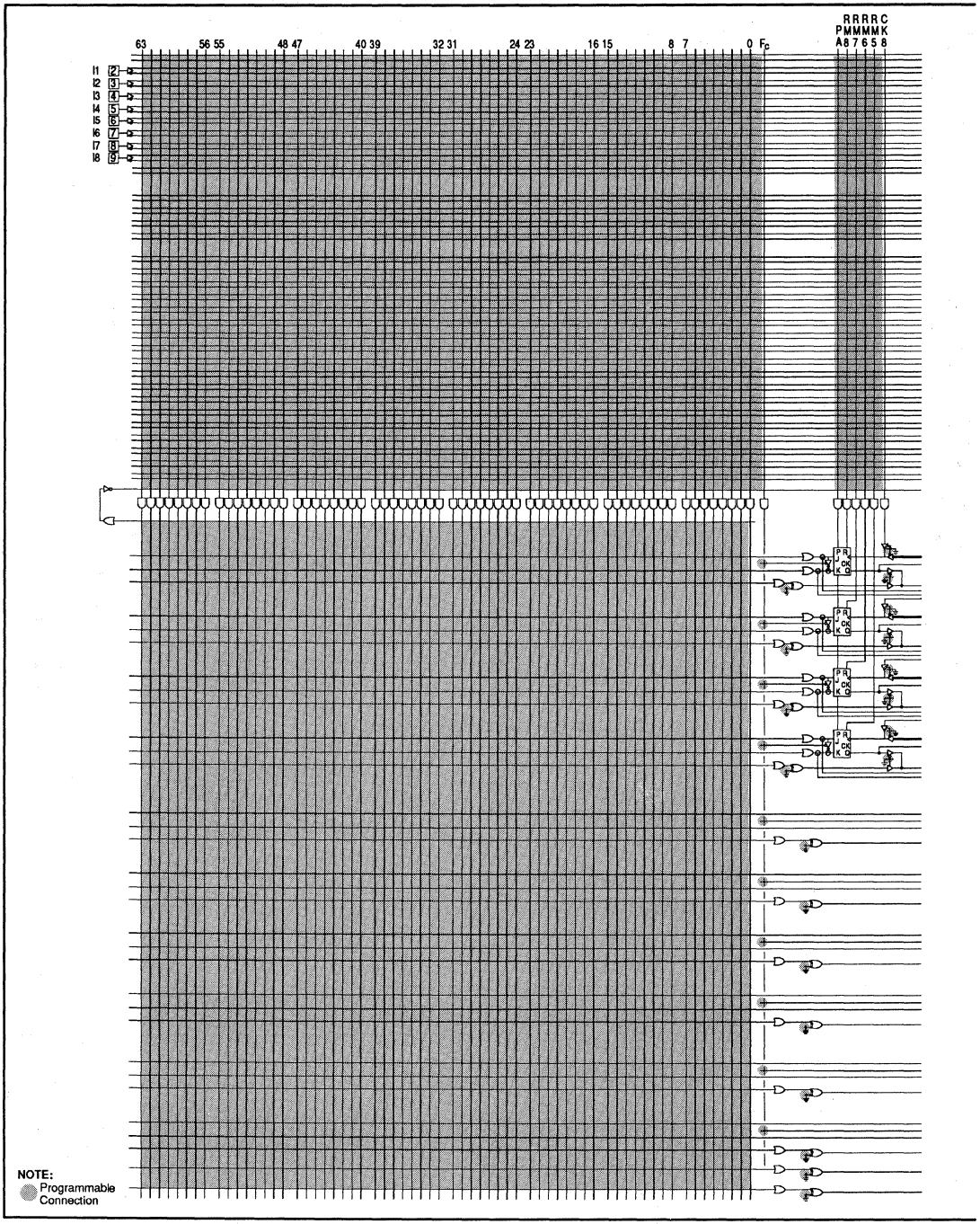
DESCRIPTION	ORDER CODE	DRAWING NUMBER
24-Pin Ceramic Dual In-Line with window, Reprogrammable (300mil-wide)	PLC42VA12FA	1478A
24-Pin Plastic Dual In-Line, One Time Programmable (300mil-wide)	PLC42VA12N	0410D
28-Pin Plastic Leaded Chip Carrier, One Time Programmable (450mil-wide)	PLC42VA12A	0401F

PAL is a registered trademark of Advanced Micro Devices, Inc.

CMOS programmable multi-function PLD ($42 \times 105 \times 12$)

PLC42VA12

LOGIC DIAGRAM

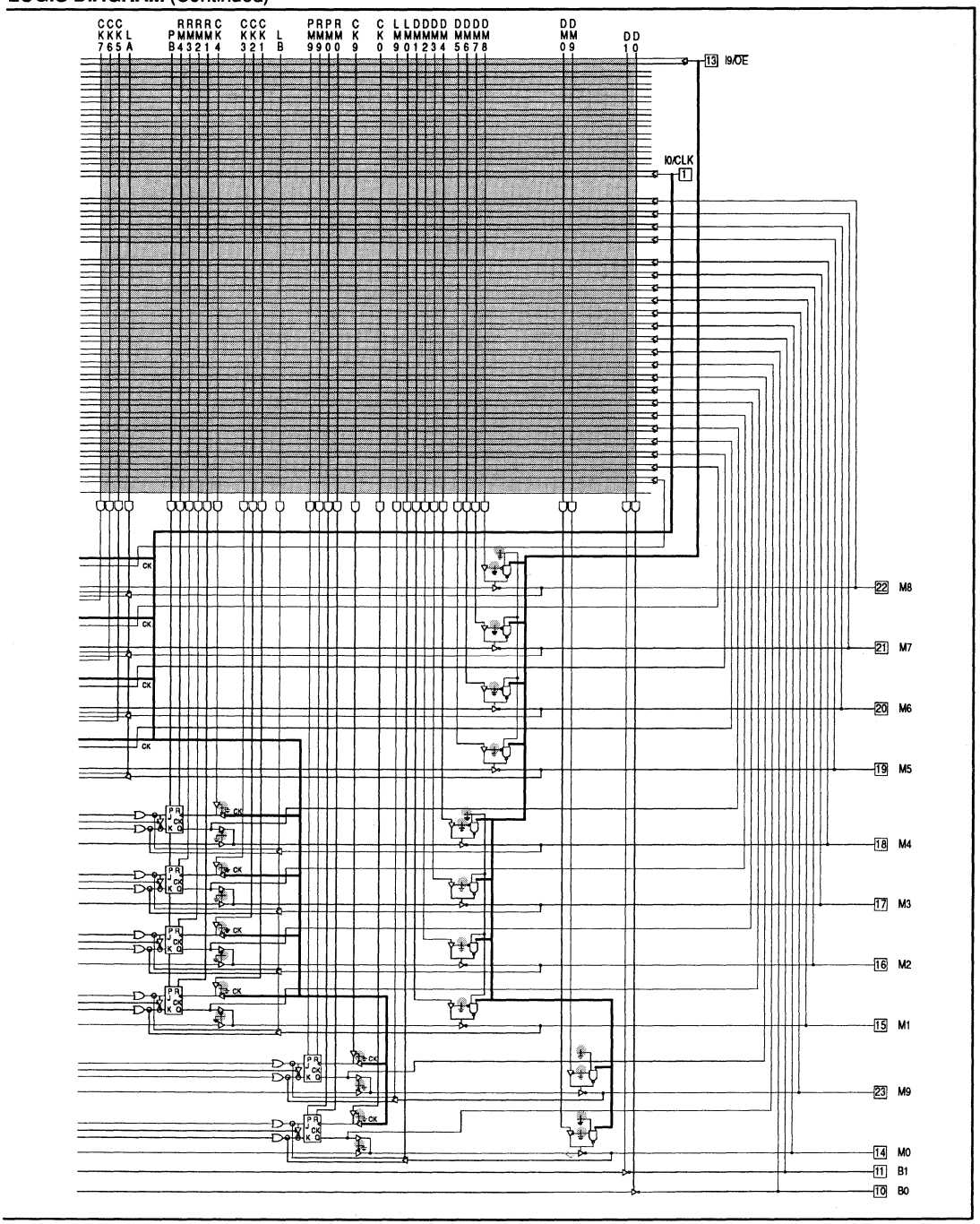


NOTE:
Programmable Connection

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

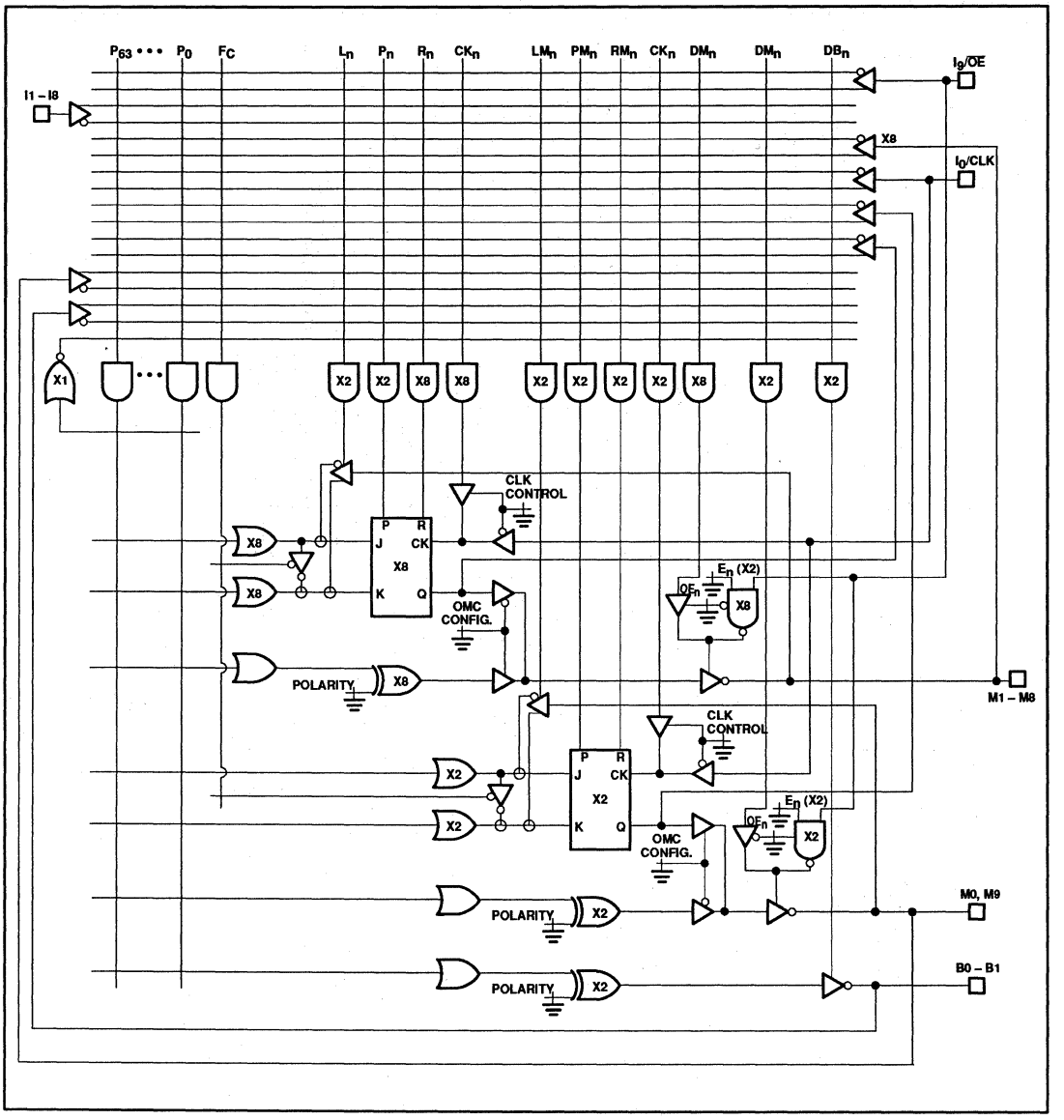
LOGIC DIAGRAM (Continued)



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

FUNCTIONAL DIAGRAM



CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	−0.5 to +7	V_{DC}
V_{IN}	Input voltage	−0.5 to V_{CC} +0.5	V_{DC}
V_{OUT}	Output voltage	−0.5 to V_{CC} +0.5	V_{DC}
I_{IN}	Input currents	−10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	−65 to +150	°C

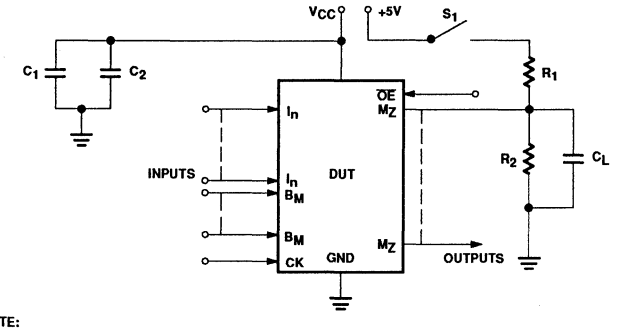
NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

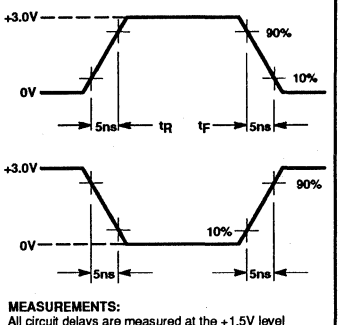
AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

Test Load Circuit

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN; I_{OL} = 16mA		0.3	0.5	V
V_{OH}	High	V_{CC} = MIN; I_{OH} = –3.2mA	2.4	4.3		V
Input current						
I_{IL}	Low	V_{IN} = GND		–1	–10	µA
I_{IH}	High	V_{IN} = V_{CC}		+1	10	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC} V_{OUT} = GND		1 –1	10 –10	µA µA
I_{OS}	Short-circuit[3,7]	V_{OUT} = GND			–130	mA
I_{CC1}	V_{CC} supply current (Active)[4]	I_{OUT} = 0mA, f = 15MHz[6], V_{CC} = MAX		90	120	mA
I_{CC2}	V_{CC} supply current (Active)[5]	I_{OUT} = 0mA, f = 15MHz[6], V_{CC} = MAX		70	100	mA
Capacitance						
C_I	Input	V_{CC} = 5V; V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short–circuit should not exceed one second. Test one at a time.
4. Tested with V_{IL} = 0.45V, V_{IH} = 2.4V.
5. Tested with V_{IL} = 0V, V_{IH} = V_{CC}.
6. Refer to Figure 1, ΔI_{CC} vs Frequency (worst case). (Referenced from 15MHz)
   The I_{CC} increases by 1.5mA per MHz for the frequency range of 16MHz up to 25MHz.
   The I_{CC} remains at a worst case for the frequency range of 26MHz up to 37MHz.
   The I_{CC} decreases by 1.0mA per MHz for the frequency range of 14MHz down to 1MHz.
   The worst case I_{CC} is calculated as follows:
   - All dedicated inputs are switching.
   - All OMCs are configured as JK flip-flops in the toggle mode. . .all are toggling.
   - All 12 outputs are disabled.
   - The number of product terms connected does not impact the I_{CC}.
7. Refer to Figure 2 for Δt_{PD} vs output capacitance loading.

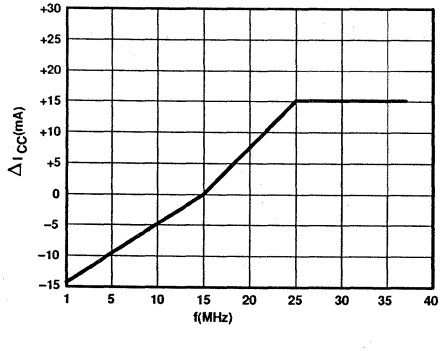


Figure 1. ΔI_{CC} vs Frequency (Worst Case) (Referenced from 15MHz)

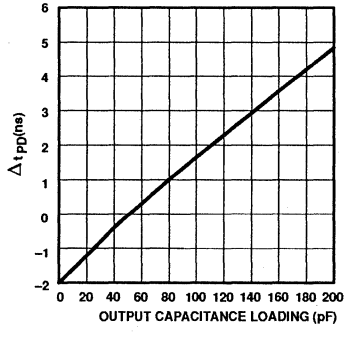


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$; $R_1 = 238\Omega$, $R_2 = 170\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST[2] CONDITION (C_L (pF))	PLC42VA12 MIN	PLC42VA12 TYP[1]	PLC42VA12 MAX	UNIT
Set-up Time								
t_{IS1}	Input; dedicated clock	(I, B, M) +/–	CK+	50	23	16		ns
t_{IS2}	Input; P-term clock	(I, B, M) +/–	(I, B, M) +/–	50	20	13		ns
t_{IS3}[3]	Preload; dedicated clock	(M) +/–	CK+	50	10	3.5		ns
t_{IS4}[3]	Preload; P-term clock	(M) +/–	(I, B, M) +/–	50	2	–1.0		ns
t_{IS5}[3]	Input through complement array; dedicated clock	(I, B, M) +/–	CK+	50	50	34		ns
t_{IS6}[3]	Input through complement array; P-term clock	(I, B, M) +/–	(I, B, M) +/–	50	40	30		ns
Propagation Delay								
t_{PD1}	Propagation Delay	(I, B, M) +/–	(I, B, M) +/–	50		20	35	ns
t_{PD2}	Propagation Delay with complement array (2 passes)	(I, B,) +/–	(I, B, M) +/–	50		36	55	ns
t_{CKO1}	Clock to Output; Dedicated clock	CK+	(M) +/–	50		13	17	ns
t_{CKO2}	Clock to output; P-term clock	(I, B, M) +/–	(M) +/–	50		18	27	ns
t_{RP1}	Registered operating period; Dedicated clock ($t_{IS1} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50		29	40	ns
t_{RP2}	Registered operating period; P-term clock ($t_{IS2} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50		31	47	ns
t_{RP3}[3]	Register preload operating period; Dedicated clock ($t_{IS3} + t_{CKO1}$)	(M) +/–	(M) +/–	50		16.5	27	ns
t_{RP4}[3]	Register preload operating period; P-term clock ($t_{IS4} + t_{CKO2}$)	(M) +/–	(M) +/–	50		17	29	ns
t_{RP5}[3]	Registered operating period with complement array; dedicated clock ($t_{IS5} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50		47	67	ns
t_{RP6}[3]	Registered operating period with complement array; P-term clock ($t_{IS6} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50		48	67	ns
t_{OE1}	Output Enable; from /OE pin[4]	/OE –	(M) +/–	50		10	20	ns
t_{OE2}	Output Enable; from P-term[4]	(I, B, M) +/–	(B, M) +/–	50		12.5	25	ns
t_{OD1}	Output Disable; from /OE pin[4]	/OE +	Outputs disabled	5		10	20	ns
t_{OD2}	Output Disable; from P-term[4]	(I, B, M) +/–	Outputs disabled	5		14.5	25	ns
t_{PRO}[3]	Preset to Output	(I, B, M) +/–	(M) +/–	50		25	35	ns
t_{PPR}[3]	Power-on Reset (Mn = 1)	V_{CC} +	(M) +/–	50			15	ns
Hold Time								
t_{IH1}	Input (Dedicated clock)	CK+	(I, B, M) +/–	50	0	–13		ns
t_{IH2}	Input (P-term clock)	(I, B, M) +/–	(I, B, M) +/–	50	5	–7.5		ns
t_{IH3}[3]	Input; from Mn (Dedicated clock)	CK+	(M) +/–	50	5	–1.5		ns
t_{IH4}[3]	Input; from Mn (P-term clock)	(I, B, M) +/–	(M) +/–	50	10	3.5		ns
Pulse Width								
t_{CKH1}	Clock High; Dedicated clock	CK+	CK–	50	10	5		ns
t_{CKL1}	Clock Low; Dedicated clock	CK–	CK+	50	10	5		ns
t_{CKH2}	Clock High; P-term clock	CK+	CK–	50	15	7		ns
t_{CKL2}	Clock Low; P-term clock	CK–	CK+	50	15	7		ns
t_{PRH}[3]	Width of preset/reset input pulse	(I, B, M) +/–	(I, B, M) +/–	50	30	7		ns

Notes on page 316.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

AC ELECTRICAL CHARACTERISTICS (Continued)

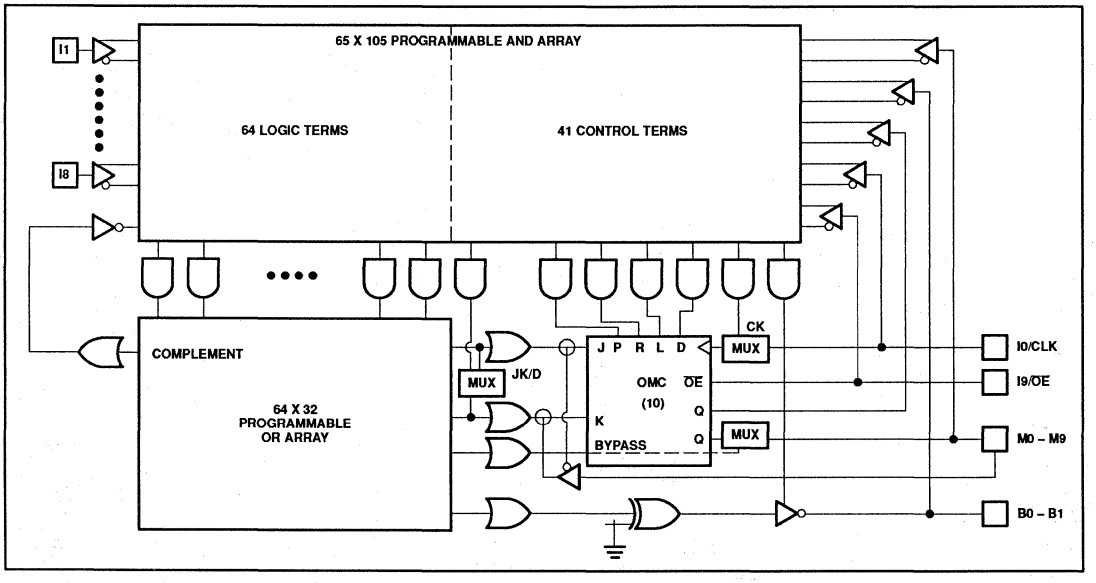
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$; $R_1 = 238\Omega$, $R_2 = 170\Omega$

SYMBOL	PARAMETER	FROM	TO	TEST[2] CONDITION (C_L (pF))	PLC42VA12 MIN	PLC42VA12 TYP[1]	PLC42VA12 MAX	UNIT
Frequency of Operation								
f_{CK1}	Dedicated clock frequency	C+	C+	50	50	100		MHz
f_{CK2}	P-term clock frequency	C+	C+	50	33	71.4		MHz
f_{MAX1}	Registered operating frequency; Dedicated clock ($t_{IS1} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50	25	34.5		MHz
f_{MAX2}	Registered operating frequency; P-term clock ($t_{IS2} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50	21.3	32.3		MHz
f_{MAX3}[3]	Register preload operating frequency; Dedicated clock ($t_{IS3} + t_{CKO1}$)	(M) +/–	(M) +/–	50	37	60.6		MHz
f_{MAX4}[3]	Register preload operating frequency; P-term clock ($t_{IS4} + t_{CKO2}$)	(M) +/–	(M) +/–	50	34.5	58.8		MHz
f_{MAX5}[3]	Registered operating frequency with complement array; Dedicated clock ($t_{IS5} + t_{CKO1}$)	(I, B, M) +/–	(M) +/–	50	14.9	21.3		MHz
f_{MAX6}[3]	Registered operating frequency with complement array; P-term clock ($t_{IS6} + t_{CKO2}$)	(I, B, M) +/–	(M) +/–	50	14.9	20.8		MHz

NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$. These limits are not tested/guaranteed.
2. Refer also to AC Test Conditions (Test Load Circuit).
3. These limits are not tested, but are characterized periodically and are guaranteed by design.
4. For 3-State output; output enable times are tested with $C_L = 50pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

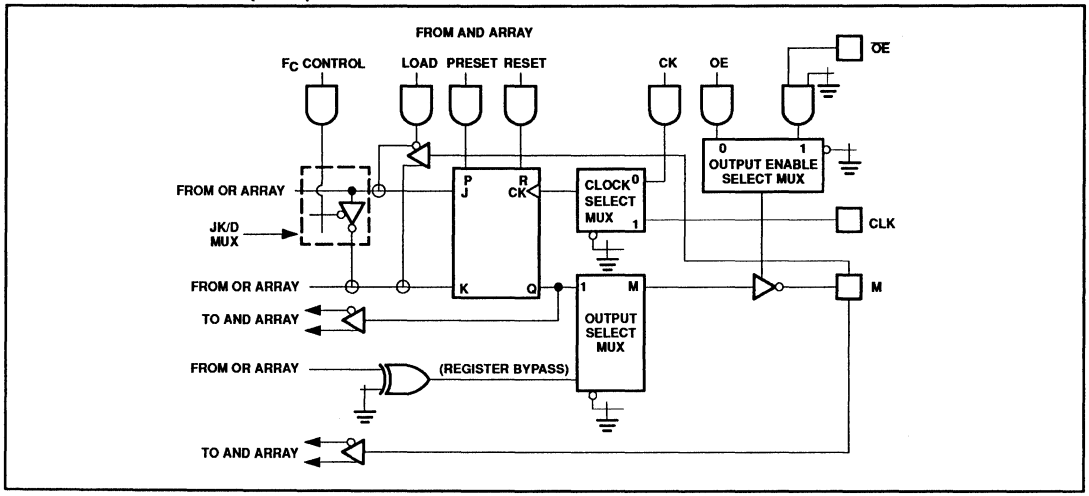
BLOCK DIAGRAM



CMOS programmable multi-function PLD (42 x 105 x 12)

PLC42VA12

OUTPUT MACRO CELL (OMC)



Output Macro Cell Configuration

Philips Semiconductors unique Output Macro Cell design represents a significant advancement in the configurability of multi-function Programmable Logic Devices.

The PLC42VA12 has 10 programmable Output Macro Cells. Each can be individually programmed in any of 5 basic configurations:

- Dedicated I/O (combinatorial) with feedback to AND array
- Dedicated Input
- Combinatorial I/O with feedback and Buried Register with feedback (register bypass)
- Registered Input
- Registered Output with feedback

Each of the registered options can be further customized as J-K type or D-type, with either an internally derived clock (from the AND array) or clocked from an external source. With these additional programmable options, it is possible to program each Output Macro Cell in any one of 14 different configurations.

These 14 configurations, combined with the fully programmable OR array, make the PLC42VA12 the most versatile and silicon efficient of all the Output Macro Cell-type PLDs.

The most significant Output Macro Cell (OMC) feature is the implementation of the register bypass function. Any of the 10 J-K/D registers can be individually bypassed, thus creating a combinatorial I/O path from the AND array to the output pin. Unlike other Output Macro Cell-type devices, the register in the OMC is fully functional as a buried register. Furthermore, both the combinatorial I/O and the buried register have separate input paths (from the AND array) and separate feedback paths (to the AND array). This feature provides the capability to operate the buried register independently from the combinatorial I/O.

The PLC42VA12 is ideally suited for both synchronous and asynchronous logic functions. Eleven clock sources – 10 driven from the AND array and one from an external source – make it possible to design synchronous state machine functions, event-driven state machine functions and combinatorial (asynchronous) functions all on the same chip.

Sophisticated control functions support individual OE control and Reset functions from the AND array. OE control is also available from the I9/$\overline{OE}$ pin. Register Preset and Load functions are controlled from the AND array, in 2 banks of 4 for OMCs M1 – M8. Output Macro Cells M0 and M9 have individual Preset and Load Control terms.

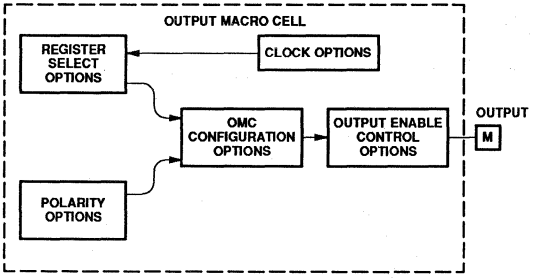
Output Polarity for the combinatorial I/O paths is configurable via 12 programmable EX-OR gates. The output of each register can be configured as inverting (active Low) or non-inverting (active High) via manipulation of the logic equations.

The output of each buried register can also be configured as inverting or non-inverting via the input buffer which feeds back to the AND array.

CMOS programmable multi-function PLD (42 x 105 x 12)

PLC42VA12

OUTPUT MACRO CELL PROGRAMMABLE OPTIONS



OMC Programmable Options

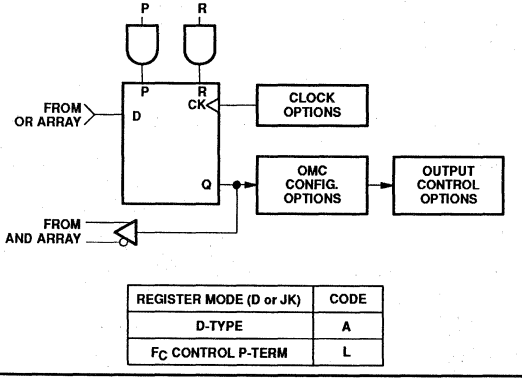
For purposes of programming, the Output Macro Cell should be considered to be partitioned into five separate blocks. As shown in the drawing titled "Output Macro Cell Programmable Options", the programmable blocks are: Register Select Options, Polarity Options, Clock Options, OMC Configuration Options and Output Enable Control Options.

There is one programmable location associated with each block except the Output Enable Control block which has two programmable fuse locations per OMC.

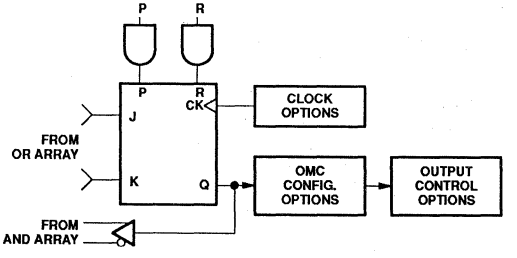
The following drawings detail the options associated with each programmable block. The associated programming codes are also included. The table titled "Output Macro Cell Configurations" (page 323) lists all the possible combinations of the five programmable options.

ARCHITECTURAL OPTIONS

REGISTER SELECT OPTIONS



REGISTER MODE (D or JK)	CODE
D-TYPE	A
F_C CONTROL P-TERM	L



REGISTER MODE (D or JK)	CODE
JK-TYPE	•
F_C CONTROL P-TERM	–

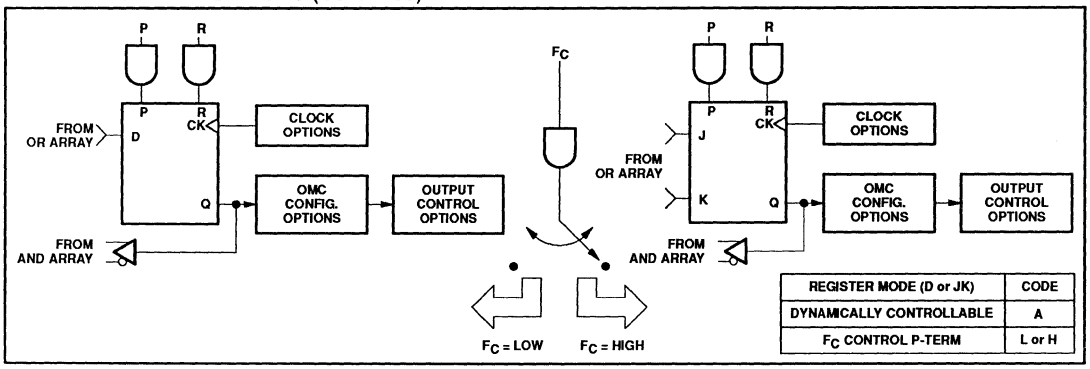
Register Select Options

Each OMC Register can be configured either as a dedicated D-type or a J-K flip-flop. The Flip-Flop Control term, Fc, provides the option to control each Register dynamically—switching from D-type to J-K type, based on the Fc control signal.

Register Preset and Reset are controlled from the AND array. Each OMC has an individual Reset Control term (RMn). The Register Preset function is controlled in two banks of 4 for OMCs M1 – M3 and M4 – M8 (via the control terms PA and PB). OMCs M0 and M9 have individual control terms (PM0 and PM9 respectively).

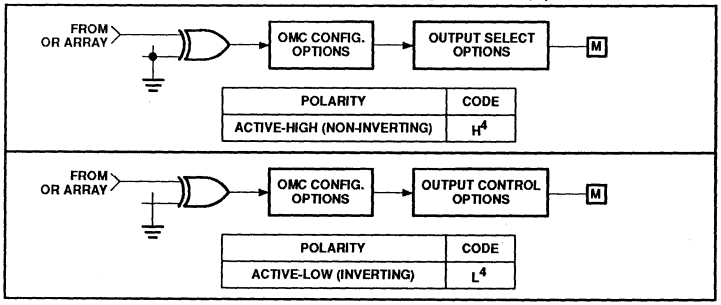
Notes on page 323.

REGISTER SELECT OPTIONS (Continued)



REGISTER MODE (D or JK)	CODE
DYNAMICALLY CONTROLLABLE	A
F_C CONTROL P-TERM	L or H

POLARITY OPTIONS (for Combinatorial I/O Configurations Only[1])



POLARITY	CODE
ACTIVE-HIGH (NON-INVERTING)	H[4]

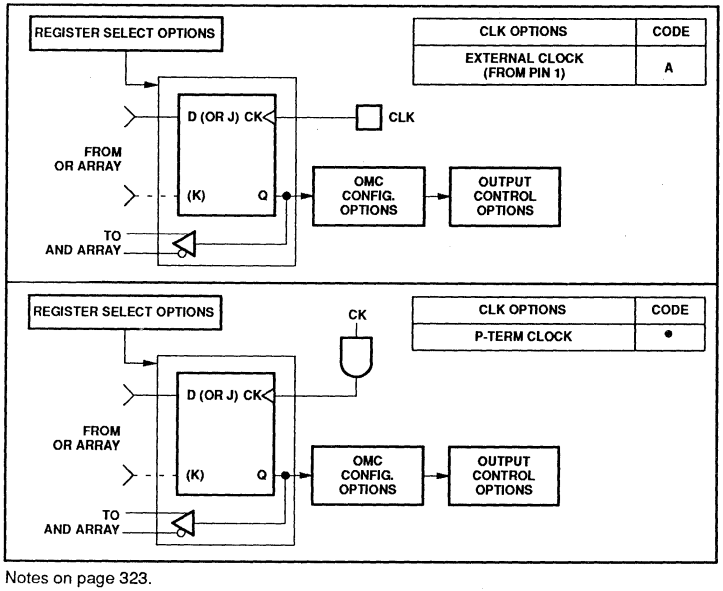
POLARITY	CODE
ACTIVE-LOW (INVERTING)	L[4]

Polarity Options

When an OMC is configured as a Combinatorial I/O with Buried Register, the polarity of the combinatorial path can be programmed as Active-High or Active-Low. A configurable EX-OR gate provides polarity control.

If an OMC is configured as a Registered Output, /Q is propagated to the output pin. Note that either Q or /Q can be fedback to the AND array by manipulating the feedback logic equations. (TRUE or COMPLEMENT).

CLOCK OPTIONS



CLK OPTIONS	CODE
EXTERNAL CLOCK (FROM PIN 1)	A

CLK OPTIONS	CODE
P-TERM CLOCK	•

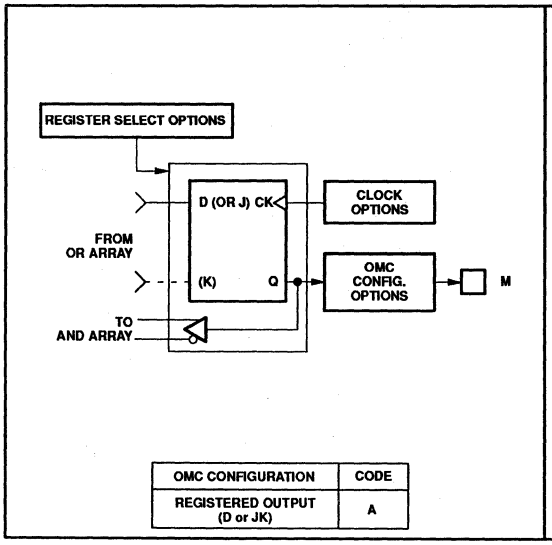
Clock Options

In the unprogrammed state, all Output Macro Cell clock sources are connected to the External Clock pin (I_0/CLK pin 1). Each OMC can be individually programmed such that its P-term Clock (CK_n) is enabled, thus disabling it from the External Clock and providing event-driven clocking capability.

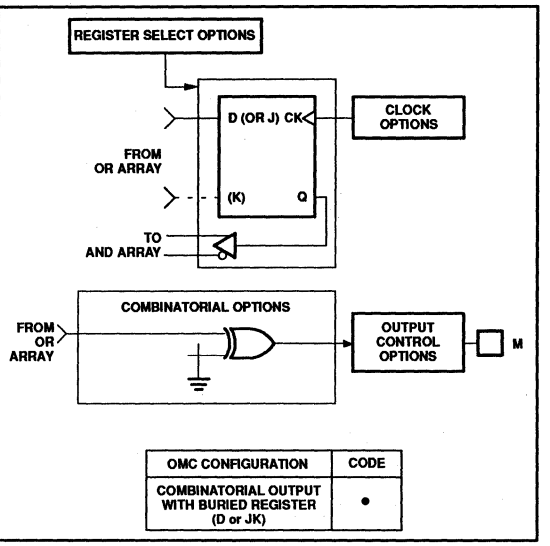
This feature supports multiple state machines, clocked at several different rates, all on one chip, or the ability to collect large amounts of random logic, including 10 separately clocked flip-flops.

Notes on page 323.

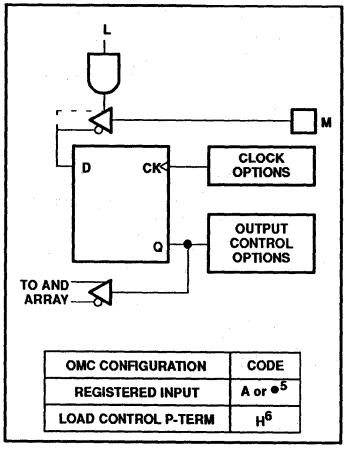
OUTPUT MACRO CELL CONFIGURATION OPTIONS



OMC CONFIGURATION	CODE
REGISTERED OUTPUT (D or JK)	A



OMC CONFIGURATION	CODE
COMBINATORIAL OUTPUT WITH BURIED REGISTER (D or JK)	•



OMC CONFIGURATION	CODE
REGISTERED INPUT	A or •[5]
LOAD CONTROL P-TERM	H[6]

Notes on page 323.

OMC Configuration Options

Each OMC can be configured as a Registered Output with feedback, a Registered Input or a Combinatorial I/O with Buried Register. Dedicated Input and dedicated I/O configurations are also possible.

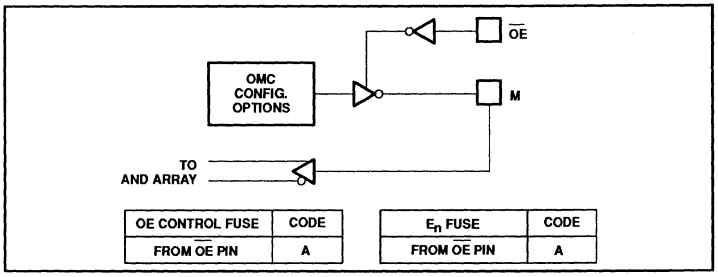
When the Combinatorial I/O option is selected, (the Register Bypass option), the Buried Register remains 100% functional, with its own inputs from the AND array and a separate feedback path. This unique feature is ideal for designing any type of state machine; synchronous Mealy-types that require both Buried and Output Registers, or asynchronous Mealy-types that require buried registers and combinatorial output functions. Both synchronous and asynchronous Moore-type state machines can also be easily accommodated with the flexible OMC structure.

Note that an OMC can be configured as either a Combinatorial I/O (with Buried Register) or a Registered Output with feedback and it can still be used as a Registered Input. By disabling the outputs via any OE control function, the M pin can be used as an input. When the Load Control P-term is asserted HIGH, the register is preloaded from the M pin(s). When the L_C P-term is Active-Low and the output is enabled, the OMC will again function as configured (either a combinatorial I/O or a registered output with feedback). This feature is suited for synchronizing input signals prior to commencing a state sequence.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

OUTPUT CONTROL OPTIONS



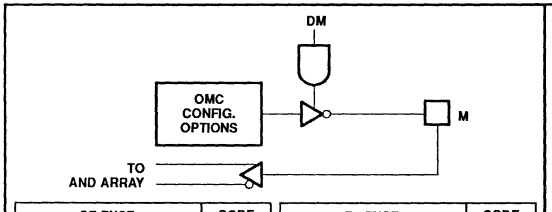
OE CONTROL FUSE	CODE
FROM $\overline{OE}$ PIN	A

E_n FUSE	CODE
FROM $\overline{OE}$ PIN	A

Output Enable Control Options

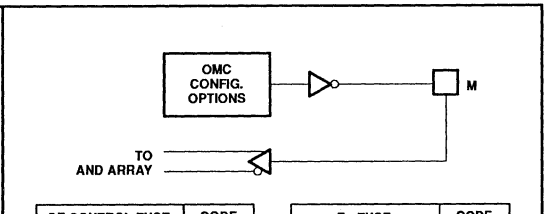
Similar to the Clock Options, the Output Enable Control for each OMC can be connected either to an external source (I9/$\overline{OE}$, pin 13) or controlled from the AND array (P-terms DM_n). Each Output can also be permanently enabled.

Output Enable control for the two bi-directional I/O (B pins 10 and 11) is from the AND array only (P-terms DB0 and DB1 respectively).



OE FUSE	CODE
FROM P-TERM CONTROL	•

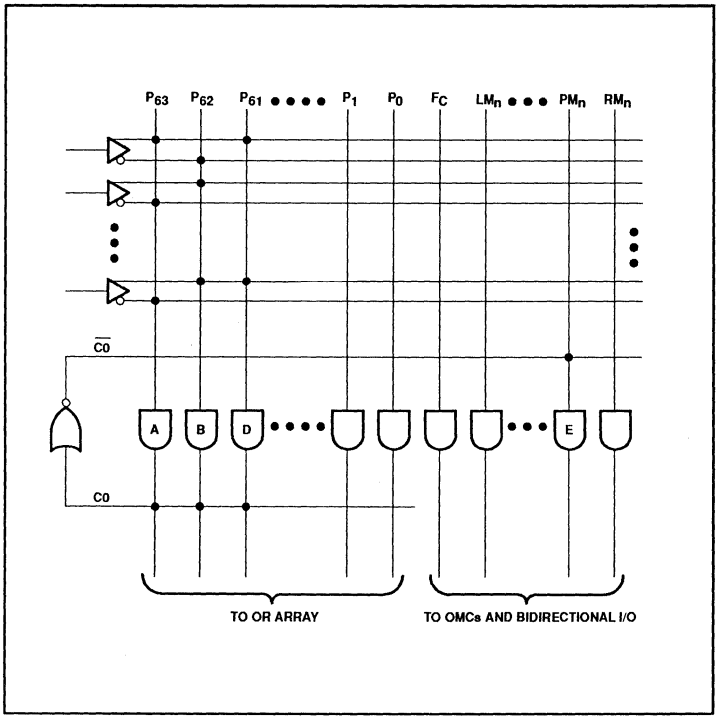
E_n FUSE	CODE
FROM P-TERM CONTROL	A or 0



OE CONTROL FUSE	CODE
ALWAYS ENABLED	A

E_n FUSE	CODE
ALWAYS ENABLED	0

COMPLEMENT ARRAY DETAIL



Complement Array Detail

The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $\overline{(A + B + C)}$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, such an approach can be prohibitive, both in terms of time and wasted resources.

Notes on page 323.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

LOGIC PROGRAMMING

The PLC42VA12 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLC42VA12 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC42VA12 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

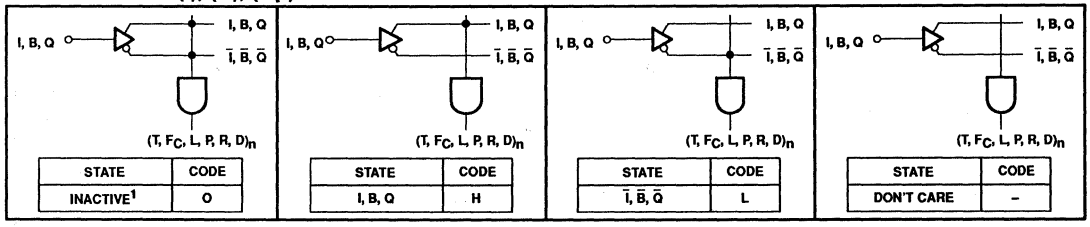
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below. Symbols for OMC configuration have been previously defined in the Architectural Options section.

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

LOGIC IMPLEMENTATION

"AND" ARRAY – (I), (B), (Qp)



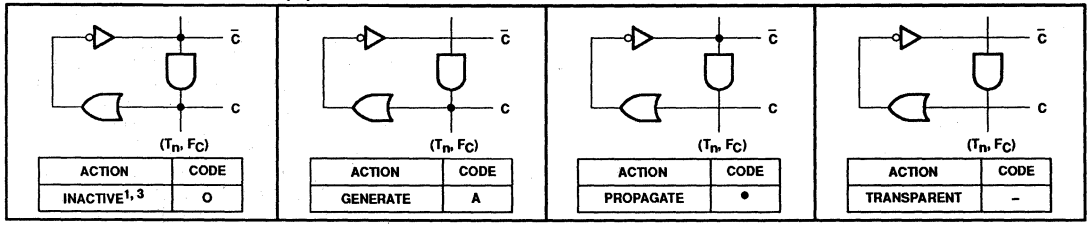
STATE	CODE
INACTIVE[1]	0

STATE	CODE
I, B, Q	H

STATE	CODE
Ī, B̄, Q̄	L

STATE	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



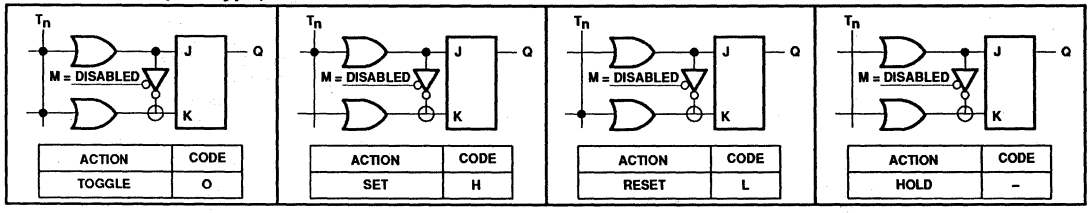
ACTION	CODE
INACTIVE[1, 3]	0

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

"OR" ARRAY – (J-K Type)



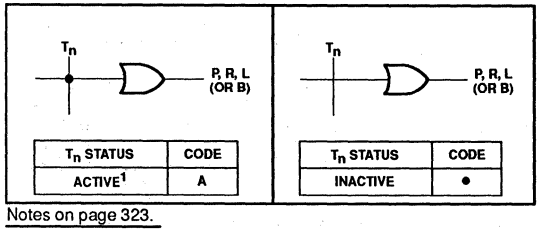
ACTION	CODE
TOGGLE	0

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
HOLD	–

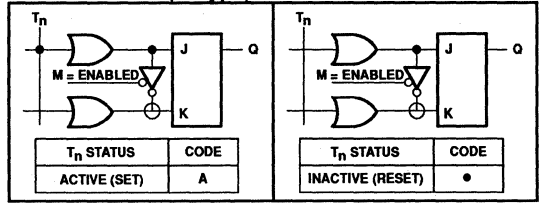
"OR" ARRAY



Tn STATUS	CODE
ACTIVE[1]	A

Tn STATUS	CODE
INACTIVE	•

"OR" ARRAY – (D-Type)



Tn STATUS	CODE
ACTIVE (SET)	A

Tn STATUS	CODE
INACTIVE (RESET)	•

Notes on page 323.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

LOGIC IMPLEMENTATION (Continued)

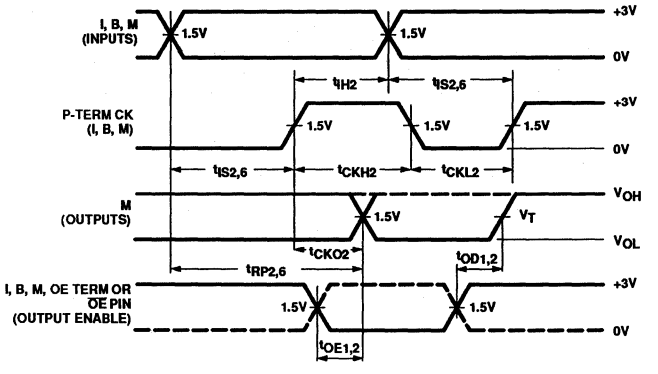
OUTPUT MACRO CELL CONFIGURATIONS

OUTPUT MACRO CELL CONFIGURATION	PROGRAMMING CODES			
	REGISTER SELECT FUSE	OMC CONFIGURATION FUSE	POLARITY FUSE	CLOCK FUSE
Combinatorial I/O with Buried D-type register				
External clock source	A	•	H or L	A
P-term clock source	A	•	H or L	•
Combinatorial I/O with Buried J-K type register				
External clock source	•	•	H or L	A
P-term clock source	•	•	H or L	•
Registered Output (D-type) with feedback				
External clock source	A	A	N/A	A
P-term clock source	A	A	N/A	•
Registered Output (J-K type) with feedback				
External clock source	•	A	N/A	A
P-term clock source	•	A	N/A	•
Registered Input (Clocked Preload) with feedback				
External clock source	A	A or •[5]	Optional[5]	A
P-term clock source	A	A or •[5]	Optional[5]	•

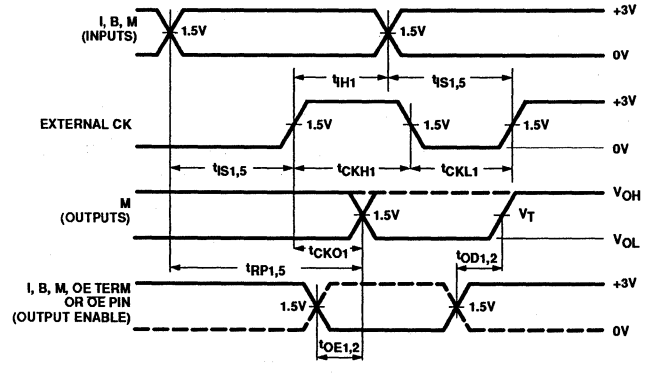
OUTPUT ENABLE CONTROL[8] CONFIGURATION	OUTPUT CONTROL FUSES		CONTROL SIGNAL
	OE CONTROL FUSE	En FUSES	
OMC controlled by /OE pin	A	A	
Output Enabled			Low
Output Disabled			High
OMC controlled by P-term	•	A or 0	
Output Enabled			High
Output Disabled			Low
Output always Enabled	A	0	Not Applicable

NOTES:
1. This is the initial (unprogrammed) state of the device.
2. Any gate will be unconditionally inhibited if both the TRUE and COMPLEMENT fuses are left intact.
3. To prevent oscillations, this state is not allowed for Complement Array fuse pairs that are coupled to active product terms.
4. The OMC Configuration fuse must be programmed as Combinatorial I/O in order to make use of the Polarity Option.
5. Regardless of the programmed state of the OMC Configuration fuse, an OMC can be used as a Registered Input. Note that the Load Control P-term must be asserted Active-High.
6. Output must be disabled.
7. Program code definitions:
   - A = Active (unprogrammed fuse)
   - 0, • = Inactive (programmed fuse)
   - – = Don't Care (both TRUE and COMPLEMENT fuses unprogrammed)
   - H = Active-High connection
   - L = Active-Low connection
8. OE control for B0 and B1 (Pins 10 and 11) is from the AND array only.

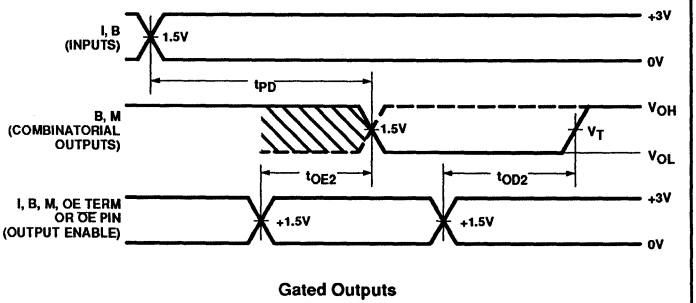
TIMING DIAGRAMS



Flip-Flop Outputs with P-term Clock



Flip-Flop Outputs with External Clock



Gated Outputs

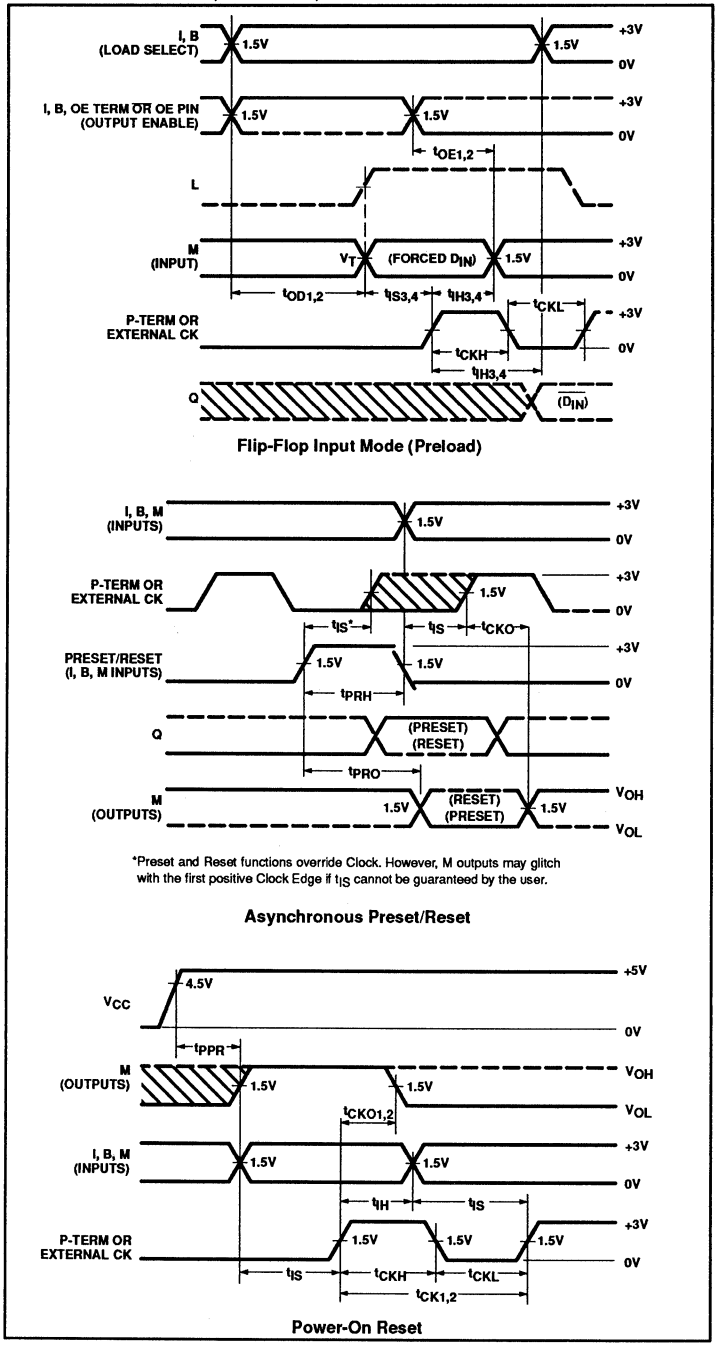
TIMING DEFINITIONS

SYMBOL	PARAMETER
f_{CK1}	Clock Frequency; External Clock
f_{CK2}	Clock Frequency; P-term Clock
t_{CKH1}	Width of Input Clock Pulse; External Clock
t_{CKH2}	Width of Input Clock Pulse; P-term Clock
t_{CKL1}	Interval between Clock pulses; External Clock
t_{CKL2}	Interval between Clock Pulses; P-term Clock
t_{CKO1}	Delay between the Positive Transition of External Clock and when M Outputs become valid.
t_{CKO2}	Delay between the Positive Transition of P-term Clock and when M Outputs become valid.
t_{RP1}	Delay between beginning of Valid Input and when the M outputs become Valid when using External Clock.
t_{RP2}	Delay between beginning of Valid Input and when the M outputs become Valid when using P-term Clock.
t_{RP3}	Delay between beginning of Valid Input and when the M outputs become Valid when using Preload Inputs (from M pins) and External Clock.
t_{RP4}	Delay between beginning of Valid Input and when the M outputs become valid when using Preload inputs (from M pins) and P-term Clock.
t_{RP5}	Delay between beginning of Valid Input and when the M outputs become Valid when using Complement Array and External clock.
t_{RP6}	Delay between beginning of Valid Input and when the M outputs become Valid when using Complement Array and P-term Clock.
f_{MAX1}	Minimum guaranteed Operating Frequency; Dedicated Clock
f_{MAX2}	Minimum guaranteed Operating Frequency; P-term Clock
f_{MAX3}	Minimum guaranteed Operating Frequency using Preload; Dedicated Clock (M pin to M pin)
f_{MAX4}	Minimum guaranteed Operating Frequency using Preload; P-term Clock (M pin to M pin)
f_{MAX5}	Minimum guaranteed Operating Frequency using Complement Array; Dedicated Clock
f_{MAX6}	Minimum Operating Frequency using Complement Array; P-term Clock
t_{IH1}	Required delay between positive transition of External Clock and end of valid input data.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

TIMING DIAGRAMS (Continued)



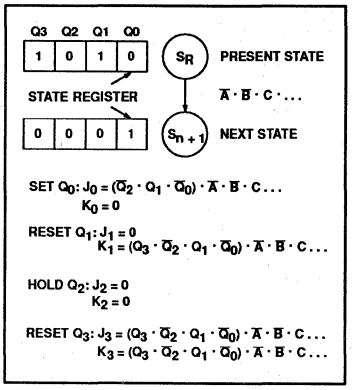
TIMING DEFINITIONS (Continued)

SYMBOL	PARAMETER
t_{IH2}	Required delay between positive transition of P-term Clock and end of valid input data.
t_{IH3}	Required delay between positive transition of External Clock and end of valid input data when using Preload Inputs (from M pins).
t_{IH4}	Required delay between positive transition of P-term Clock and end of valid input data when using Preload Inputs (from M pins).
t_{IS1}	Required delay between beginning of valid input and positive transition of External Clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of P-term Clock input.
t_{IS3}	Required delay between beginning of valid Preload input (from M pins) and positive transition of External Clock.
t_{IS4}	Required delay between beginning of valid Preload input (from M pins) and positive transition of P-term Clock input.
t_{IS5}	Required delay between beginning of valid input through Complement Array and positive transition of External Clock.
t_{IS6}	Required delay between beginning of valid input through Complement Array and positive transition of P-term Clock input.
t_{OE1}	Delay between beginning of Output Enable signal (Low) from /OE pin and when Outputs become valid.
t_{OE2}	Delay between beginning of Output Enable signal (High or Low) from OE P-term and when Outputs become valid.
t_{OD1}	Delay between beginning of Output Enable signal (HIGH) from /OE pin and when Outputs become disabled.
t_{OD2}	Delay between beginning of Output Enable signal (High or Low) from OE P-term and when Outputs become disabled.
t_{PD}	Delay between beginning of valid input and when the Outputs become valid (Combinatorial Path).
t_{PRH}	Width of Preset/Reset Pulse.
t_{PRO}	Delay between beginning of valid Preset/Reset Input and when the registered Outputs become Preset ("1") or Reset ("0").
t_{PPR}	Delay between V_{CC} (after power-up) and when flip-flops become Reset to "0". Note: Signal at Output (M pin) will be inverted.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

LOGIC FUNCTION



NOTE:
Similar logic functions are applicable for D mode flip-flops.

FLIP-FLOP TRUTH TABLE

$\overline{OE}$	L_n	CK_n	P_n	R_n	J	K	Q	M
H								Hi-Z
L	X	X	X	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

NOTES:
1. Positive Logic:
   $J\text{-}K = T_0 + \overline{T}_1 + T_2 + \ldots + T_{31}$
   $T_n = \overline{C} \cdot (I0 \cdot I1 \cdot I2 \ldots) \cdot (Q0 \cdot Q1 \ldots) \cdot (B0 \cdot B1 \ldots)$
2. ↑ denotes transition for Low to High level.
3. X = Don't care
4. * = Forced at M_n pin for loading the J-K flip-flop in the Input mode. The load control term, L_n must be enabled (HIGH) and the p-terms that are connected to the associated flip-flop must be forced LOW (disabled) during Preload.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. ** = Forced at F_n pin to load J/K flip-flop (Diagnostic mode).

PLC42VA12 UNPROGRAMMED STATE

A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

The following are:

ACTIVE:
- OR array logic terms
- Output Macro Cells M1 – M8;
  - D-type registered outputs (D = 0)
- External clock path
- Inputs: B0, B1, M0, M9

INACTIVE:
- AND array logic and control terms (except flip-flop mode control term, F_C)
- Bidirectional I/O (B0, B1);
  - Inputs are active. Outputs are 3-Stated via the OE P-terms, D0 and D1.
  - D-type registers (D = 0).
- Output Macro Cells M0 and M9;
  - Bidirectional I/O, 3-Stated via the OE P-terms, DM0 and DM9. The inputs are active.
- P-term clocks
- Complement Array
- J-K Flip-Flop mode

PROGRAMMING AND SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* in this data handbook for additional information.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC42VA12 devices are such that erasure begins to occur upon exposure to light with wavelength shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PLC42VA12 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC42VA12 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

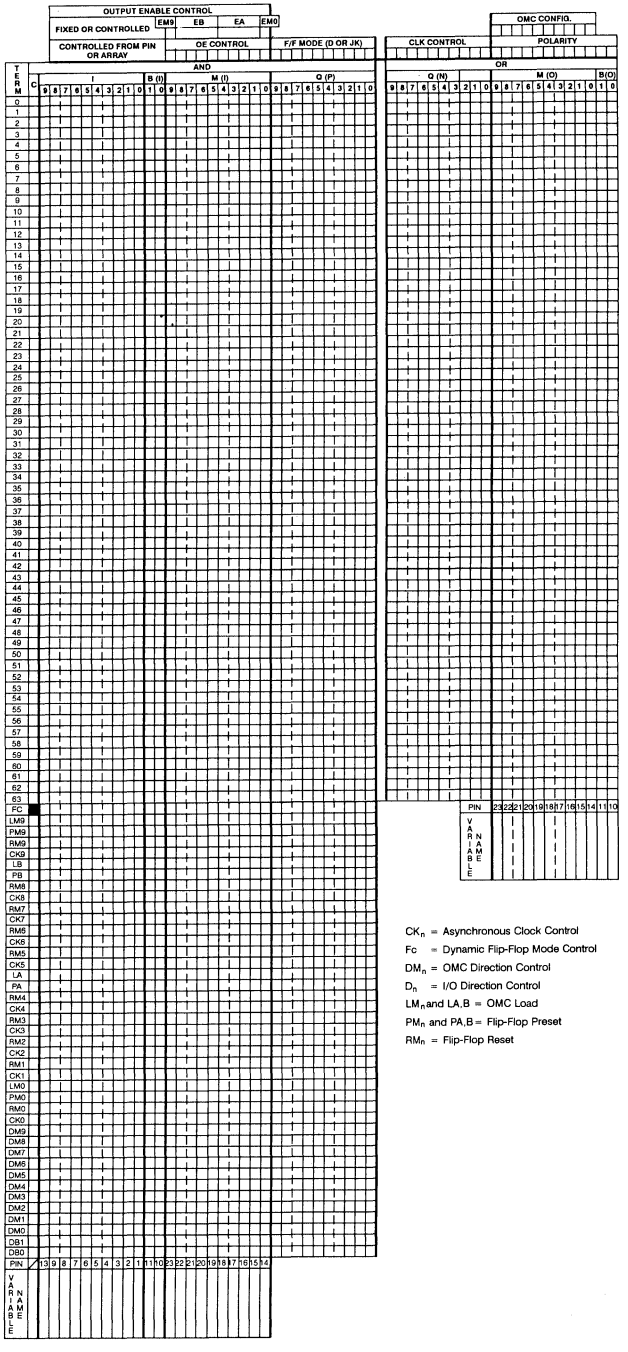
The recommended erasure procedure for the PLC42VA12 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000μW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12000μW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retentions exceeds 20 years.

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

PROGRAM TABLE

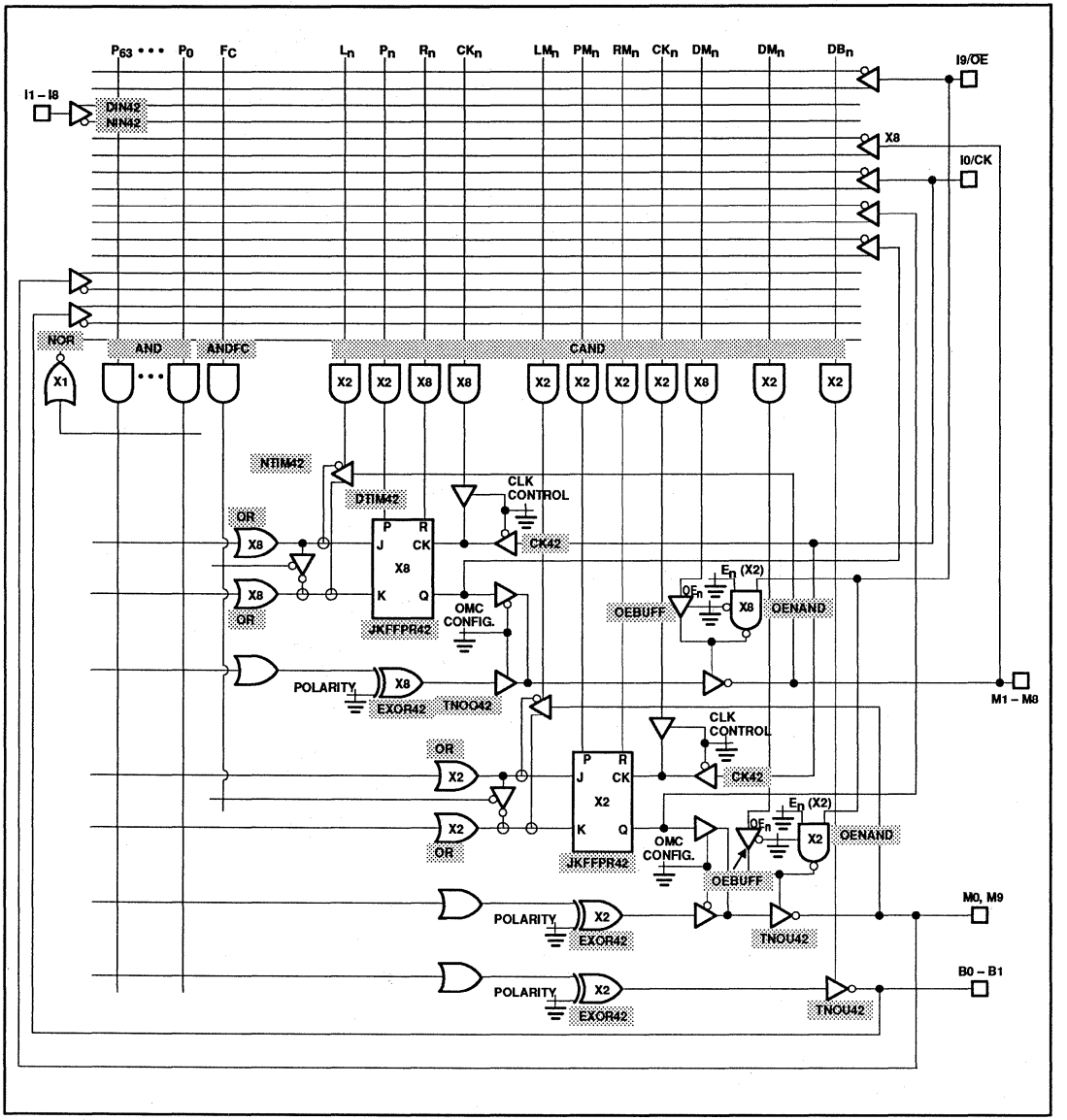


CK_n = Asynchronous Clock Control
Fc = Dynamic Flip-Flop Mode Control
DM_n = OMC Direction Control
D_n = I/O Direction Control
LM_n and LA,B = OMC Load
PM_n and PA,B = Flip-Flop Preset
RM_n = Flip-Flop Reset

CMOS programmable multi-function PLD (42 × 105 × 12)

PLC42VA12

SNAP RESOURCE SUMMARY DESIGNATIONS



CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

DESCRIPTION

The PLC415-16 PLD is a CMOS Programmable Logic Sequencer of the Mealy type. The PLC415-16 is a pin-for-pin compatible, functional superset of the PLS105 and PLUS405 Bipolar Programmable Logic Sequencer devices.

The PLC415 is ideally suited for high density, power sensitive controller functions. The Power Down feature provides true CMOS standby power levels of less than 100µA. The EPROM-based process technology supports operating frequencies of 16 to 20MHz. The PLC415-16 has been designed to accept both CMOS and TTL input levels to facilitate logic integration in almost any system environment.

The PLC415 architecture has been tailored for state machine functions. Both arrays are programmable, thus providing full interconnectability. Any one or all of the 64 AND transition terms can be connected to any (or all) of the 8 buried state and 8 output registers.

Two clock sources enable the design of 2 state machines on one chip. Separate INIT functions and Output Enable functions for each are controllable either from the array or from an external pin. The J-K flip-flops provide the added flexibility of the toggle function which is indeterminate on S-R flip-flops. The programmable Initialization feature supports asynchronous initialization of the state machine to any user defined pattern.

The unique Complement Array feature supports complex ELSE transition statements with a single product term. The PLC415-16 has 2 Complement Arrays which allows the user to design two independent complement functions. This is particularly useful if two state machines have been implemented on one chip.

FEATURES

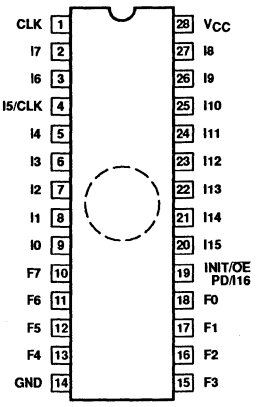
- Pin-for-Pin compatible, functional superset of PLS105/A and PLUS405 Logic Sequencers
- Zero standby power of less than 100µA (worst case)
  - Power dissipation at f_{MAX} = 80mA (worst case)
- CMOS and TTL compatible
- Programmable asynchronous Initialization and OE functions
  - Controllable from AND Array or external source
- 17 input variables
- 8 output functions
- 68 Product Terms
  - 64 transition terms
  - 4 control terms
- 8-bit State Register
- 8-bit Output Register
- 2 Transition Complement Arrays
- Multiple clocks
- Diagnostic test modes features for access to state and output registers
- Power-on preset of all registers to "1"
- J-K flip-flops
  - Automatic Hold states
- Security Fuse
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Arbitration functions
- Sequential circuits
- Security locking systems
- Counters
- Shift Registers

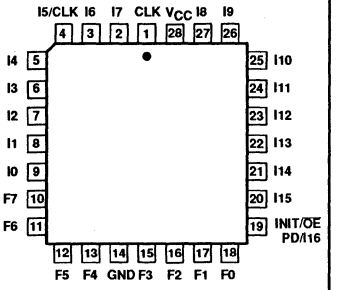
PIN CONFIGURATIONS

N, FA Packages



NOTE:
N = Plastic DIP (600mil-wide)
FA = Ceramic DIP with Quartz window (600mil-wide)

A Package



NOTE:
A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE	DRAWING NUMBER
28-Pin Ceramic DIP with window; Reprogrammable (600mil-wide)	f_{MAX} = 16MHz	PLC415–16FA	1478A
28-Pin Plastic DIP; One-Time Programmable (600mil-wide)	f_{MAX} = 16MHz	PLC415–16N	0413B
28-Pin Plastic Leaded Chip Carrier; One-Time Programmable (450mil-wide)	f_{MAX} = 16MHz	PLC415–16A	0401F

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers. Pin 1 only clocks P0–3 and F0–3 if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	I0–I4, I7, I6 I8–I9 I13–I15	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	I5/CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P4–7 and Output Registers F4–7, as above. Note that input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I12 is held at +11V, device outputs F0–F7 reflect the contents of State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I11 is held at +11V, device outputs F0–F7 become direct inputs for State Register bits P0–P7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I10 is held at +11V, device outputs F0–F7 become direct inputs for Output Register bits Q0–Q7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the Output Register bits Q0–Q7. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	F0–F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits Q0–Q7, when enabled. When I12 is held at +11V, F0–F7 = (P0–P7). When I11 is held at +11V, F0–F7 become inputs to State Register bits P0–P7. When I10 is held at +11V, F0–F7 become inputs to Output Register bits Q0–Q7.	Active-High (H)
19	INIT/$\overline{OE}$ I16/PD	**External Initialization, External /OE, PD or I16:** A user programmable function: Only one of the four options below may be selected. Note that both Initialization and /OE options are alternately available via the AND array. (P-terms INA, INB, OEA, and OEB.)	
		• **External Initialization:** Provides an asynchronous Preset to logic "1" or Reset to logic "0" of any or all State and Output Registers, determined individually on a register-by-register basis. INIT overrides the clock, and when held High, clocking is inhibited. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the INIT pulse goes Low. See timing diagrams for t_{NVCK} and t_{VCK}. Note that if the External Initialization option is selected, I16 is disabled automatically via the design software and the Power Down and External OE options are not available. Internal OE is available via P-Terms OEA and/or OEB. This option can be selected for one or both banks of registers.	Active-High (H)
		• **External Output Enable:** Provides an Output Enable/Disable function for Output Registers. Note that if the External OE option is selected, I16 is disabled automatically via the design software and the Power Down and External INIT options are not available. Internal INIT is available via P-terms INA and/or INB. This option can be selected for one or both banks of registers.	Active-Low (L)
		• **Power Down:** When invoked, provides a Power Down (zero power) mode. The contents of all Registers is retained, despite the toggling of the Inputs or the clocks. To obtain the lowest possible power level, all Inputs should be static and at CMOS input levels. Note that if the PD options is selected, I_{16} is disabled automatically via the design software and the External INIT and External OE options are not available. Internal INIT is available via P-terms INA and/or INB and Internal OE is available via P-terms OEA and/or OEB.	Active-High (H)
		• **Logic Input:** The 17th external logic input to the AND array as above. Note that when the I16 option is selected, the Power Down, External /OE and External INIT are not available. Internal OE and Internal INIT are available from P-Terms OEA/OEB and INA/INB, respectively.	Active-High/Low (H/L)

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

TRUTH TABLE [1, 2, 3, 4, 5]

V_{CC}	OPTION										
	INIT	$\overline{OE}$	I_{10}	I_{11}	I_{12}	CK	J	K	Q_P	Q_F	F
	H		X	X	X	X	X	X	H/L	H/L	Q_F
	X		+11V	X	X	↑	X	X	Q_P	L	L
	X		+11V	X	X	↑	X	X	Q_P	H	H
	X		X	+11V	X	↑	X	X	L	Q_F	L
	X		X	+11V	X	↑	X	X	H	Q_F	H
	X		X	X	+11V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
+5V		H	X	X	X	X	X	X	Q_P	Q_F	Hi-Z
		X	+11V	X	X	↑	X	X	Q_P	L	L
		X	+11V	X	X	↑	X	X	Q_P	H	H
		X	X	+11V	X	↑	X	X	L	Q_F	L
		X	X	+11V	X	↑	X	X	H	Q_F	H
		L	X	X	+11V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	L	L	X	X	X	X	X	X	H	H	

NOTES:

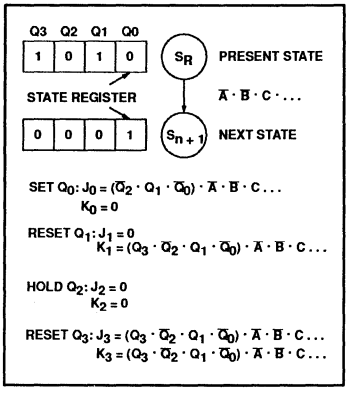
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   $T_n = (C_0, C_1)(I0, I1, I2, \ldots)$ (P0, P1 . . . P7)
2. ↑ denotes transition from Low-to-High level.
3. X = Don't Care (≤5.5V)
4. H/L implies that either a High or a Low can occur, depending upon user-programmed Initialization selection (each State and Output Register individually programmable).
5. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$/PD/I16 is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All J/K flip-flop inputs are disabled (0).
4. The Complement Arrays are inactive.
5. Clock 1 is connected to all State and Output Registers.

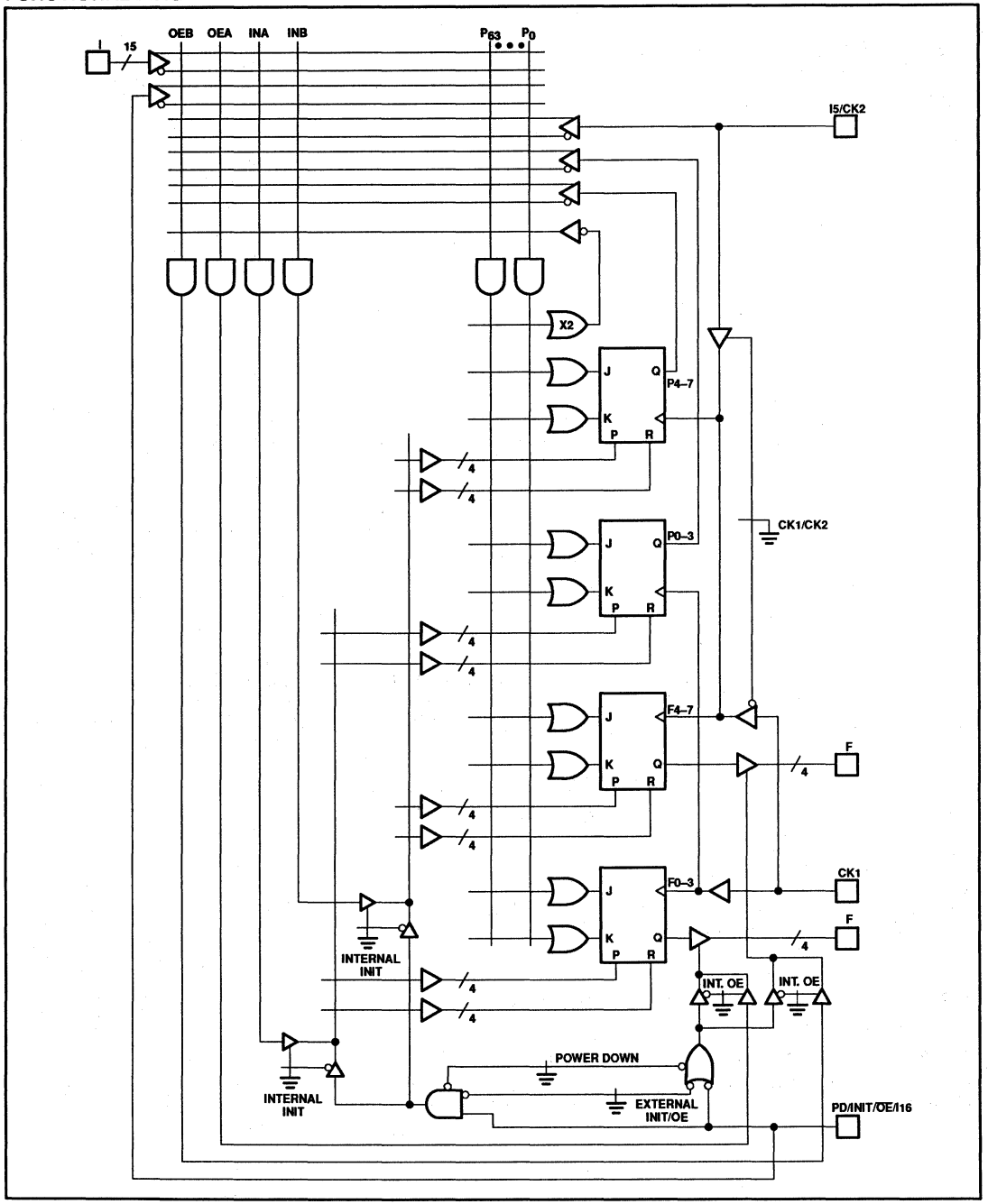
LOGIC FUNCTION



CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

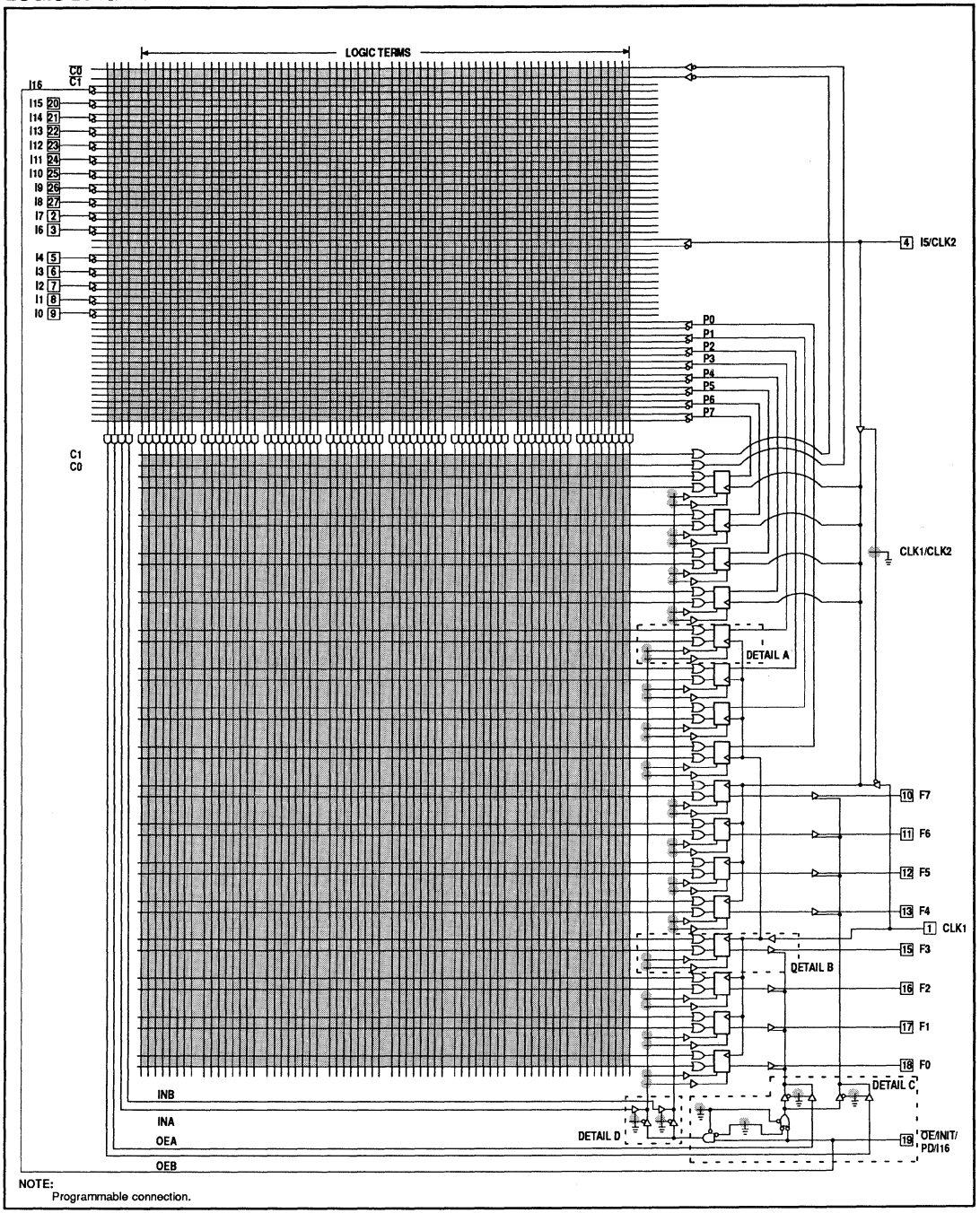
FUNCTIONAL DIAGRAM



CMOS programmable logic sequencer ($17 \times 68 \times 8$)

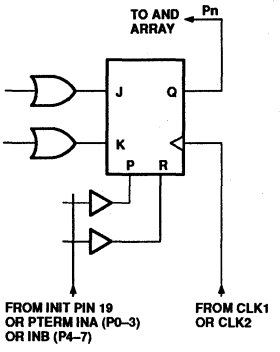
PLC415-16

LOGIC DIAGRAM

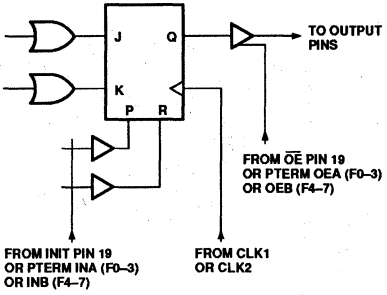


NOTE:
Programmable connection.

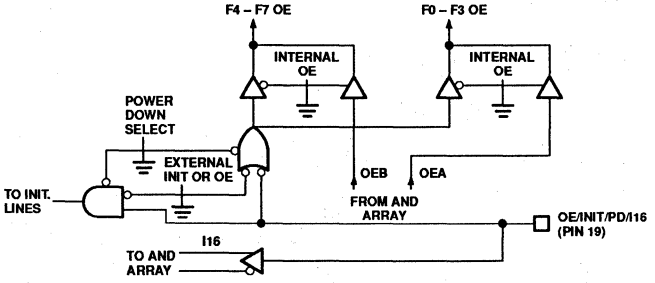
DETAILS FOR PLC415-16 LOGIC DIAGRAM



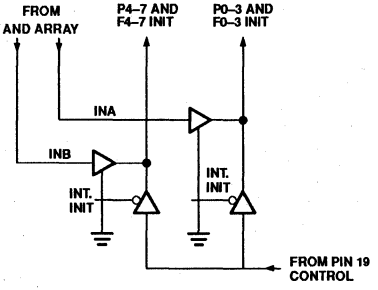
Detail A
State Registers P0 – P7



Detail B
Output Registers F0 – F7

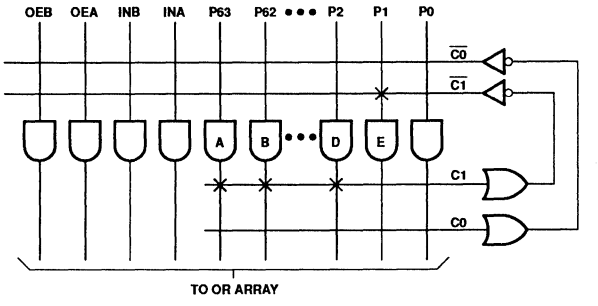


Detail C
Pin 19 Options: $\overline{OE}$, Initialization, Power Down and Input 16



Detail D
Internal and External Initialization

DETAILS FOR PLC415-16 LOGIC DIAGRAM (Continued)



Complement Array Detail

The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and ($\overline{A + B + C}$) are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, such an approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLC416-16 has 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTES:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 16mA			0.5	V
V_{OH}	High	I_{OH} = –3.2mA	2.4			V
Input current						
I_{IL}	Low	V_{IN} = GND			–10	µA
I_{IH}	High	V_{IN} = V_{CC}			10	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC} V_{OUT} = GND			10 –10	µA µA
I_{OS}	Short-circuit[3, 6]	V_{OUT} = GND			–130	mA
I_{CCSB}	V_{CC} supply current with PD asserted[7]	V_{CC} = MAX, V_{IN} = 0 or V_{CC}		50	100	µA
I_{CC}	V_{CC} supply current Active[4, 5] (TTL or CMOS Inputs)	I_{OUT} = 0mA, at f = 1MHz			55	mA
		V_{CC} = MAX, at f = MAX			80	mA
Capacitance						
C_I	Input	V_{CC} = 5V, V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

NOTES:

1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short-circuit should not exceed one second. Test one at a time.
4. Tested with TTL input levels: V_{IL} = 0.45V, V_{IH} = 2.4V. Measured with all inputs and outputs switching.
5. Refer to Figure 1, I_{CC} vs Frequency (worst case).
6. Refer to Figure 2 for Δt_{PD} vs output capacitance loading.
7. The outputs are automatically 3-Stated when the device is in the Power Down mode. To achieve the lowest possible current, the inputs and clocks should be at CMOS static levels.

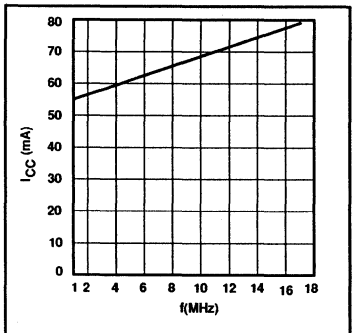


Figure 1. I_{CC} vs Frequency (Worst Case)

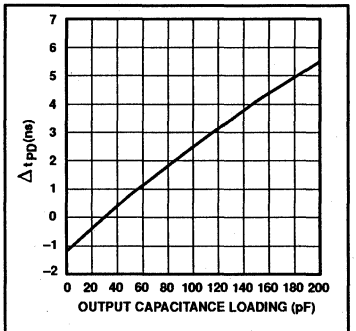


Figure 2. Δt_{PD} vs Output Capacitance Loading (Typical)

AC ELECTRICAL CHARACTERISTICS

$R_1 = 252\Omega$, $R_2 = 178\Omega$, $0°C \leq T_{amb} \leq +75°C$, $4.75 \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
Pulse width								
t_{CKH}	Clock High	CK+	CK–	30pF	25	10		ns
t_{CKL}	Clock Low	CK–	CK+	30pF	25	10		ns
t_{INITH}	Initialization Input pulse	INIT+	INIT–	30pF	20			ns
Set–up time								
t_{IS1}	Input	(I) +/–	CK+	30pF	38	25		ns
t_{IS2}[1]	Input through Complement array	(I) +/–	CK+	30pF	60	40		ns
t_{ISPD}	Power Down Setup (from PD pin)	PD+	CK+	30pF	38	15		ns
t_{ISPU}	Power Up Setup (from PD pin)	PD–	First Valid CK+	30pF	38	30		ns
t_{VS}[1]	Power on Preset Setup	V_{CC}+	CK–	30pF	0			ns
t_{VCK1}	Clock resume (after INIT) when using INIT pin (pin 19)	INIT–	CK–	30pF	10	–5		ns
t_{VCK2}[1]	Clock resume (after INIT) when using P-term INIT (from AND array)	(I) +/–	CK–	30pF	20	8		ns
t_{NVCK1}	Clock lockout (before INIT) when using INIT pin (pin 19)	CK–	INIT–	30pF	10	–3		ns
t_{NVCK2}[1]	Clock lockout (before INIT) when using P-term INIT (from AND array)	CK–	INIT–	30pF	0	–5		ns
Propagation delays								
t_{CKO}	Clock to Output	CK+	(F) +/–	30pF		15	22	ns
t_{PDZ}	Power Down to outputs off	PD+	Outputs Off	5pF		25	30	ns
t_{PUA1}	Power Up to outputs Active with dedicated Output Enable	PD–	Outputs Active	30pF		20	35	ns
t_{PUA2}[1]	Power Up to outputs Active with P-term Output Enable[1]	PD–	Outputs Active	30pF		37	55	ns
t_{IHPU}	Last valid clock to Power Down delay (Hold)	Last Valid Clock	PD+	30pF	25	15		ns
t_{IHPD}	First valid clock cycle before Power Up	Beginning of First Valid Clock Cycle	PD–	30pF	0	–25		ns
t_{OE1}[3]	Output Enable: from /OE pin	OE–	Output Enabled	30pF		15	30	ns
t_{OE2}[1]	Output Enable; from P–term	(I) +/–	Output Enabled	30pF		25	40	ns
t_{OD1}[3]	Output Disable; from /OE pin	OE+	Output Disabled	5pF		20	30	ns
t_{OD2}[3]	Output Disable; from P-term	(I) +/–	Output Disabled	5pF		30	40	ns
t_{INIT1}	INIT to output when using INIT pin	INIT+	(F) +/–	30pF		22	35	ns
t_{INIT2}	INIT to output when using P-term INIT	(I) +/–	(F) +/–	30pF		35	45	ns
t_{PPR}[1]	Power-on Preset (F_n = 1)	V_{CC} +	(F) +	30pF			15	ns
t_{RP1}	Registered operating period; (t_{IS1} + t_{CKO1})	(I) +/–	(F) +/–	30pF		40	60	ns
t_{RP2}[1]	Registered operating period with Complement Array (t_{IS2} + t_{CKO1})	(I) +/–	(F) +/–	30pF		55	75	ns

Notes on following page

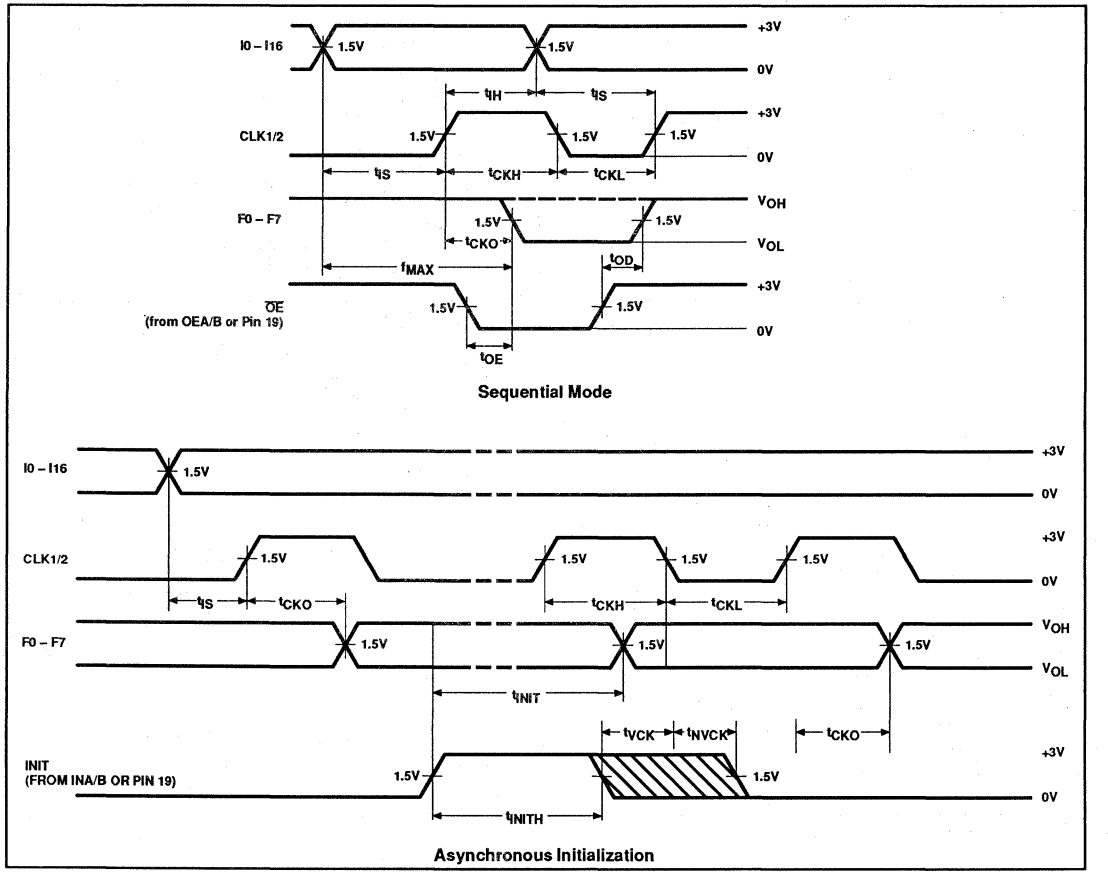
AC ELECTRICAL CHARACTERISTICS (Continued)

$R_1 = 252\Omega$, $R_2 = 178\Omega$, $0°C \le T_{amb} \le +75°C$, $4.75 \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT
Hold time								
t_{IH}	Input Hold	CK+	(F) +/–	30pF		–10	0	ns
Frequency of operation								
f_{CLK}[1]	Clock (toggle) frequency	C+	C+	30pF	20	50		MHz
f_{MAX1}	Registered operating frequency ($t_{IS1} + t_{CKO1}$)	(I) +/–	(F) +/–	30pF	16.7	25		MHz
f_{MAX2}	Registered operating frequency with Complement Array ($t_{IS2} + t_{CKO1}$)[1]	(I) +/–	(F) +/–	30pF	13.3	18.2		MHz

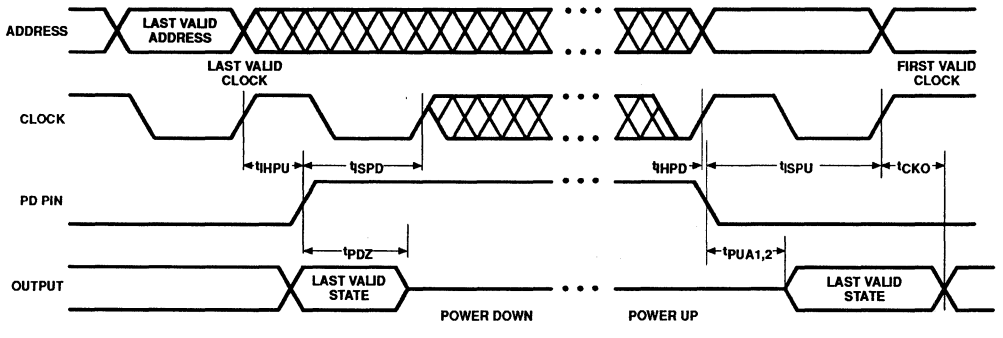
NOTE:
1. Not 100% tested, but guaranteed by design/characterization.
2. All propagation delays and setup times are measured and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS

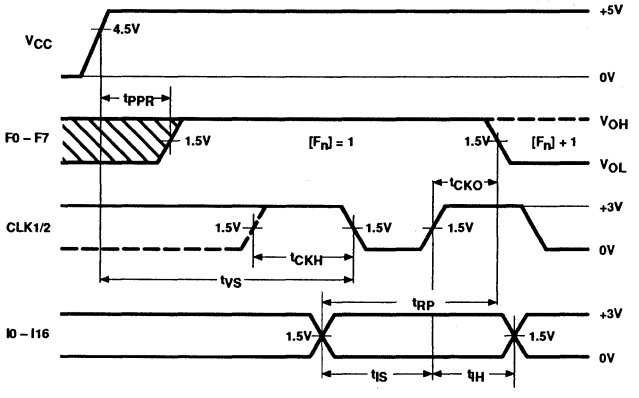


The PLC416-16 has a unique power down feature that is ideal for power sensitive controller and state machine applications. During idle periods, the PLC415 can be powered down to a near zero power consumption level of less than 100 micro Amps. Externally controlled from Pin 19, the power down sequence first saves the data in all the State and Output registers. In order to insure that the last valid states are saved, there are certain hold times associated with the first and last valid clock edges and the Power Down input pulse. The Outputs are then automatically 3-Stated and power consumption is reduced to a minimum. Once in the power down mode, any or all of the inputs, including the clocks, may be toggled without the loss of data. To obtain the lowest possible power level, the inputs should be at static CMOS input levels during the power down period.

TIMING DIAGRAMS (Continued)

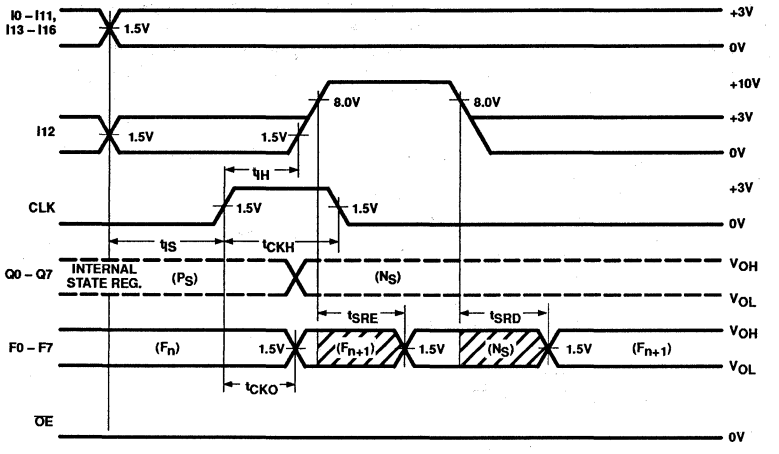


Power Down Enable and Disable

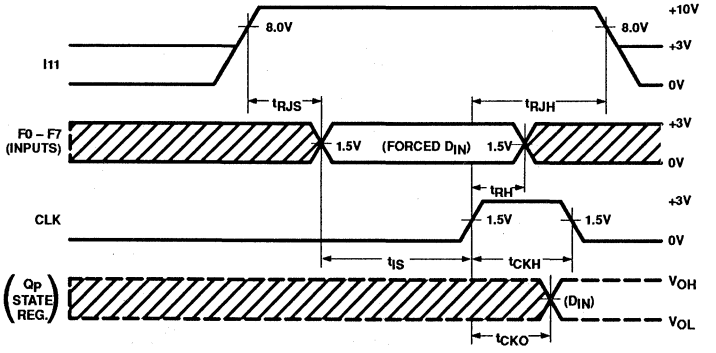


Power-On Preset

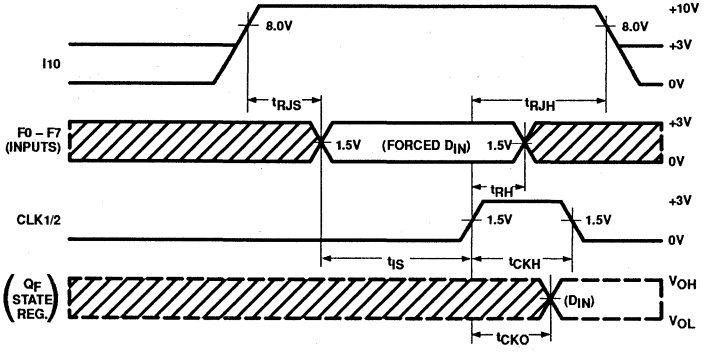
TIMING DIAGRAMS (Continued)



Diagnostic Mode—State Register Outputs



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—Output Register Input Jam

TIMING DEFINITIONS

SYMBOL	PARAMETER
f_{CLK}	Minimum guaranteed toggle frequency of the clock (from Clock HIGH to Clock HIGH).
$f_{MAX1,2}$	Minimum guaranteed operating frequency.
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{RP1}	Minimum guaranteed operating period – when not using Complement Array.
t_{RP2}	Minimum guaranteed operating period – when using Complement Array.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with outputs enabled).
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IHPD}	Required delay between the positive transition of the beginning of the first valid clock cycle to the beginning of Power Down LOW to insure that the last valid states are intact and that the next positive transition of the clock is valid.
t_{IHPU}	Required delay between the positive transition of the last valid clock and the beginning of Power Down HIGH to insure that last valid states are saved.
t_{INITH}	Width of initialization input pulse.
t_{INIT1}	Delay between positive transition of Initialization and when Outputs become valid when using external INIT control (from pin 19).
t_{INIT2}	Delay between positive transition of Initialization and when outputs become valid when using internal INIT control (from P-terms INA and INB).
t_{ISPD}	Required delay between the beginning of Power Down HIGH (from pin 19) and the positive transition of the next clock to insure that the clock edge is not detected as a valid Clock and that the last valid states are saved.

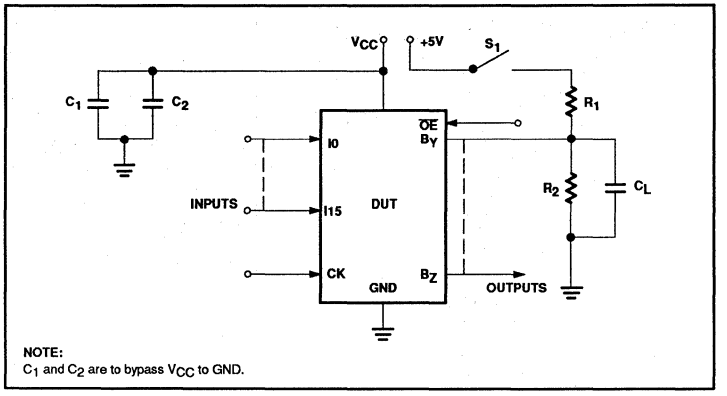
SYMBOL	PARAMETER
t_{ISPU}	Required delay between the beginning of Power Down LOW and the positive transition of the first valid clock.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{NVCK1}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization when using external INIT control (from pin 19) to guarantee that the clock edge is not detected as a valid negative transition.
t_{NVCK2}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization, when using the internal INIT control (from P-terms INA and INB), to guarantee that the clock edge is not detected as a valid negative transition.
t_{OD1}	Delay between beginning of Output Enable High and when Outputs are in the OFF-State, when using external OE control (from pin 19).
t_{OD2}	Delay between beginning of Output Enable High and when outputs are in the OFF-State when using internal OE control (from P-terms OEA and OEB).
t_{OE1}	Delay between beginning of Output Enable Low and when Outputs become valid when using external OE control from pin 19.
t_{OE2}	Delay between beginning of Output Enable Low and when outputs become valid when using internal OE control (from P-terms OEA and OEB).
t_{PDZ}	Delay between beginning of Power Down HIGH and when outputs are in OFF–State and the circuit is "powered down".

SYMBOL	PARAMETER
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
$t_{PUA1,2}$	Delay between beginning of Power Down LOW and when outputs become Active (valid) and the circuit is "powered up". See AC Specifications.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{RJH}	Required delay between positive transition of Clock and end of inputs I11 or I10 transition to State and Output Register Input Jam Diagnostic Modes, respectively.
t_{RJS}	Required delay between when inputs I11 or I10 transition to State and Output Register Input Jam Diagnostic Modes, respectively, and when the output pins become available as inputs.
t_{SRD}	Delay between input I12 transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VCK1}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding the first valid clock pulse when using external INIT control (pin 19).
t_{VCK2}	Required delay between the negative transition of the Asynchronous Initialization and the negative transition of the clock preceding the first valid clock pulse when using internal INIT control (from P–terms INA and INB).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.

CMOS programmable logic sequencer (17 × 68 × 8)

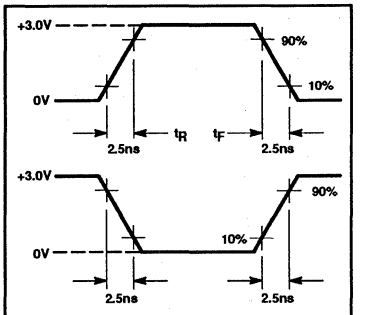
PLC415-16

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

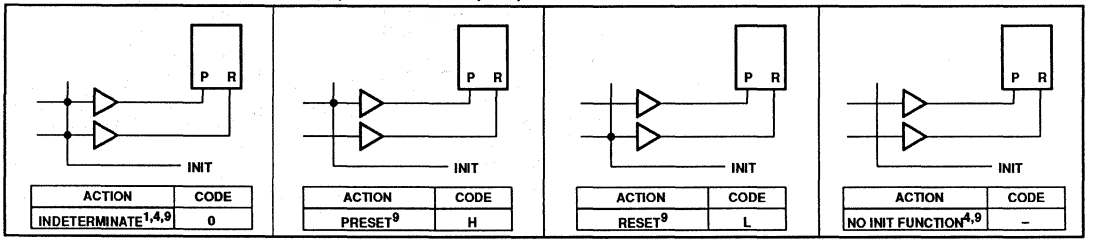
The PLC416-16 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLC416-16 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLC416-16 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations if assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION (PRESET/RESET)[11] OPTION – (P/R)



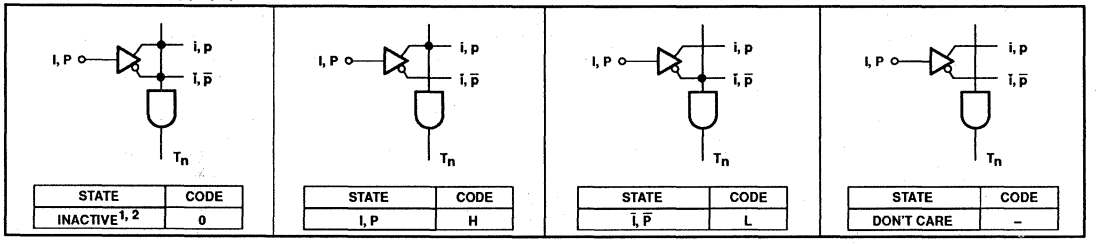
ACTION	CODE
INDETERMINATE[1,4,9]	0

ACTION	CODE
PRESET[9]	H

ACTION	CODE
RESET[9]	L

ACTION	CODE
NO INIT FUNCTION[4,9]	–

"AND" ARRAY – (I), (P)



STATE	CODE
INACTIVE[1, 2]	0

STATE	CODE
I, P	H

STATE	CODE
ī, p̄	L

STATE	CODE
DON'T CARE	–

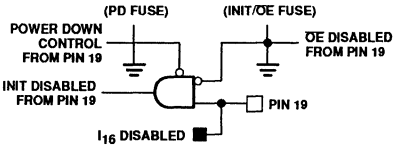
Notes are on page 344.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

LOGIC PROGRAMMING (Continued)

PIN 19 FUNCTION: POWER DOWN, INIITALIZATION, $\overline{OE}$, OR INPUT

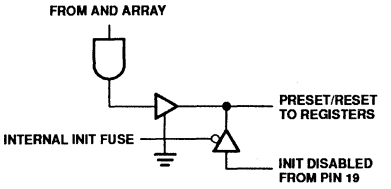
Power Down Mode



POWER DOWN FUSE	CODE
PIN 19 AS POWER DOWN	H[6]

EXTERNAL INIT/$\overline{OE}$ FUSE	CODE
EXTERNAL INIT/$\overline{OE}$ DISABLED	L

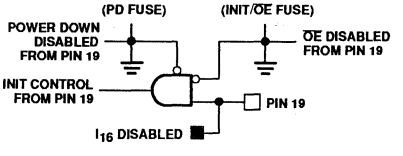
P-Term Initialization Control



INTERNAL INIT FUSES	CODE
P-TERM INIT CONTROL	H[7, 8]

POWER DOWN FUSE	CODE
POWER DOWN ENABLED OR DISABLED	H OR L

External Initialization Control

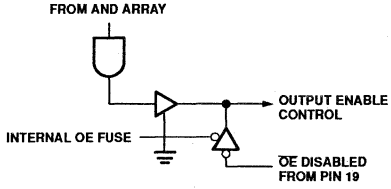


PD FUSE	CODE
POWER DOWN DISABLED	L[1]

EXTERNAL INIT/$\overline{OE}$ FUSE	CODE
PIN 19 AS EXTERNAL INIT	L[1]

INTERNAL INIT FUSES	CODE
P-TERM INIT ACTIVE OR INACTIVE	H OR L[7, 8]

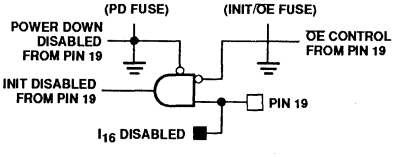
P-Term OE Control



INTERNAL OE FUSES	CODE
P-TERM OE CONTROL	H[7, 8]

POWER DOWN FUSE	CODE
POWER DOWN ENABLED OR DISABLED	H OR L

External Output Enable Control



PD FUSE	CODE
POWER DOWN DISABLED	L

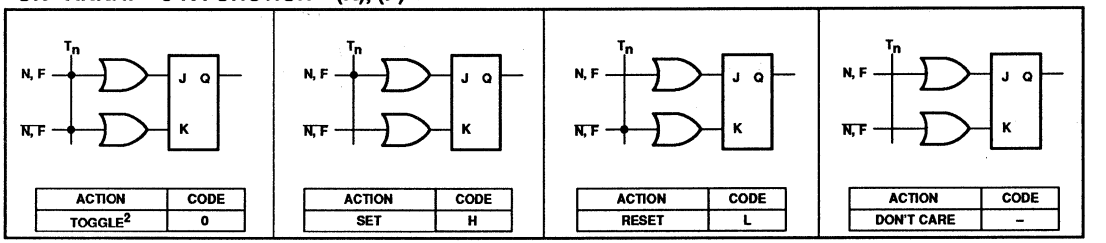
EXTERNAL INIT/$\overline{OE}$ FUSE	CODE
PIN 19 AS EXTERNAL $\overline{OE}$	H

INTERNAL INIT FUSES	CODE
P-TERM $\overline{OE}$ ACTIVE OR INACTIVE	H OR L[7, 8]

Notes are on page 344.

LOGIC PROGRAMMING (Continued)

"OR" ARRAY – J-K FUNCTION – (N), (F)



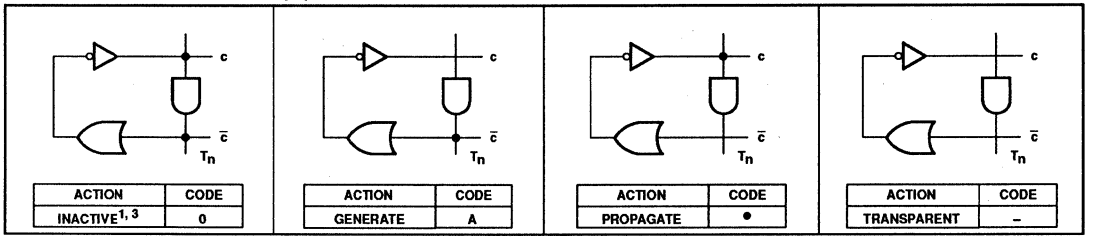
ACTION	CODE
TOGGLE[2]	0

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
DON'T CARE	–

"COMPLEMENT" ARRAY – (C)



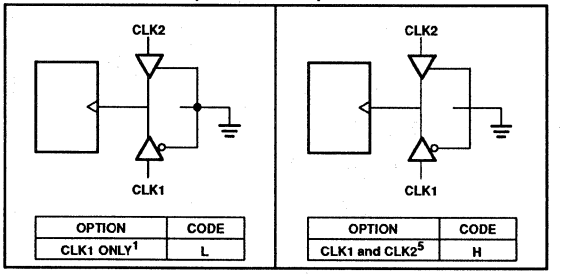
ACTION	CODE
INACTIVE[1, 3]	0

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L

OPTION	CODE
CLK1 and CLK2[5]	H

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using PRESET/RESET option.
5. Input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. When using Power Down feature, INPUT 16 is automatically disabled via the design software.
7. If the internal (P-term) control fuse for INIT and/or OE is programmed as Active High, the associated External Control function will be permanently disabled, regardless of the state of the External INIT/OE fuse.
8. One internal control fuse exists for each group of 8 registers. P0 – 3 and F0 – 3 are banked together in one group, as are P4 – 7 and F4 – 7. Control can be split between the INIT/OE pin (Pin 19) and P–terms INA, INB, OEA and OEB.
9. The PLC416-16 also has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.
10. L = cell unprogrammed.
    H = cell programmed.
11. Inputs 10, 11 and 12 (pins 25, 24, & 23) can be used for supervoltage diagnostic mode tests. It is recommended that these inputs not be connected to product terms INA, INB, OEA or OEB if you intend to make use of the diagnostic modes due to the fact that the patterns associated with the internal INIT and OE control product terms may interfere with the diagnostic mode data loading and reading.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

PROGRAM TABLE

AND

(Im, Ps)		(Cn)	
INACTIVE	O	INACTIVE	O
I, P	H	GENERATE	A
$\overline{I}$, $\overline{P}$	L	PROPAGATE	•
DON'T CARE	–	TRANSPARENT	–

OR (Ns, Fn)

TOGGLE	O
SET	H
RESET	L
NO CHANGE	–

OPTIONS

ENTERNAL INIT	L
EXTERNAL OE	H
INTERNAL INIT/OE ENABLED	H
INTERNAL INIT/OE DISABLED	L

POWER DOWN ENABLED	H
POWER DOWN DISABLED	L
CLOCK 1 ONLY	L
CLOCK 1 AND 2	H

INITIALIZATION	
PRESET	H
RESET	L
NO INIT	–
INDETERMINATE	O

PROJECT NAME _____ DATE _____ REVISION _____

CUSTOMER NAME _____ SIGNETICS PART # _____ CUSTOMER PART # _____

TERMS	COMP. ARRAY	INTERNAL INIT INB	INTERNAL INIT INA	INTERNAL OE OEB	INTERNAL OE OEA	EXTERNAL INIT/OE	CLK 1/1&2	PD	INITIALIZATION
	NOR	AND							OR
	Cn	INPUT (Im)					PRESENT STATE (Ps)		NEXT STATE (Ns) / OUTPUT (Fn)
	C1 C0	I16 I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0					P7 P6 P5 P4 P3 P2 P1 P0		N7 N6 N5 N4 N3 N2 N1 N0 / F7 F6 F5 F4 F3 F2 F1 F0
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									
56									
57									
58									
59									
60									
61									
62									
63									
INB									
OEB									
INA									
OEA									
PIN NO.		19 20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9							10 11 12 13 15 16 17 18
PIN LABELS									

NOTES:

1. In the unprogrammed state all cells are conducting. Thus, the program table for an unprogrammed device would contain "0"s for all product terms (inactive) and initialization states (indeterminate). The default or unprogrammed state of all other options is "L".
2. Unused Cn, Im and Ps cells are normally programmed as Don't Care (–).
3. Unused product terms can be left blank (inactive) for future code modification.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PLC415 Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps has wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PLC415 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PLC415 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

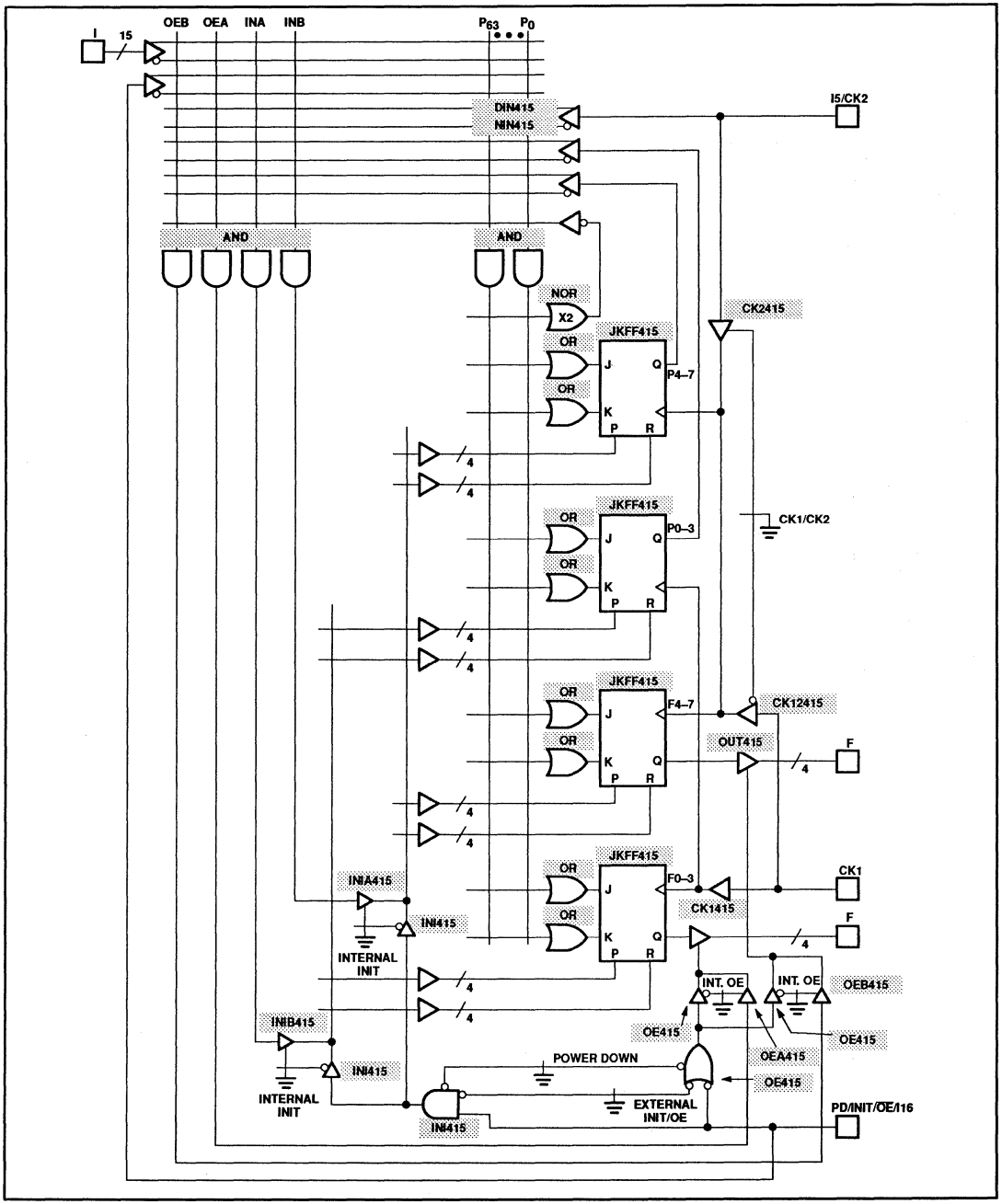
The recommended erasure procedure for the PLC415 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000μW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12000μW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

CMOS programmable logic sequencer (17 × 68 × 8)

PLC415-16

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencers (16 × 48 × 8)

PLS105/A

DESCRIPTION

The PLS105and the PLS105A are bipolar Programmable Logic State machines of the Mealy type. They contain logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 6 Q_P, and 8 Q_F edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option. all flip-flops are unconditionally preset to "1" during power turn on.

The AND array combines 16 external inputs $I_0 - I_{15}$ with six internal inputs P_{0-5}, which are fed back from the State Registers to form up to 48 transition terms (AND terms). All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse. Both True and Complement transition terms can be generated by optional use of the internal input variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to $\overline{\text{Output Enable}}$ function, as an additional user-programmable option.

Order codes are listed below in the Ordering Information Table.

FEATURES

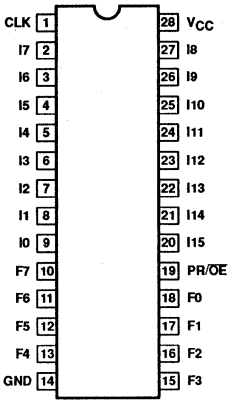
- PLS105
  - f_{MAX} = 13.9MHz
  - 20MHz clock rate
- PLS105A
  - f_{MAX} = 20MHz
  - 25MHz clock rate
- Field-Programmable (Ni-Cr link)
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked flip-flops
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to all "1" of internal registers
- Power dissipation: 600mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

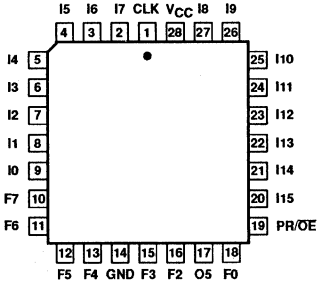
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

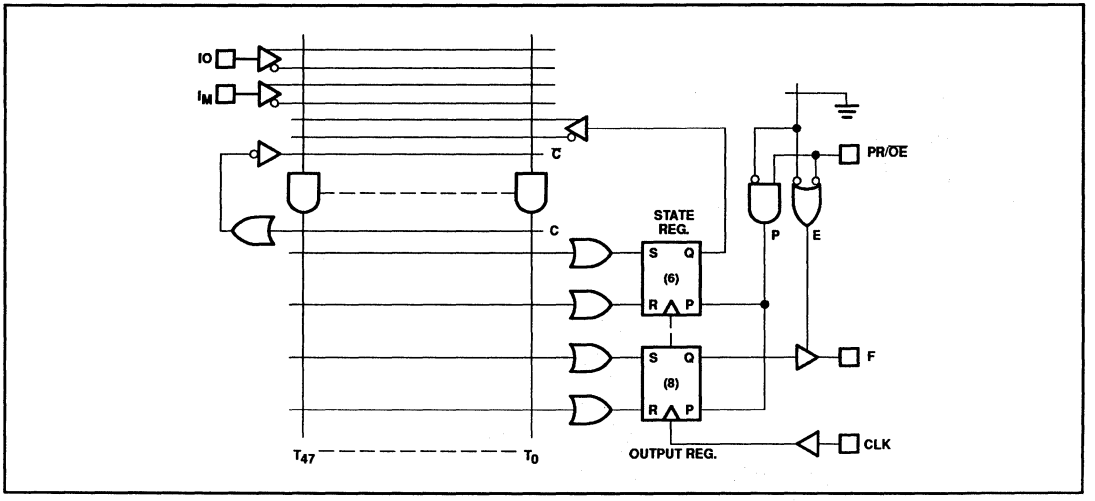
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
28-Pin Plastic DIP (600mil-wide)	PLS105N, PLS105AN	0413B
28-Pin Plastic Leaded Chip Carrier	PLS105A, PLS105AA	0401F

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

FUNCTIONAL DIAGRAM



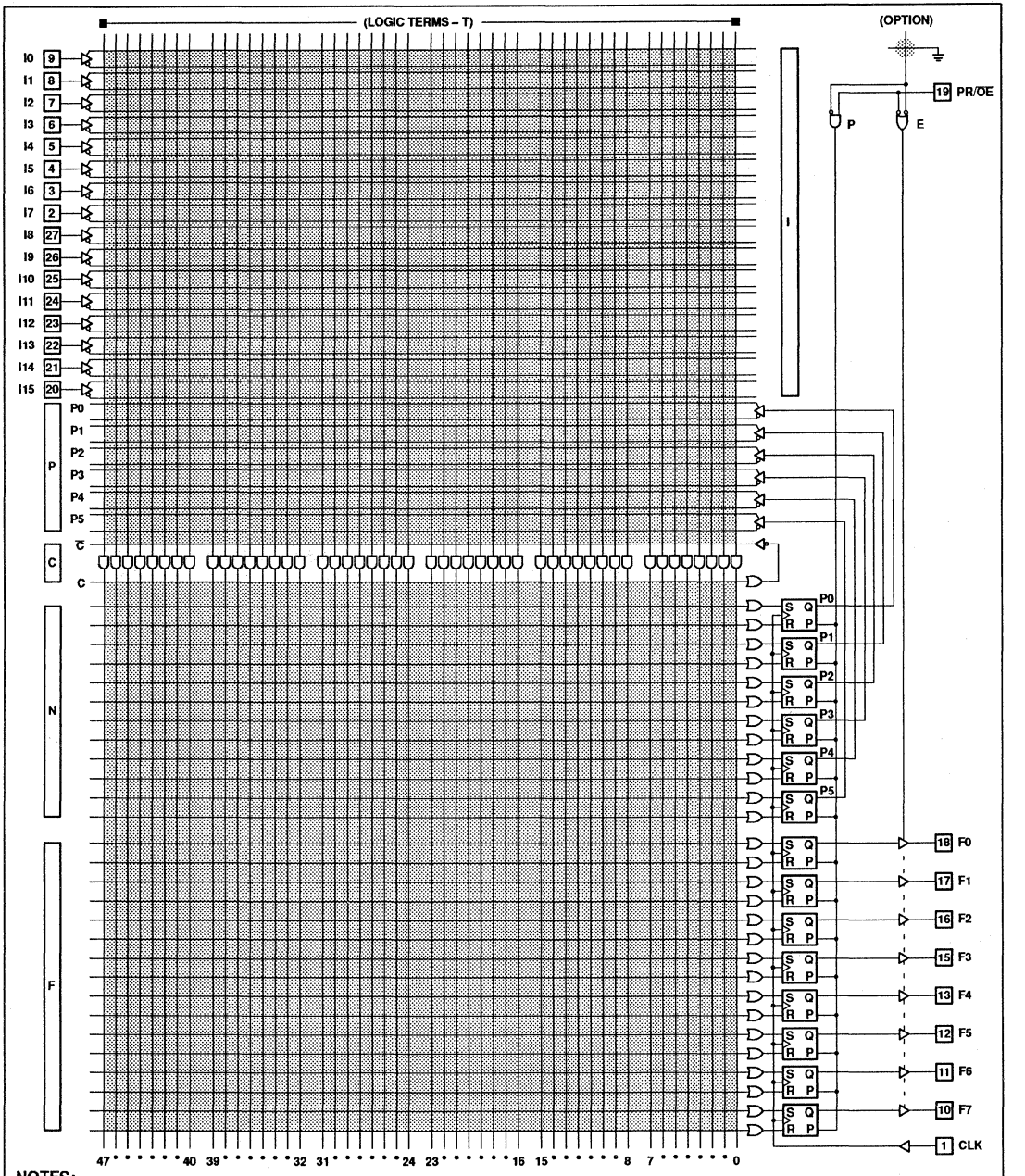
PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2 – 8 20 – 27	I1 – I15	**Logic Inputs:** The 15 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
9	I0	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercised with standard TTL levels. When I0 is held at +10V, device outputs F0 – 5 reflect the contents of State Register bits P0 – 5. The contents each Output Register remains unaltered.	Active-High/Low
10 – 13 15 – 18	F0 – 7	**Logic/Diagnostic Outputs:** Eight device outputs which normally reflect the contents of Output Register bits Q0 – 7, when enabled. When I0 is held at +10V, F0 – 5 = (P0 – 5), and F6, 7 = Logic "1".	Active-High
19	PR/$\overline{OE}$	**Preset or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Preset:** Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and F0 – 7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an Output Enable function to all output buffers F0 – 7 from the Output Register.	Active-Low (L)

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

LOGIC DIAGRAM



NOTES:

1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ▒ Programmable connection.

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

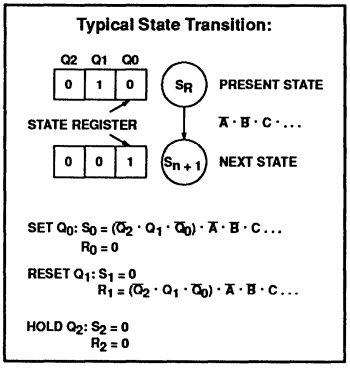
TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I_0	CK	S	R	$Q_{P/F}$	F
+5V	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q_n	$(Q_P)_n$
	L		X	X	X	X	Q_n	$(Q_F)_n$
		H	*	X	X	X	Q_n	Hi–Z
		L	+10V	X	X	X	Q_n	$(Q_P)_n$
		L	X	X	X	X	Q_n	$(Q_F)_n$
		L	X	↑	L	L	Q_n	$(Q_F)_n$
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

NOTES:
1. Positive Logic:
   $S/R = T_0 + T_1 + T_2 + \ldots + T_{47}$
   $T_n = C(I0\ I1\ I2 \ldots)\ (P0\ P1 \ldots P5)$
2. Either Preset (Active–High) or $\overline{\text{Output Enable}}$ (Active–Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. R = S = High is an illegal input condition.
5. * = H or L or +10V.
6. X = Don't Care (≤5.5V).

LOGIC FUNCTION

Typical State Transition:



VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
2. All transition terms are disabled (0).
3. All S/R flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array pre-programmed with a standard test pattern.
   NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Philips Semiconductors qualified programming equipment.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
V_{OH}	High[4]	V_{CC} = MIN I_{OH} = −2mA	2.4			V
V_{OL}	Low[5]	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{IN} = 5.5V		<1	25	µA
I_{IL}	Low	V_{IN} = 0.45V		−10	−100	µA
I_{IL}	Low (CK input)	V_{IN} = 0.45V		−50	−250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[6]	V_{CC} = MAX V_{OUT} = 5.5V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit [3, 7]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		120	180	mA
Capacitance[6]						
C_{IN}	Input	V_{CC} = 5.0V V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Measured with V_{IL} applied to $\overline{OE}$ and a logic high stored, or with V_{IH} applied to PR.
5. Measured with a programmed logic condition for which the output is at a low logic level, and V_{IL} applied to PR/$\overline{OE}$ Output sink current is supplied through a resistor to V_{CC}.
6. Measured with V_{IH} applied to PR/$\overline{OE}$.
7. Duration of short circuit should not exceed 1 second.
8. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

AC ELECTRICAL CHARACTERISTICS

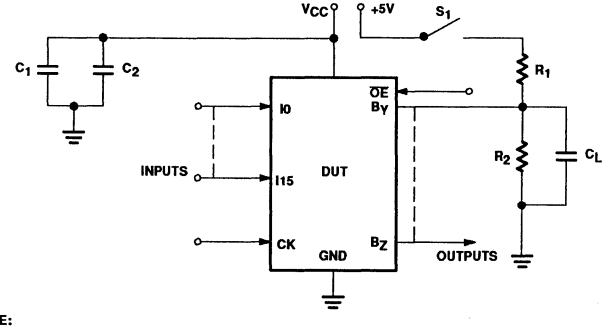
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLS105			PLS105A			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width										
t_{CKH}	Clock[2] High	CK +	CK –	25	15		20	15		ns
t_{CKL}	Clock Low	CK –	CK +	25	15		20	15		ns
t_{CKP}	Clock period	CK +	CK +	50	30		40	30		ns
t_{PRH}	Preset pulse	PR +	PR –	25	15		25	15		ns
Setup time[3]										
$t_{IS1}A$	Input	Input ±	CK +	60			40			ns
$t_{IS1}B$	Input	Input ±	CK +	50			30			ns
$t_{IS1}C$	Input	Input ±	CK +	42			N/A			ns
$t_{IS2}A$	Input (through Complement Array)	Input ±	CK +	90			70			ns
$t_{IS2}B$	Input (through Complement Array)	Input	CK +	80			60			ns
$t_{IS2}C$	Input (through Complement Array)	Input	CK +	72			N/A			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		0	–10		ns
t_{PRS}	Preset	PR –	CK –	0	–10		0	–10		ns
Hold time										
t_{IH}	Input	CK +	Input ±	5	–10		5	–10		ns
Propagation delay										
t_{CKO}	Clock	CK +	Output ±		15	30		15	20	ns
t_{OE}	Output enable[4]	OE –	Output –		20	30		20	30	ns
t_{OD}	Output disable[4]	OE +	Output +		20	30		20	30	ns
t_{PR}	Preset	PR +	Output +		18	30		18	30	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10		0	10	ns
Frequency of operation[3]										
$f_{MAX}C$	Without Complement Array			13.9			20.0			MHz
$f_{MAX}C$	With Complement Array			9.8			12.5			MHz

NOTES:

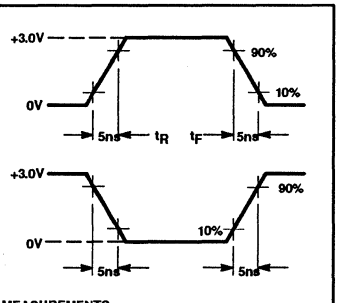
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. To prevent spurious clocking, clock rise time (10% – 90%) ≤ 30ns.
3. See "Speed vs. OR Loading" diagrams.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

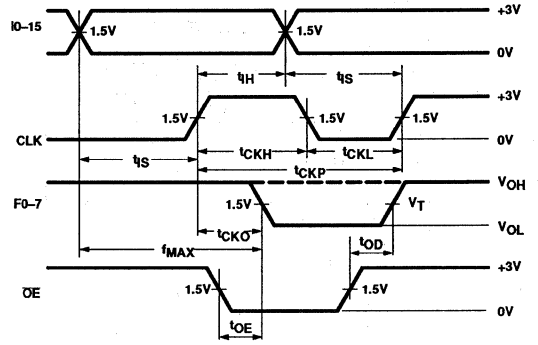
VOLTAGE WAVEFORMS



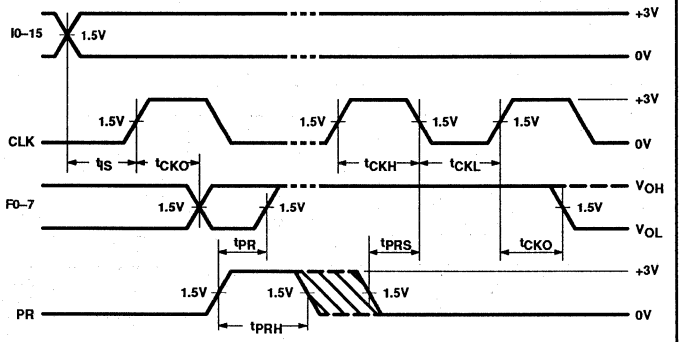
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

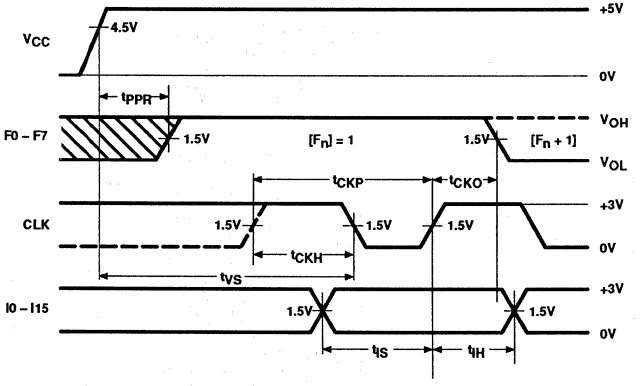
TIMING DIAGRAMS



Sequential Mode



Asynchronous Preset

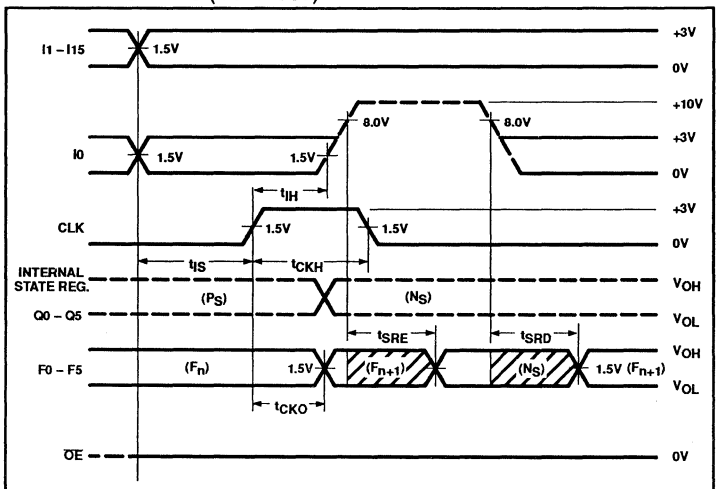


Power-On Preset

TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{CKH}	Width of input clock pulse.
t_{CKL}	Interval between clock pulses.
t_{CKP}	Minimum guaranteed Clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of clock.
t_{IS2}	Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND array).
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and negative transition of Clock preceding first reliable clock pulse.
t_{IH}	Required delay between positive transition of Clock and end of valid input data.
t_{CKO}	Delay between positive transition of clock and when outputs become valid (with PR/$\overline{OE}$ Low).
t_{OE}	Delay between beginning of Output Enable Low and when outputs become valid.
t_{OD}	Delay between beginning of Output Enable High and when outputs are in the OFF-State.
t_{SRE}	Delay between input I_0 transition to Diagnostic mode and when the outputs reflect the contents of the State Register.
t_{SRD}	Delay between input I_0 transition to Logic mode and when the outputs reflect the contents of the Output Register.
t_{PR}	Delay between positive transition of Preset and when outputs become valid at "1".
t_{PPR}	Delay between V_{CC} (after power-on) and when outputs become preset at "1".
t_{PRH}	Width of preset input pulse.
f_{MAX}	Minimum guaranteed operating frequency.

TIMING DIAGRAMS (Continued)



Diagnostic Mode

SPEED VS. "OR" LOADING

The maximum frequency at which the PLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms T_n used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects t_{IS}, due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of t_{IS1} with the number of terms connected per OR.

The PLS105 AC electrical characteristics contain three limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 1). The first, t_{IS1A} is guaranteed for a device with 48 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 32 terms connected to any OR line. And t_{IS1C} is guranteed for a device with 24 terms conntected to any OR line.

The three other entries in the AC table, t_{IS2} A, B, and C are corresponding 48, 32, and 24 term limits when using the on–chip Complement Array.

The PLS105A AC electrical characteristics contain two limits for the parameters t_{IS1} and t_{IS2} (refer to Figure 2). The first, t_{IS1A} is guaranteed for a device with 24 terms connected to any OR line. t_{IS1B} is guaranteed for a device with 16 terms connected to any OR line.

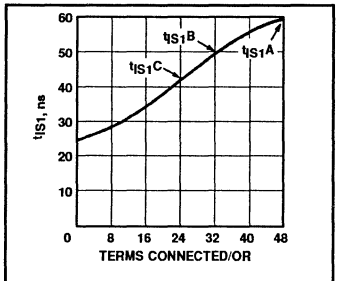


Figure 1. PLS105 t_{IS1} vs. Terms/OR Connected

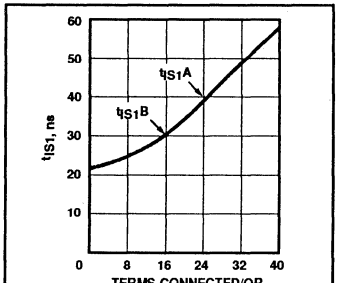


Figure 2. PLS105A t_{IS1} vs. Terms/OR Connected

The two other entries in the AC table, t_{IS2} A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of t_{IS} for a given application can be determined by identifying the OR line with the maximum number of T_n connections. This can be done by referring to the interconnect pattern in the PLS logic diagram, typically illustrated in Figure 3, or by counting the maximum number of "H" *or* "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 or 2 will yield the worst case t_{IS} and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

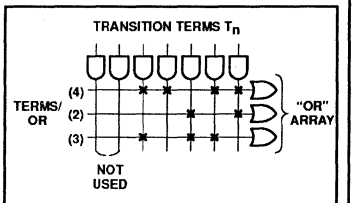


Figure 3. Typical OR Array Interconnect Pattern

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

LOGIC PROGRAMMING

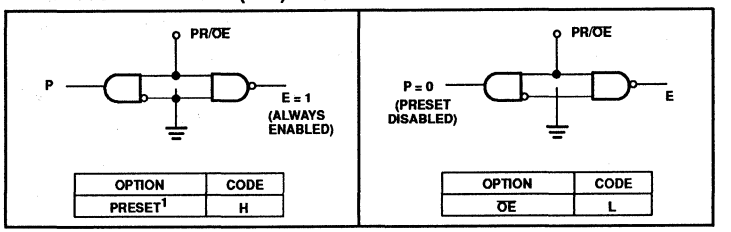
The PLS105/A devices are fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP, Data I/O Corporation's ABEL™ and Logical Devices Inc.'s CUPL™ design software packages.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLS105/A logic designs can also be generated using the program table entry format detailed on the following pages. This program table entry format is supported by the Philips Semiconductors SNAP PLD design software package.

To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

PRESET/$\overline{OE}$ OPTION – (P/E)



OPTION	CODE
PRESET[1]	H

OPTION	CODE
$\overline{OE}$	L

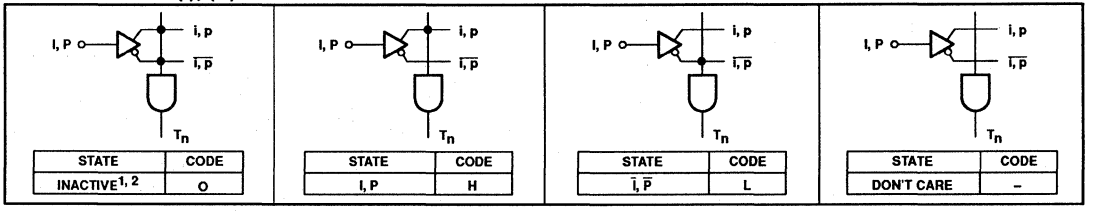
PROGRAMMING:

The PLS105/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

"AND" ARRAY – (I), (P)



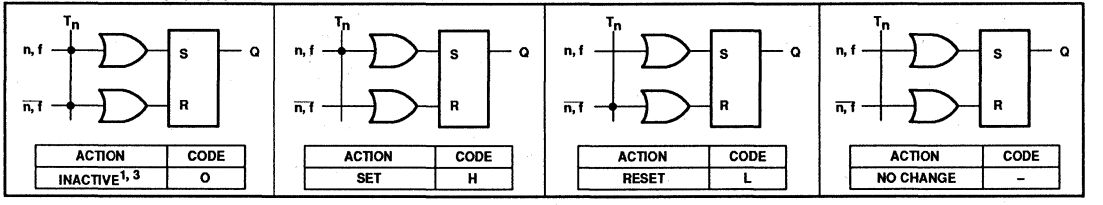
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



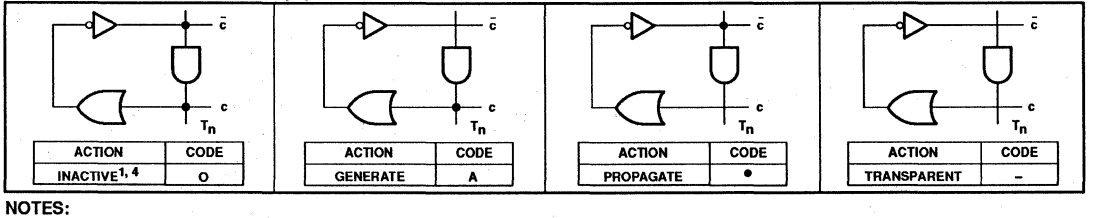
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I or P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

PROGRAM TABLE

CUSTOMER NAME _____

PURCHASE ORDER # _____

PHILIPS DEVICE # CF (XXXX)

CUSTOMER SYMBOLIZED PART # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE _____

REV _____

DATE _____

PROGRAM TABLE ENTRIES

AND

INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

C_n

INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

I_m, P_s

OR

INACTIVE	0
SET	H
RESET	L
NO CHANGE	—

Ns, Fr

OPTION

PRESET	H
$\overline{OE}$	L

P/E

TERM	C_n	AND: INPUT (I_m) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (Ns) 5 4 3 2 1 0	OUTPUT (Fr) 7 6 5 4 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9				10 11 12 13 15 16 17 18
VARIABLE NAME						

NOTES:

1. The FPLS is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencers (16 × 48 × 8)

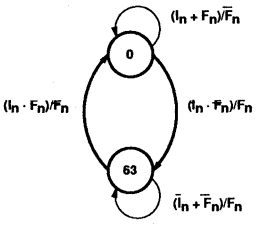
PLS105/A

TEST ARRAY

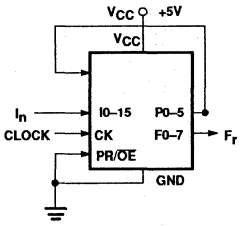
The PLS105/A may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the PLS105/A and applying the proper input sequence to I0 – I15 as shown in the test circuit timing diagram.



State Diagram

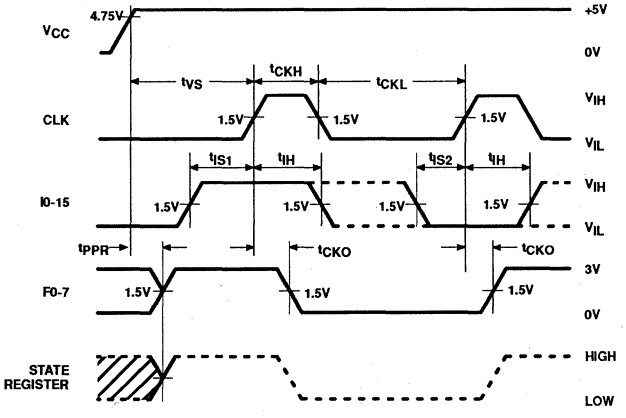


PLS Under Test

TERM	C	AND: INPUT (Im) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (Ps) 5 4 3 2 1 0	OPTION (P/E) H; OR: NEXT STATE (Ns) 5 4 3 2 1 0	OR: OUTPUT (Fr) 7 6 5 4 3 2 1 0
48	A	H H H H H H H H H H H H H H H H	H H H H H H	L L L L L L	L L L L L L L L
49	•	L L L L L L L L L L L L L L L L	L L L L L L	H H H H H H	H H H H H H H H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any Philips Semiconductors' qualified programming equipment.



Test Circuit Timing Diagram

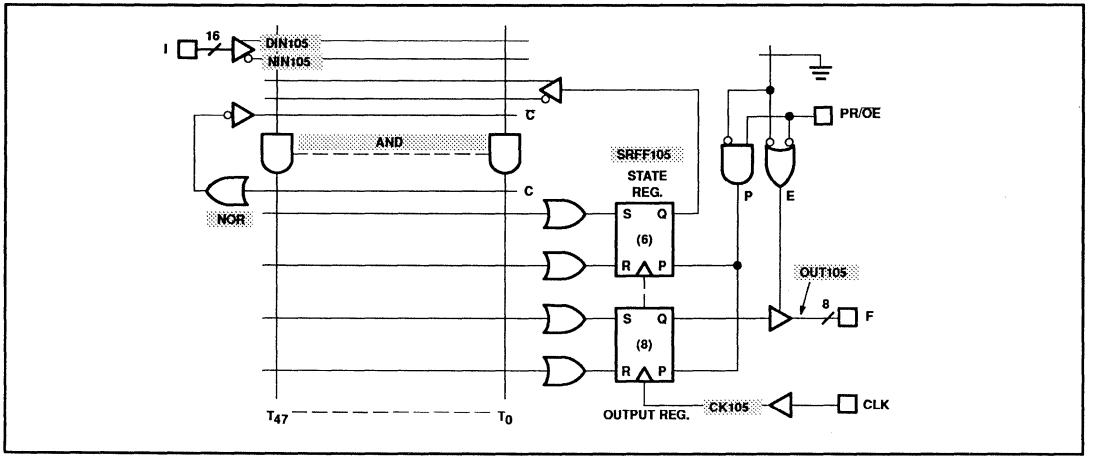
TERM	C	AND: INPUT (Im) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (Ps) 5 4 3 2 1 0	OPTION (P/E) H; OR: NEXT STATE (Ns) 5 4 3 2 1 0	OR: OUTPUT (Fr) 7 6 5 4 3 2 1 0
48	–	H H H H H H H H H H H H H H H H	H H H H H H	– – – – – –	– – – – – – – –
49	•	L L L L L L L L L L L L L L L L	L L L L L L	– – – – – –	– – – – – – – –

Test Array Deleted

Programmable logic sequencers (16 × 48 × 8)

PLS105/A

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

DESCRIPTION

The PLUS105-45 is a bipolar programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 48 AND gates are connected to the 16 external dedicated inputs (I0-I15) and to the feedback paths of the 6 buried State Registers (Q_{P0}-Q_{P5}). Because the OR array is programmable, any one or all of the 48 transition terms can be connected to any or all of the State and Output Registers.

All state transition terms can include True, False and Don't Care states of the controlling state variables. A Complement Transition Array supports complex IF-THEN-ELSE state transitions with a single product term.

The PLUS105-45 device features edge-triggered, J-K flip-flops, which provide the added flexibility of the toggle function which is indeterminate on S-R flip-flops. Because the J-K function is a superset of the S-R flip-flop function, the PLUS105-45 is backward compatible with all 105-type devices that have S-R flip-flops. Asynchronous Preset/Output Enable functions are available.

The PLUS105-45 is pin-for-pin and software compatible with the Philips Semiconductors PLS105 and PLS105A Logic Sequencers, as well as other commercially available 105-type programmable logic devices.

To facilitate testing of state machine designs, diagnostic mode features for register preset and buried state register observability have been incorporated into the PLUS105-45 device architecture.

Ordering codes are listed in the Ordering Information Table.

FEATURES

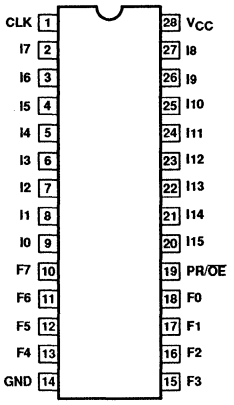
- 45MHz operating frequency
  - 55.6MHz clock rate
  - No OR term loading restrictions
- Available in 300mil skinny DIP, 600mil-wide Plastic DIP and PLCC packages
- Pin and software compatible with other commercially available 105 logic sequencers
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked J-K (or S-R) flip-flops
- Security fuse
- Programmable Asynchronous Preset or Output Enable
- Power-on preset (to all "1"s) of internal registers
- Power dissipation: 800mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security Locking systems
- Counters
- Shift registers

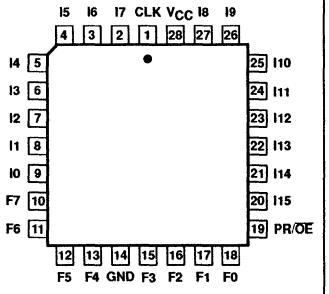
PIN CONFIGURATIONS

N Packages



N = Plastic DIP (600mil-wide)
N3 = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

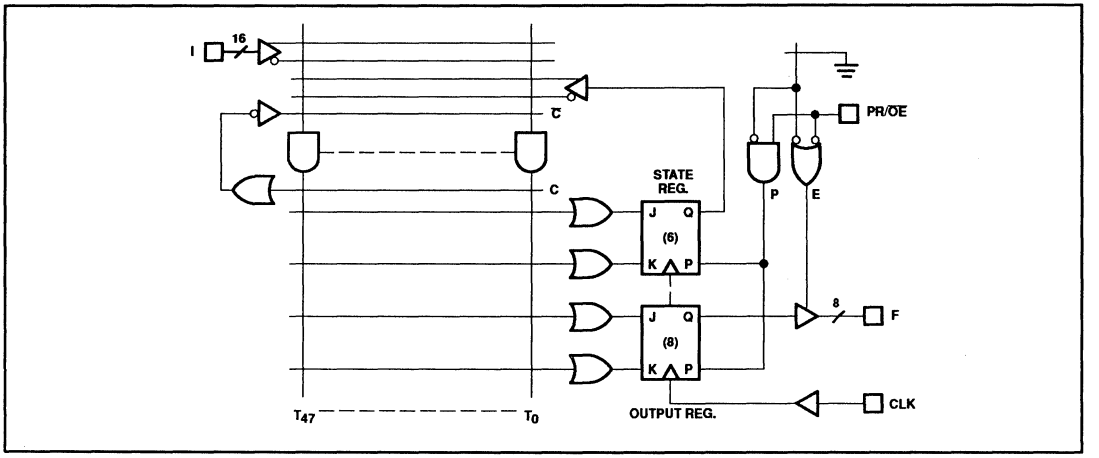
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
28-pin Plastic Dual-In-Line, 600mil-wide	PLUS105–45N	0413B
28-pin Plastic Dual-In-Line, 300mil-wide	PLUS105–45N3	0864D
28-pin Plastic Leaded Chip Carrier, 450mil-square	PLUS105–45A	0401F

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

FUNCTIONAL DIAGRAM



PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both state and output registers.	Active-High (H)
2-9, 26, 27 20-22	I0 - I9, I13 - I15	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/ Low (H/L)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When I12 is held at +10V, device outputs F0 - F5 reflect the contents of State Register bits P_0 - P_5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0 - F5 become direct inputs for State Register bits P_0 - P_5; a Low-to-High transition on the clock line loads the values on pins F0 - F5 into the State Register bits P_0 - P_5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I10 is held at +10V, device outputs F0 - F7 become direct inputs for Output Register bits Q0 - Q7; a Low-to-High transition on the clock line loads the values on pins F0 - F7 into the Output Register bits Q0 - Q7. The contents of each State Register remains unaltered.	Active-High/ Low (H/L)
10-13 15-18	F0 - F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register bits Q0 - Q7, when enabled. When I12 is held at +10V, F0 - F5 = (P_0 - P_5). When I11 is held at +10V, F0 - F5 become inputs to State Register bits P_0 - P_5. When I10 is held at +10V, F0 - F7 become inputs to Output Register bits Q0 - Q7.	Active-High (H)
19	PR/$\overline{OE}$	**Preset or Output Enable Input:** **A user programmable function:**	
		• **Preset:** Provides an asynchronous preset to logic "1" of all State and Output Register bits. PR overrides Clock, and when held High, clocking is inhibited and F0 - F7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the Preset signal goes low. See timing definitions.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F0 - F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

TRUTH TABLE [1, 2, 3, 4, 5, 6]

V_{CC}	OPTION										
	PR	$\overline{OE}$	I10	I11	I12	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H	H	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

1. Positive Logic:
   J-K (or S/R) = $T_0 + T_1 + T_2 + \ldots T_{47}$
   $T_n = (C_0)\ (I0, I1, I2, \ldots)\ (P_0, P_1, \ldots P_5)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤ 5.5V)
6. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

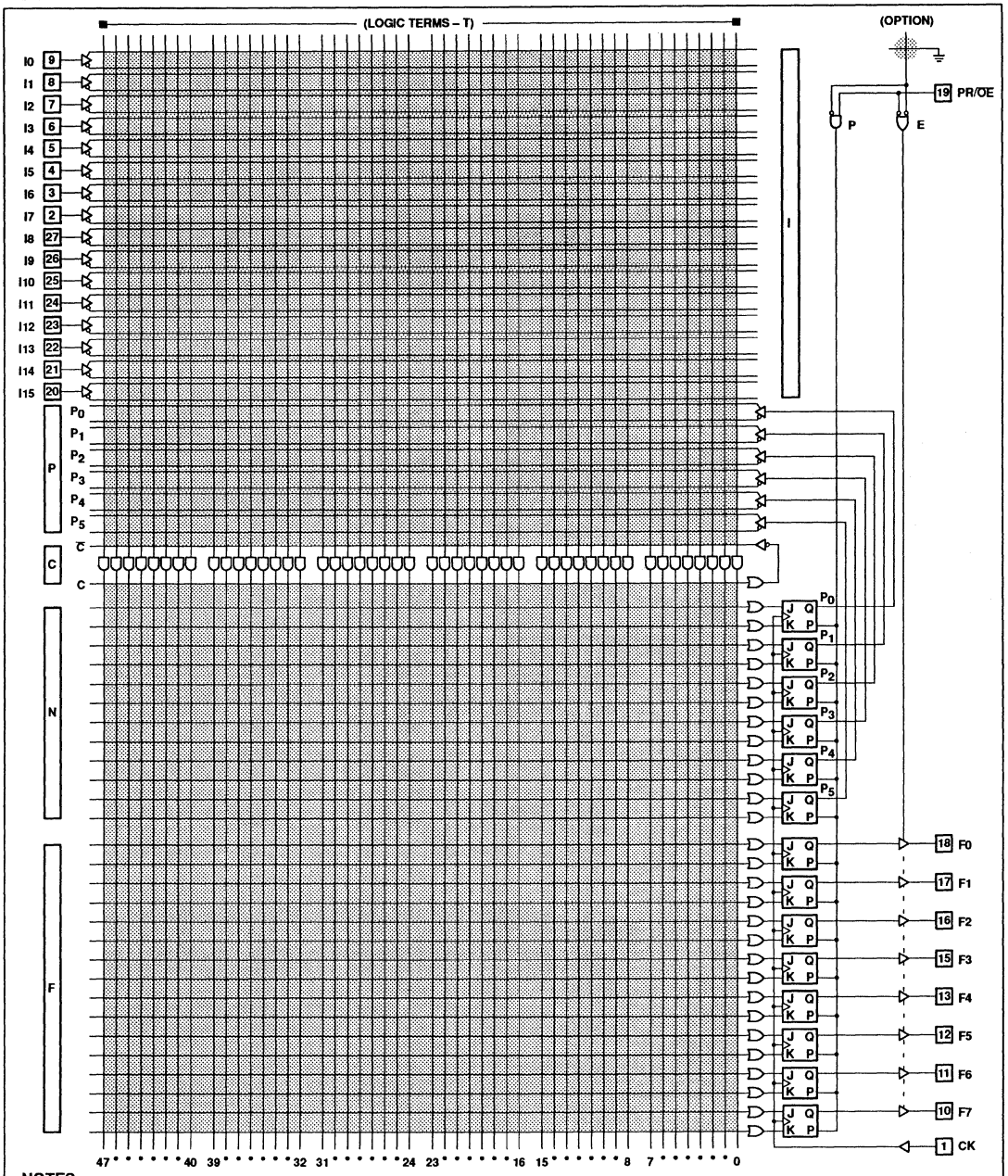
A factory-shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Note that even if the PR function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are disabled (0).
3. All J-K flip-flop inputs are disabled (0).

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

LOGIC DIAGRAM



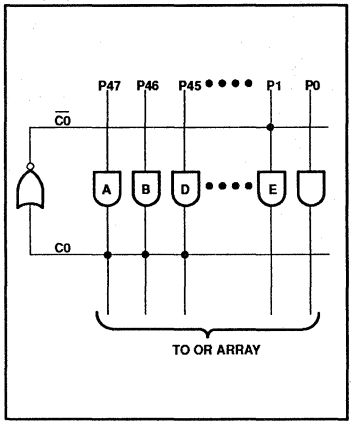
NOTES:

1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ▒ Programmable connection.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

COMPLEMENT ARRAY DETAIL



The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to a known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that use of the Complement Array adds an additional delay path through the device. Refer to the AC Electrical Characteristics for details.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

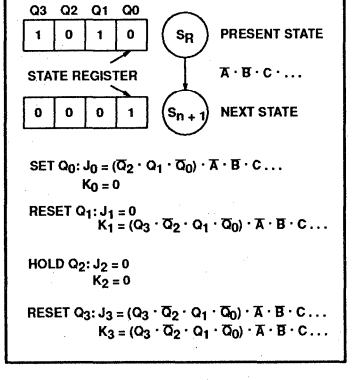
ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS MIN	RATINGS MAX	UNIT
V_{CC}	Supply voltage		+7.0	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq 75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = V_{CC}		<1	30	µA
I_{IL}	Low	V_{IN} = 0.45V		−20	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 2.7V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		160	200	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

Programmable logic sequencer (16 × 48 × 8) — PLUS105-45

AC ELECTRICAL CHARACTERISTICS

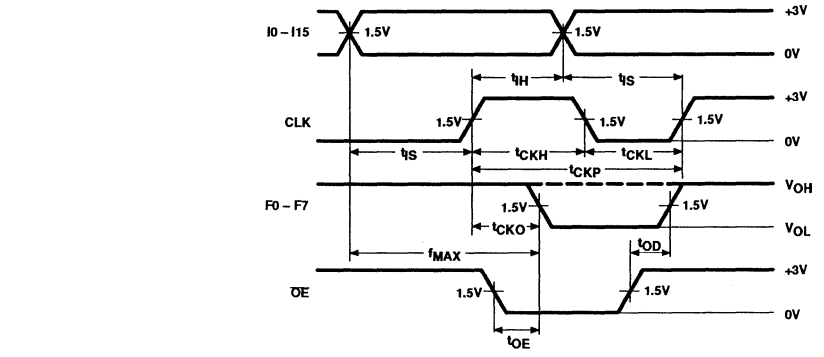
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30PF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse Width							
t_{CKH}	Clock High	CK +	CK –	9	8		ns
t_{CKL}	Clock Low	CK –	CK +	9	8		ns
t_{CKP}	Clock Period	CK +	CK +	18	16		ns
t_{PRH}	Preset pulse	PR +	PR -	10	8		ns
Setup Time							
t_{IS1}	Input	Input ±	CK+	13	12		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK +	23	20		ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	–10		ns
t_{PRS}	Clock resume (after preset)	PR –	CK –	0	–5		ns
t_{NVCK}	Clock lockout (before preset)	CK –	PR –	10	5		ns
Hold Time							
t_{IH}	Input	CK +	Input ±	0	–5		ns
Diagnostic Mode							
t_{RJS}	Initialization of diagnostic mode	I10, I11 or I12+ (to 10V)	F_n as inputs	50	25		ns
t_{RJH}	Clock for diagnostic mode	CK +	Register input jam	50	25		ns
Propagation Delay[3]							
t_{CKO}	Clock	CK +	Output ±		8	9	ns
t_{OE}	Output enable[2]	OE –	Output –		8	9	ns
t_{OD}	Output disable[2]	OE +	Output +		8	9	ns
t_{PR}	Preset	PR +	Output +		12	15	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		0	10	ns
Frequency of Operation							
f_{MAX1}	Without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO}}\right)$	Input ±	Output ±	45.5	50.0		MHz
f_{MAX2}	With Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO}}\right)$	Input thru Complement Array ±	Output ±	31.3	35.7		MHz
f_{MAX3}	Internal feedback without Complement Array $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	55.6	62.5		MHz
f_{MAX4}	Internal feedback with Complement Array $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Input ±	43.5	50.0		MHz
f_{CLK}	Clock frequency	CK +	CK +	55.6	62.5		MHz

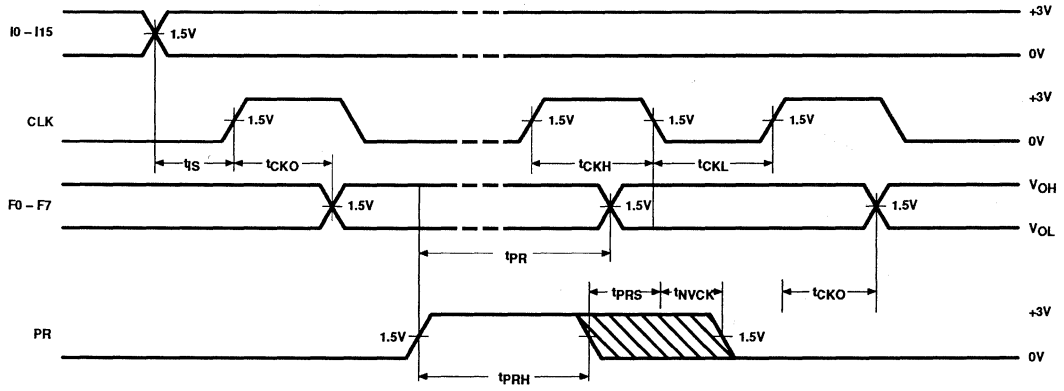
NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} – 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times are meausred and specified under worst case conditions.

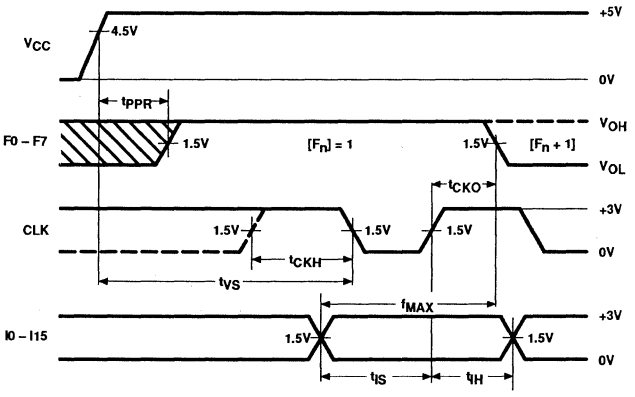
TIMING DIAGRAMS



Sequential Mode

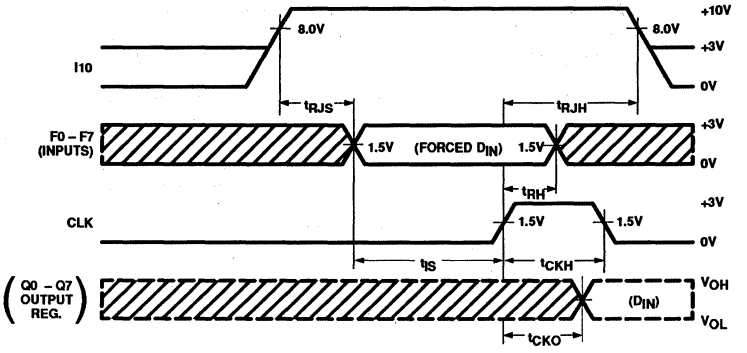


Asynchronous Preset

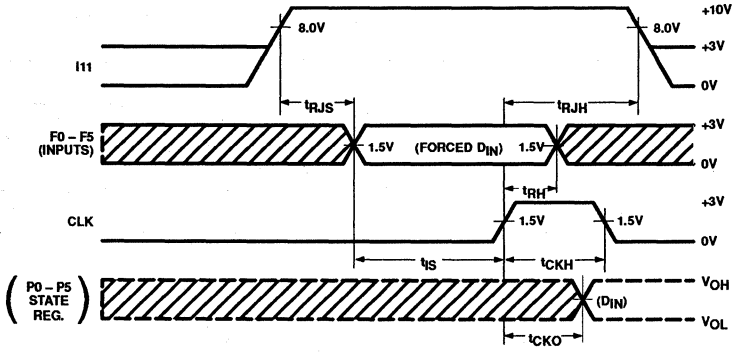


Power-On Preset

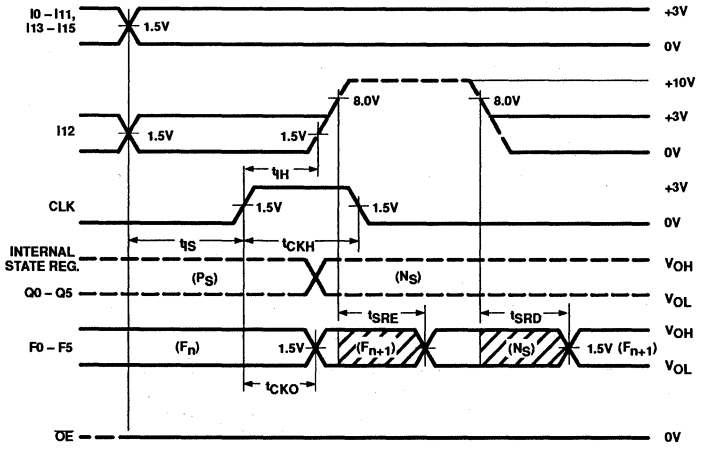
TIMING DIAGRAMS (Continued)



Diagnostic Mode—Output Register Input Jam



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—State Register Outputs

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

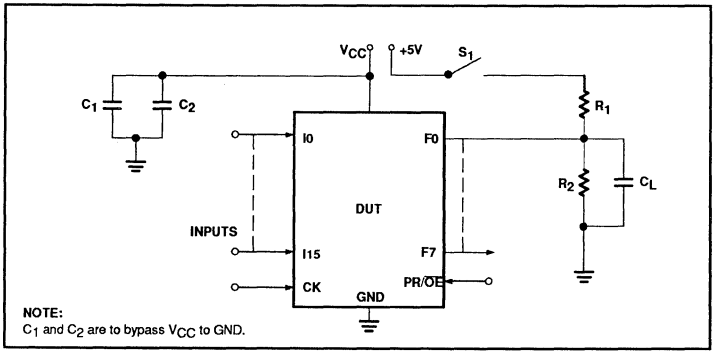
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{CKH}	Width of input clock pulse
t_{CKL}	Interval between clock pulses.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with PR/$\overline{OE}$ Low).
t_{CKP}	Minimum guaranteed clock period.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous PRESET to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{PR}	Delay between positive transition of Preset and when Outputs become valid at "1".
t_{PRH}	Width of preset input pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and the first positive transition of Clock.
t_{RH}	Required delay between positive transition of clock and the end of valid input data (F0–F7 as inputs), when jamming data into the State or Output registers in the Diagnostic Mode.

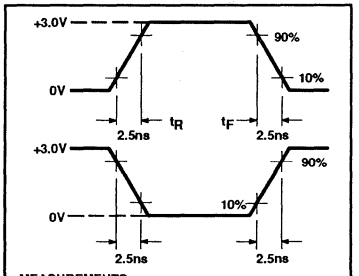
SYMBOL	PARAMETER
t_{RJH}	Required delay between positive transition of clock and return of input I10, I11 OR I12 from Diagnostic Mode (10V).
t_{RJS}	Required delay between inputs I10, I11 or I12 transition to Diagnostic Mode (10V), and when the output pins become available as inputs.
t_{SRD}	Delay between input (I12) transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency)
f_{MAX}	Minimum guaranteed operating frequency.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

PLUS105-45 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SLICE and SNAP design software packages. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLUS105-45 architecture.

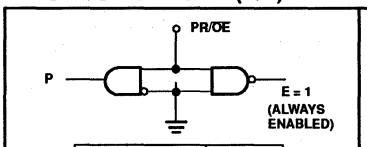
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS105-45 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SLICE only. The SLICE design package is available, free of charge, to qualified users.

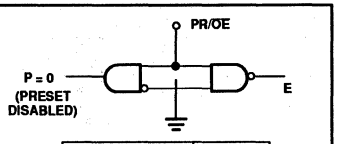
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

PRESET/OE OPTION – (P/E)



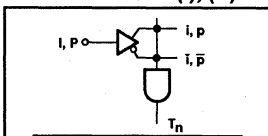
OPTION	CODE
PRESET[1]	H



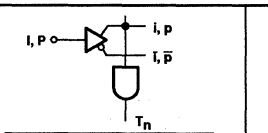
OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS105–45:
The PLUS105–45 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

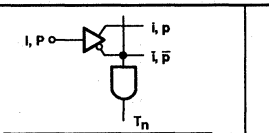
"AND" ARRAY – (I), (P)



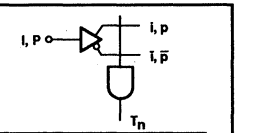
STATE	CODE
INACTIVE[1, 2]	0



STATE	CODE
I, P	H

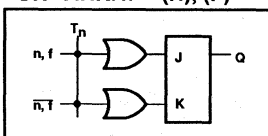


STATE	CODE
$\overline{I}$, $\overline{P}$	L

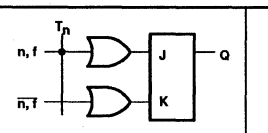


STATE	CODE
DON'T CARE	–

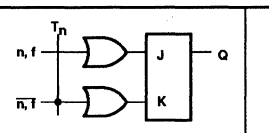
"OR" ARRAY – (N), (F)



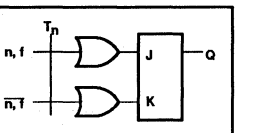
ACTION	CODE
TOGGLE[1]	0



ACTION	CODE
SET	H

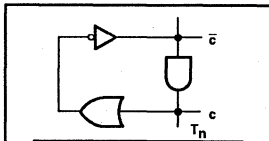


ACTION	CODE
RESET	L

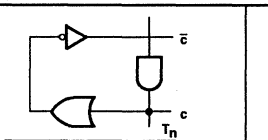


ACTION	CODE
NO CHANGE	–

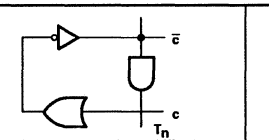
"COMPLEMENT" ARRAY – (C)



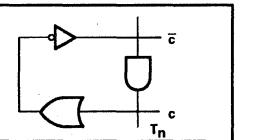
ACTION	CODE
INACTIVE[1, 3]	0



ACTION	CODE
GENERATE	A



ACTION	CODE
PROPAGATE	•



ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all link pairs.
2. Any gate T_n will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

PLUS105 PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
PHILIPS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

C_n

INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

I_m, P_s

OR

INACTIVE or TOGGLE	0
SET	H
RESET	L
NO CHANGE	—

Ns, Fr

OPTION

PRESET	H
$\overline{OE}$	L

P/E

TERM	C_n	AND: INPUT (I_m) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (Ns) 5 4 3 2 1 0	OUTPUT (Fr) 7 6 5 4 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9				10 11 12 13 15 16 17 18
VARIABLE NAME						

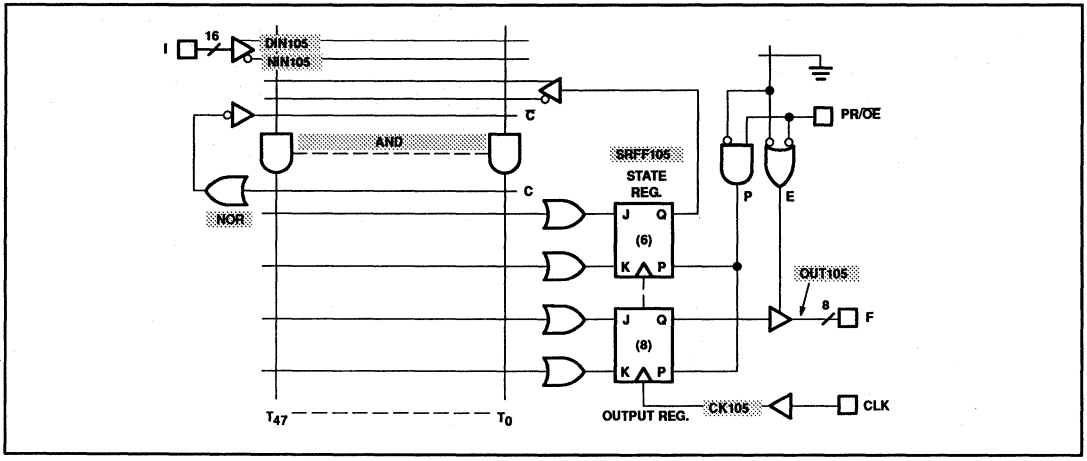
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-45

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

DESCRIPTION

The PLUS105-55 is a bipolar programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 48 AND gates are connected to the 16 external dedicated inputs (I0 - I15) and to the feedback paths of the 6 buried State Registers (Q_{P0}-Q_{P5}). Because the OR array is programmable, any one or all of the 48 transition terms can be connected to any or all of the State and Output Registers.

All state transition terms can include True, False and Don't Care states of the controlling state variables. A Complement Transition Array supports complex IF-THEN-ELSE state transitions with a single product term.

The PLUS105-55 device features edge-triggered, J-K flip-flops, which provide the added flexibility of the toggle function which is indeterminate on S-R flip-flops. Because the J-K function is a superset of the S-R flip-flop function, the PLUS105-55 is backward compatible with all 105-type devices that have S-R flip-flops. Asynchronous Preset/Output Enable functions are available.

The PLUS105-55 is pin-for-pin and software compatible with Philips Semiconductors PLS105 and PLS105A Logic Sequencers, as well as other commercially available 105-type programmable logic devices.

To facilitate testing of state machine designs, diagnostic mode features for register preset and buried state register observability have been incorporated into the PLUS105-55 device architecture.

Ordering codes are listed in the Ordering Information Table.

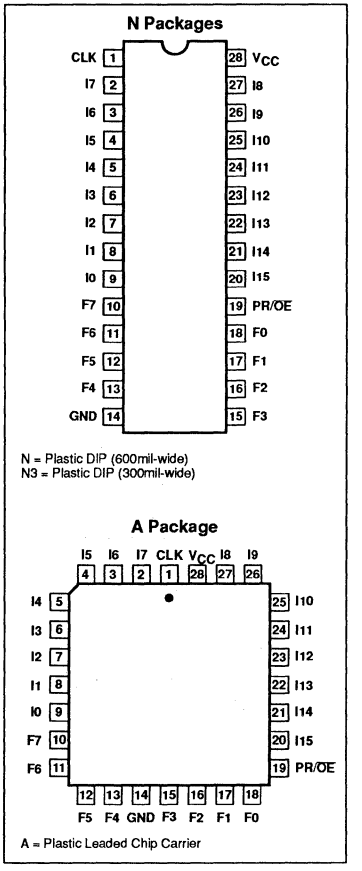
FEATURES

- 55MHz operating frequency
  - 71.4MHz clock rate
  - No OR term loading restrictions
- Available in 300mil skinny DIP, 600mil-wide DIP, and PLCC packages
- Pin and software compatible with other commerically available 105 sequencers
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked J-K (or S-R) flip-flops
- Security fuse
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to (all "1"s) of internal registers
- Power dissipation: 800mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security Locking systems
- Counters
- Shift registers

PIN CONFIGURATIONS



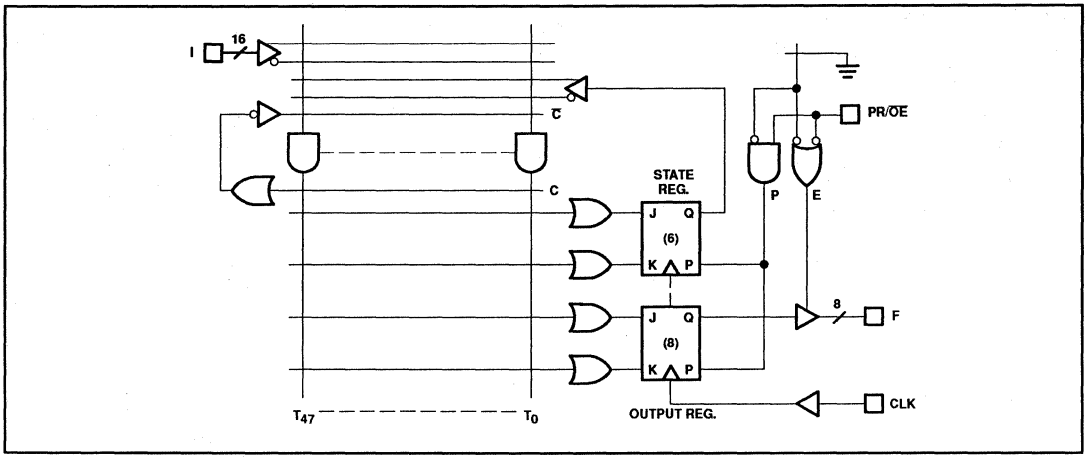
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
28-pin Plastic Dual-In-Line, 600mil-wide	PLUS105-55N	0413B
28-pin Plastic Dual-In-Line, 300mil-wide	PLUS105-55N3	0864D
28-pin Plastic Leaded Chip Carrier, 450mil-square	PLUS105-55A	0401F

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

FUNCTIONAL DIAGRAM



PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both state and output registers.	Active-High (H)
2-9, 26, 27 20-22	I0 - I9, I13 - I15	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/ Low (H/L)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When I12 is held at +10V, device outputs F0 - F5 reflect the contents of State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0 - F5 become direct inputs for State Register bits P0 - P5; a Low-to-High transition on the clock line loads the values on pins F0 - F5 into the State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I10 is held at +10V, device outputs F0 - F7 become direct inputs for Output Register bits Q0 - Q7; a Low-to-High transition on the clock line loads the values on pins F0 - F7 into the Output Register bits Q0 - Q7. The contents of each State Register remains unaltered.	Active-High/ Low (H/L)
10-13 15-18	F0 - F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register bits Q0 - Q7, when enabled. When I12 is held at +10V, F0 - F5 = (P0 - P5). When I11 is held at +10V, F0 - F5 become inputs to State Register bits P0 - P5. When I10 is held at +10V, F0 - F7 become inputs to Output Register bits Q0 - Q7.	Active-High (H)
19	PR/$\overline{OE}$	**Preset or Output Enable Input:** **A user programmable function:**	
		• **Preset:** Provides an asynchronous preset to logic "1" of all State and Output Register bits. PR overrides Clock, and when held High, clocking is inhibited and F0 - F7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the Preset signal goes Low. See timing definitions.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F0 - F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

TRUTH TABLE[1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I10	I11	I12	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H	H	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

1. Positive Logic:
   J-K (or S/R) = $T_0 + T_1 + T_2 + \ldots T_{48}$
   $T_n = (C_0)\ (I0, I1, I2, \ldots)\ (P_0, P_1, \ldots P_5)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤ 5.5V)
6. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

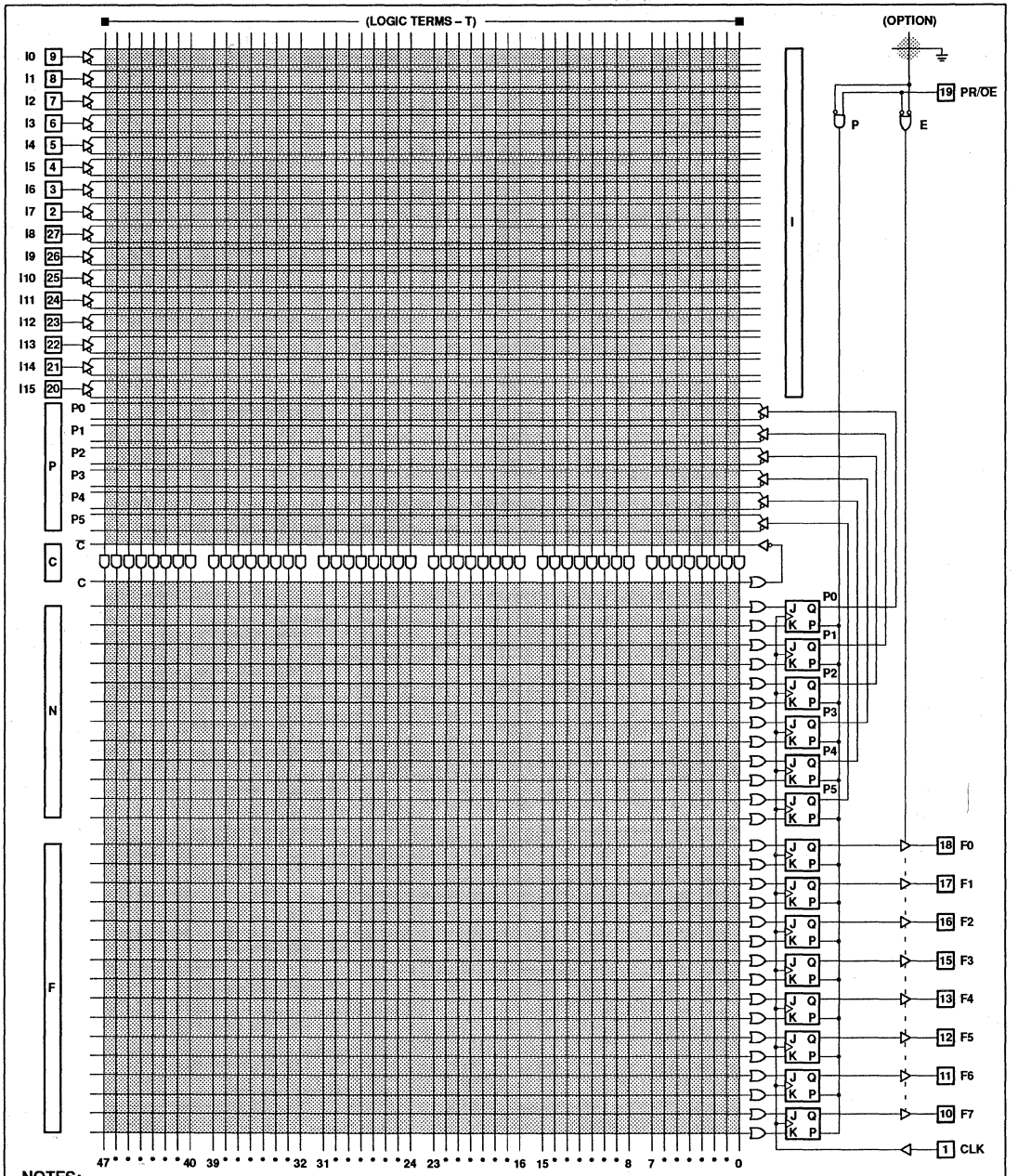
A factory-shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Note that even if the PR function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are disabled (0).
3. All J-K flip-flop inputs are disabled (0).

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

LOGIC DIAGRAM



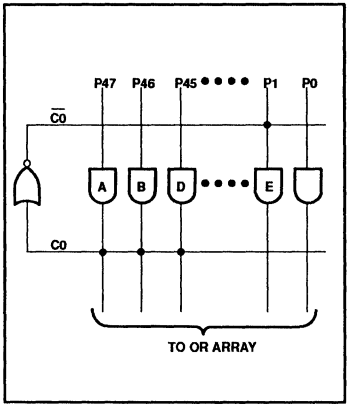
NOTES:

1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. ▒ Programmable connection.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

COMPLEMENT ARRAY DETAIL



The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to a known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that use of the Complement Array adds an additional delay path through the device. Refer to the AC Electrical Characteristics for details.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

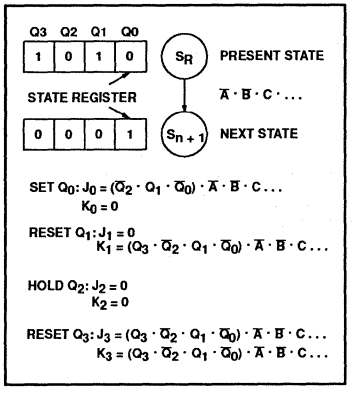
ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage		+7.0	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	-30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	-65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq 75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = –2mA	2.4			V
V_{OL}	Low	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = V_{CC}		<1	30	µA
I_{IL}	Low	V_{IN} = 0.45V		–20	–250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 2.7V		1	40	µA
		V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		160	200	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

AC ELECTRICAL CHARACTERISTICS

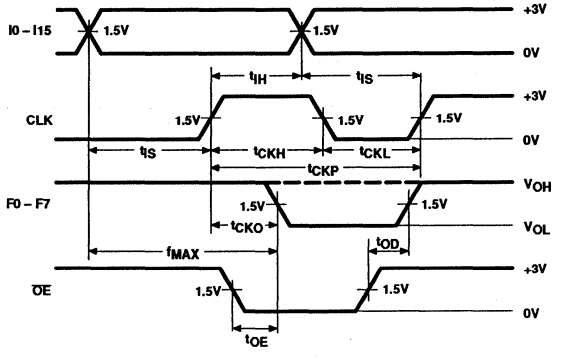
$R_1 = 470\Omega$, $R_2 = 1K\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse Width							
t_{CKH}	Clock High	CK +	CK –	7	6.5		ns
t_{CKL}	Clock Low	CK –	CK +	7	6.5		ns
t_{CKP}	Clock Period	CK +	CK +	14	13		ns
t_{PRH}	Preset pulse	PR +	PR –	10	8.0		ns
Setup Time							
t_{IS1}	Input	Input ±	CK +	10	9.5		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK +	20	18.0		ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0	0		ns
t_{PRS}	Clock resume (after preset)	PR –	CK –	0	0		ns
t_{NVCK}	Clock lockout (before preset)	CK –	PR –	12	10.0		ns
Hold Time							
t_{IH}	Input	CK +	Input ±	0	–5		ns
Diagnostic Mode							
t_{RJS}	Initialization of diagnostic mode	I10, I11 or I12 + (to 10V)	F_n as inputs	50	25		ns
t_{RJH}	Clock for diagnostic mode	CK +	Register input jam	50	25		ns
Propagation Delay[2]							
t_{CKO}	Clock	CK +	Output ±		7	8	ns
t_{OE}	Output enable[3]	OE –	Output –		6	8	ns
t_{OD}	Output disable[3]	OE +	Output +		6	8	ns
t_{PR}	Preset	PR +	Output +		12	15	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +		5	10	ns
Frequency of Operation							
f_{MAX1}	Without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO}}\right)$	Input ±	Output ±	55.6	60.6		MHz
f_{MAX2}	With Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO}}\right)$	Input thru Complement Array ±	Output ±	35.7	40.0		MHz
f_{MAX3}	Internal feedback without Complement Array $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	71.4	76.9		MHz
f_{MAX4}	Internal feedback with Complement Array $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Input ±	50.0	55.6		MHz
f_{CLK}	Clock period	CK +	CK +	71.4	76.9		MHz

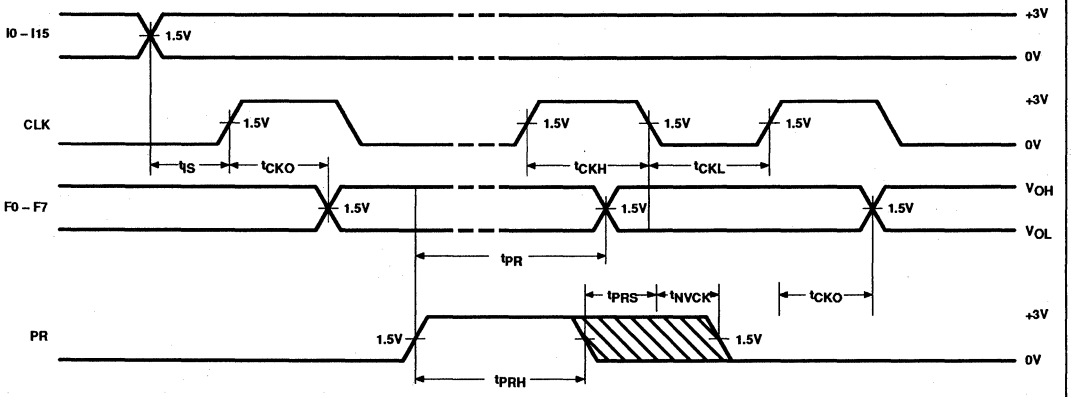
NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. All propagation delays and setup times are meausred and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

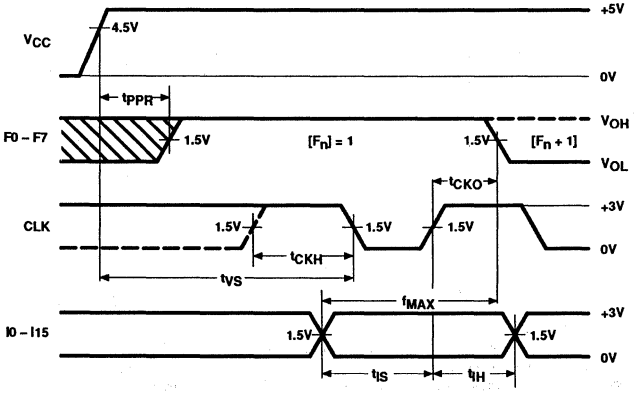
TIMING DIAGRAMS



Sequential Mode

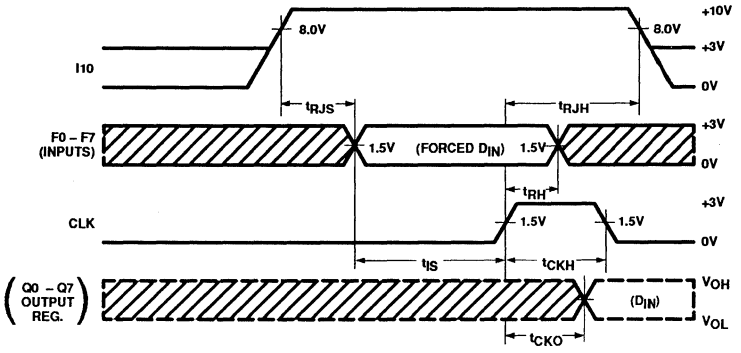


Asynchronous Preset

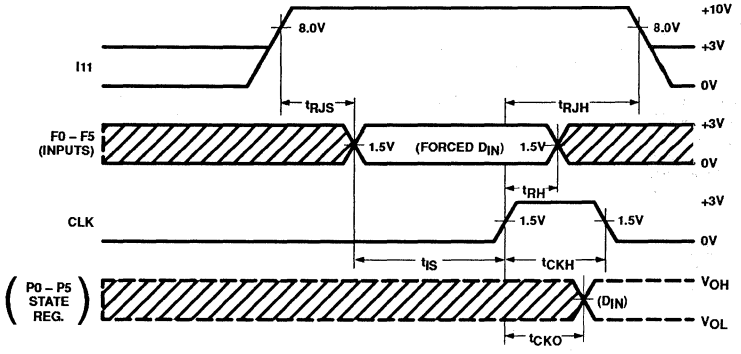


Power-On Preset

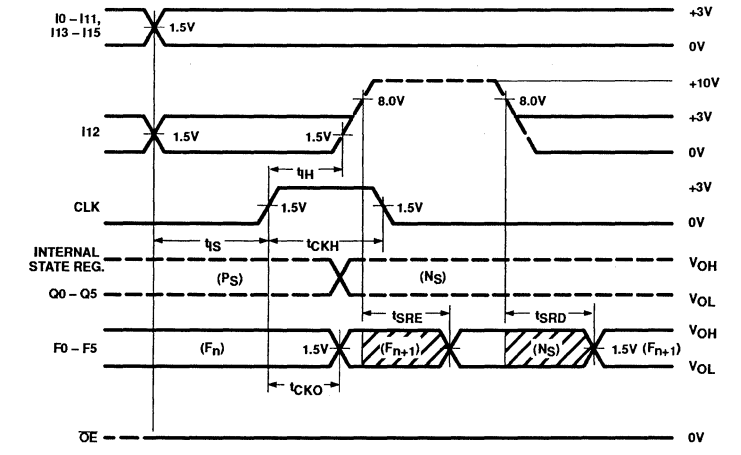
TIMING DIAGRAMS (Continued)



Diagnostic Mode—Output Register Input Jam



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—State Register Outputs

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

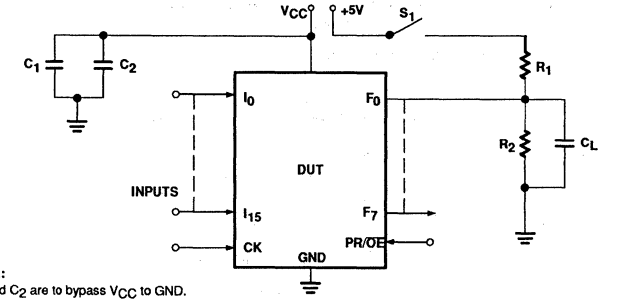
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{CKH}	Width of input clock pulse
t_{CKL}	Interval between clock pulses.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with PR/$\overline{OE}$ Low).
t_{CKP}	Minimum guaranteed clock period.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous PRESET to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{PR}	Delay between positive transition of Preset and when Outputs become valid at "1".
t_{PRH}	Width of preset input pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and the first positive transition of Clock.
t_{RH}	Required delay between positive transition of clock and the end of valid input data (F0 – F7 as inputs), when jamming data into the State or Output registers in the Diagnostic Mode.

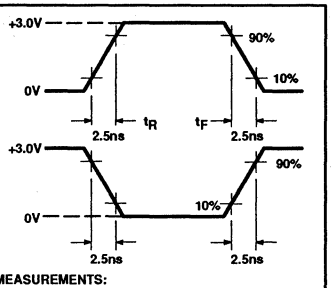
SYMBOL	PARAMETER
t_{RJH}	Required delay between positive transition of clock and return of input I10, I11 or I12 from Diagnostic Mode (10V).
t_{RJS}	Required delay between inputs I10, I11 or I12 transition to Diagnostic Mode (10V), and when the output pins become available as inputs.
t_{SRD}	Delay between input (I12) transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency)
f_{MAX}	Minimum guaranteed operating frequency.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

PLUS105-55 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLUS105-55 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS105-55 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP.

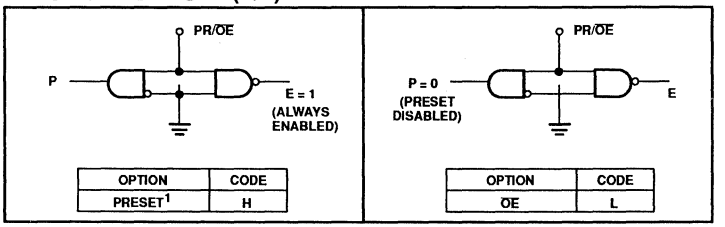
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

PRESET/OE OPTION - (P/E)

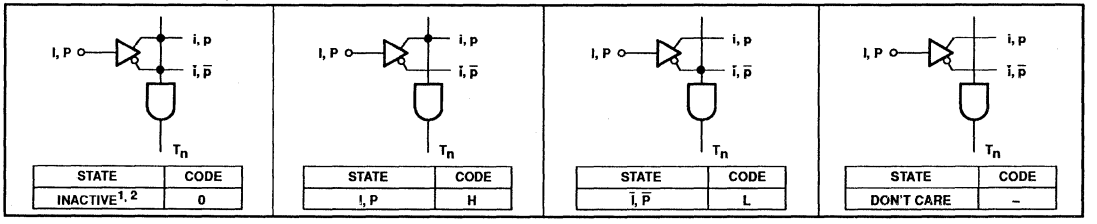


OPTION	CODE
PRESET[1]	H

OPTION	CODE
OE	L

PROGRAMMING THE PLUS105-55:
The PLUS105-55 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY - (I), (P)



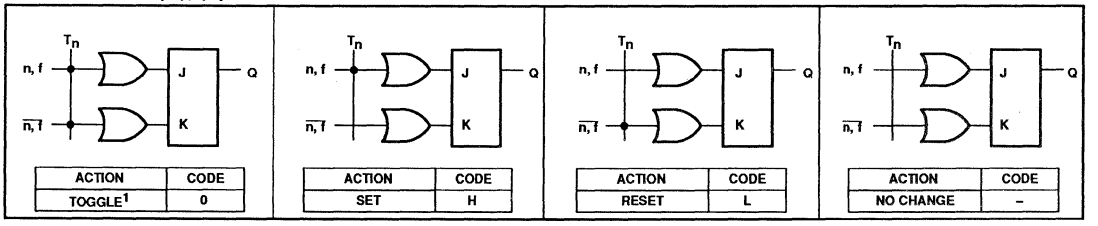
STATE	CODE
INACTIVE[1, 2]	0

STATE	CODE
I, P	H

STATE	CODE
Ī, P̄	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY - (N), (F)



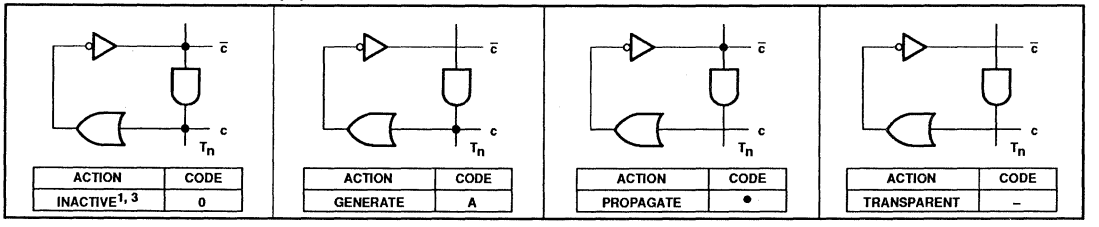
ACTION	CODE
TOGGLE[1]	0

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY - (C)



ACTION	CODE
INACTIVE[1, 3]	0

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all link pairs
2. Any gate T_n will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

PLUS105 PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
PHILIPS DEVICE # CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

C_n	
INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

I_m, P_s	
INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

OR

Ns, Fr	
INACTIVE or TOGGLE	0
SET	H
RESET	L
NO CHANGE	—

OPTION

P/E	
PRESET	H
$\overline{OE}$	L

		AND																OPTION (P/E) OR																
TERM	C_n	INPUT (I_m)																PRESENT STATE (P_s)						REMARKS	NEXT STATE (Ns)						OUTPUT (Fr)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0		5	4	3	2	1	0	7	6	5	4	3	2	1	0
0																																					
1																																					
2																																					
3																																					
4																																					
5																																					
6																																					
7																																					
8																																					
9																																					
10																																					
11																																					
12																																					
13																																					
14																																					
15																																					
16																																					
17																																					
18																																					
19																																					
20																																					
21																																					
22																																					
23																																					
24																																					
25																																					
26																																					
27																																					
28																																					
29																																					
30																																					
31																																					
32																																					
33																																					
34																																					
35																																					
36																																					
37																																					
38																																					
39																																					
40																																					
41																																					
42																																					
43																																					
44																																					
45																																					
46																																					
47																																					
PIN NO.		20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9														10	11	12	13	15	16	17	18
VARIABLE NAME																																					

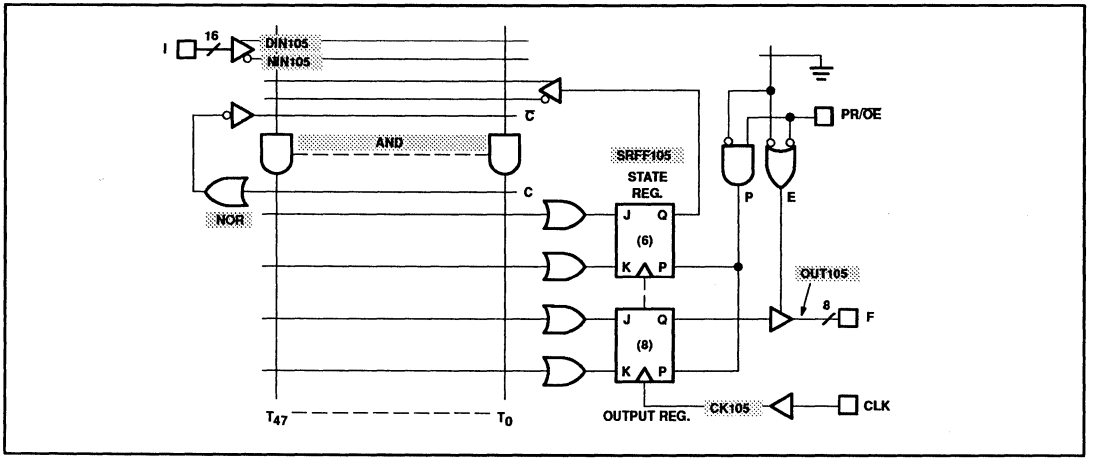
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-55

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

DESCRIPTION

The PLUS105-70 is a bipolar programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 48 AND gates are connected to the 16 external dedicated inputs (I0 - I15) and to the feedback paths of the 6 buried State Registers (Q_{P0}-Q_{P5}). Because the OR array is programmable, any one or all of the 48 transition terms can be connected to any or all of the State and Output Registers.

All state transition terms can include True, False and Don't Care states of the controlling state variables. A Complement Transition Array supports complex IF-THEN-ELSE state transitions with a single product term.

The PLUS105-70 device features edge-triggered, J-K flip-flops, which provide the added flexibility of the toggle function which is indeterminate on S-R flip-flops. Because the J-K function is a superset of the S-R flip-flop function, the PLUS105-70 is backward compatible with all 105-type devices that have S-R flip-flops. Asynchronous Preset/Output Enable functions are available.

The PLUS105-70 is pin-for-pin and software compatible with Philips Semiconductors PLS105 and PLS105A Logic Sequencers, as well as other commercially available 105-type programmable logic devices.

To facilitate testing of state machine designs, diagnostic mode features for register preset and buried state register observability have been incorporated into the PLUS105-70 device architecture.

Ordering codes are listed in the Ordering Information Table.

FEATURES

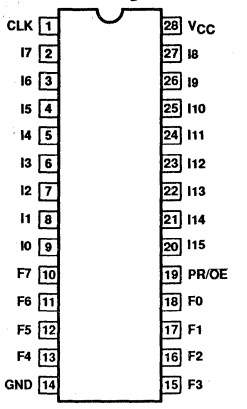
- 70MHz operating frequency
  - 100MHz clock rate
  - No OR term loading restrictions
- Available in 300mil skinny DIP, 600mil-wide DIP, and PLCC packages
- Pin and software compatible with other commericially available 105 sequencers
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked J-K (or S-R) flip-flops
- Security fuse
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to (all "1"s) of internal registers
- Power dissipation: 800mW (typ.)
- TTL compatible
- Single +5V supply
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security Locking systems
- Counters
- Shift registers

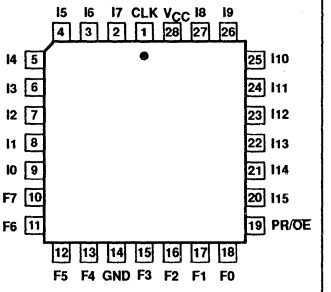
PIN CONFIGURATIONS

N Packages



N = Plastic DIP (600mil-wide)
N3 = Plastic DIP (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

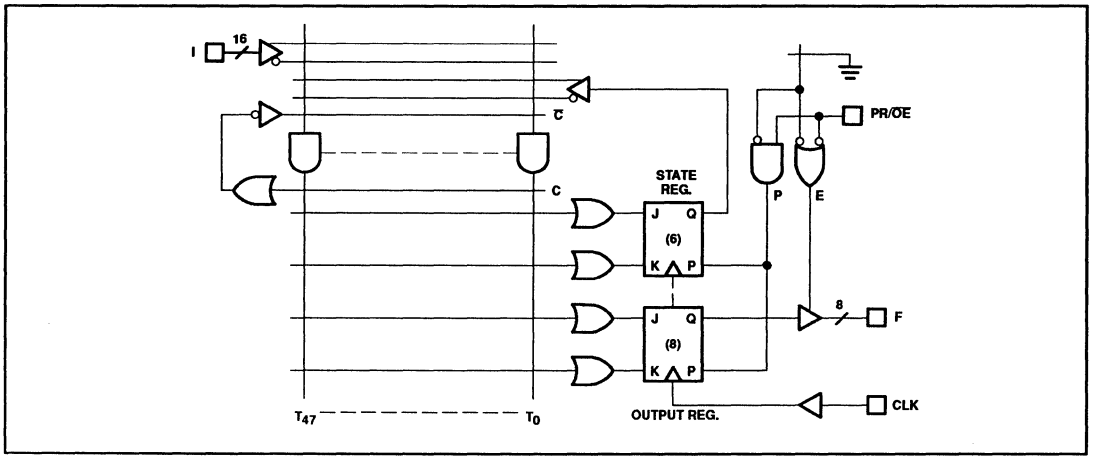
ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
28-pin Plastic Dual-In-Line, 600mil-wide	PLUS105-70N	0413B
28-pin Plastic Dual-In-Line, 300mil-wide	PLUS105-70N3	0864D
28-pin Plastic Leaded Chip Carrier, 450mil-square	PLUS105-70A	0401F

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

FUNCTIONAL DIAGRAM



PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both state and output registers.	Active-High (H)
2-9, 26, 27 20-22	I0 - I9, I13 - I15	**Logic Inputs:** The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/ Low (H/L)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When I12 is held at +10V, device outputs F0 - F5 reflect the contents of State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0 - F5 become direct inputs for State Register bits P0 - P5; a Low-to-High transition on the clock line loads the values on pins F0 - F5 into the State Register bits P0 - P5. The contents of each Output Register remains unaltered.	Active-High/ Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I10 is held at +10V, device outputs F0 - F7 become direct inputs for Output Register bits Q0 - Q7; a Low-to-High transition on the clock line loads the values on pins F0 - F7 into the Output Register bits Q0 - Q7. The contents of each State Register remains unaltered.	Active-High/ Low (H/L)
10-13 15-18	F0 - F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register bits Q0 - Q7, when enabled. When I12 is held at +10V, F0 - F5 = (P0 - P5). When I11 is held at +10V, F0 - F5 become inputs to State Register bits P0 - P5. When I10 is held at +10V, F0 - F7 become inputs to Output Register bits Q0 - Q7.	Active-High (H)
19	PR/$\overline{OE}$	**Preset or Output Enable Input:** **A user programmable function:**	
		• **Preset:** Provides an asynchronous preset to logic "1" of all State and Output Register bits. PR overrides Clock, and when held High, clocking is inhibited and F0 - F7 are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after the Preset signal goes Low. See timing definitions.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F0 - F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

TRUTH TABLE[1, 2, 3, 4, 5, 6]

V_{CC}	OPTION PR	OPTION $\overline{OE}$	I10	I11	I12	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H	H	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

1. Positive Logic:
   J-K (or S/R) = $T_0 + T_1 + T_2 + \ldots T_{48}$
   $T_n = (C_0)\ (I0, I1, I2, \ldots)\ (P_0, P_1, \ldots P_5)$
2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤ 5.5V)
6. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

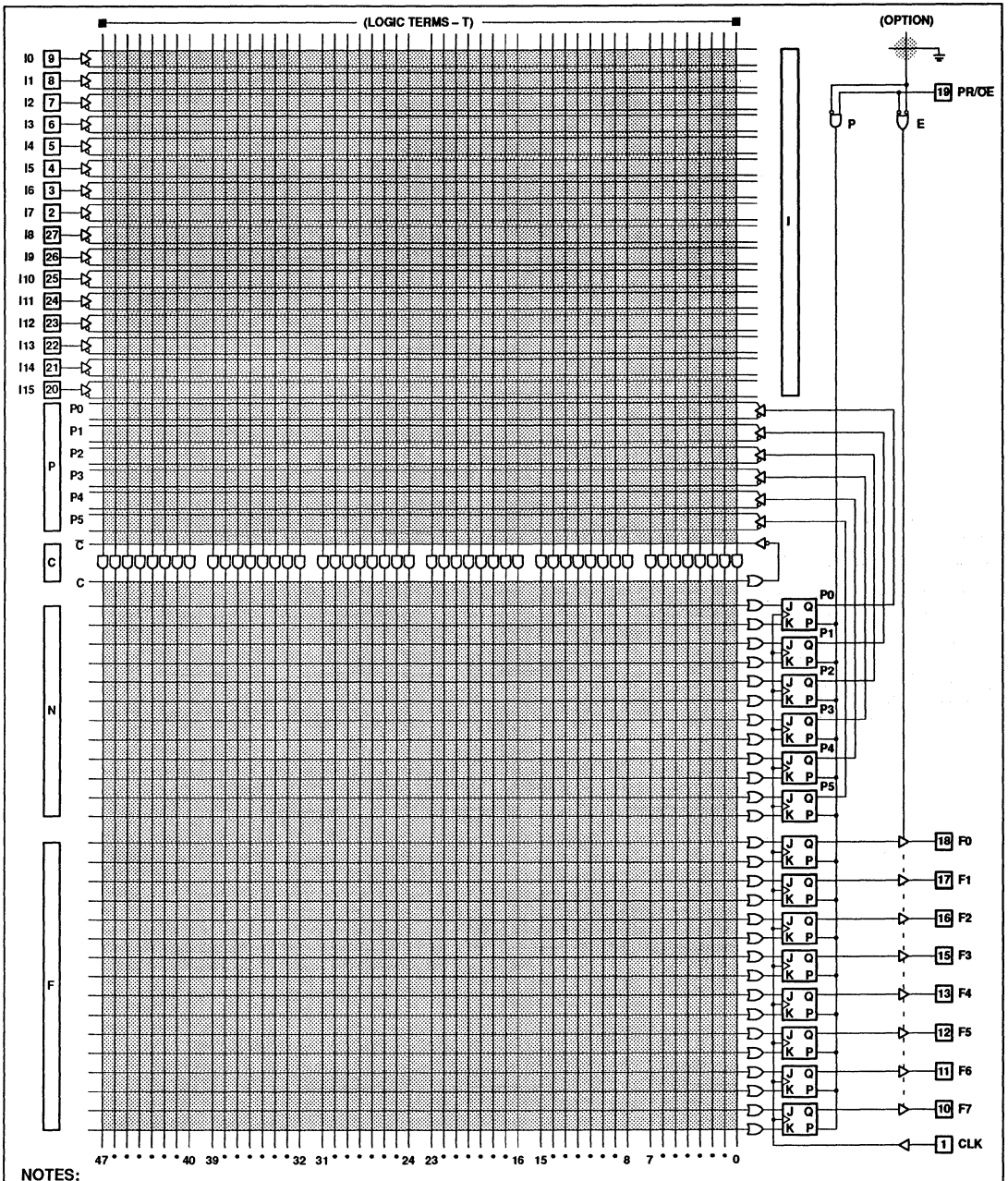
A factory-shipped virgin device contains all fusible links intact, such that:

1. PR/$\overline{OE}$ option is set to PR. Note that even if the PR function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are disabled (0).
3. All J-K flip-flop inputs are disabled (0).

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

LOGIC DIAGRAM

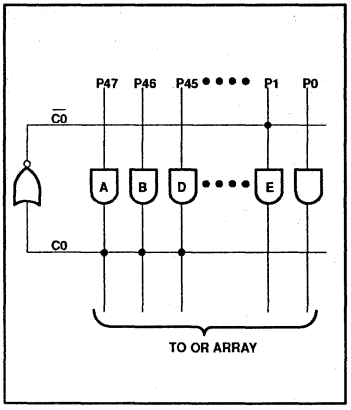


NOTES:
1. All AND gate inputs with a blown link float to a logic "1".
2. All OR gate inputs with a blown fuse float to logic "0".
3. Programmable connection.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

COMPLEMENT ARRAY DETAIL



The complement array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The complement array is a single OR gate with inputs from the AND array. The output of the complement array is inverted and fedback to the AND array (NOR function). The output of the array will be LOW if any one or more of the AND terms connected to it are active (HIGH). If, however, all the connected terms are inactive (LOW), which is a classic unknown state, the output of the complement array will be HIGH.

Consider the product terms A, B and D that represent defined states. They are also connected to the input of the complement array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to product term E, which could be used in turn to preset the state machine to a known state. Without the complement array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that use of the Complement Array adds an additional delay path through the device. Refer to the AC Electrical Characteristics for details.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

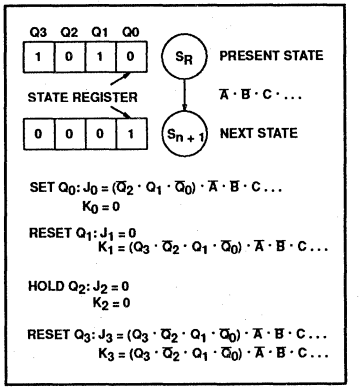
ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage		+7.0	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	-30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	-65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

LOGIC FUNCTION



DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq 75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage[2]						
		V_{CC} = MIN				
V_{OH}	High	I_{OH} = −2mA	2.4			V
V_{OL}	Low	I_{OL} = 9.6mA		0.35	0.45	V
Input current						
		V_{CC} = MAX				
I_{IH}	High	V_{IN} = V_{CC}		<1	30	µA
I_{IL}	Low	V_{IN} = 0.45V		−20	−250	µA
Output current						
		V_{CC} = MAX				
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = 2.7V		1	40	µA
		V_{OUT} = 0.45V		−1	−40	µA
I_{OS}	Short circuit[3, 4]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		160	200	mA
Capacitance						
		V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{OUT} = 2.0V		10		pF

NOTES:

1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short circuit should not exceed 1 second.
5. I_{CC} is measured with the PR/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

AC ELECTRICAL CHARACTERISTICS

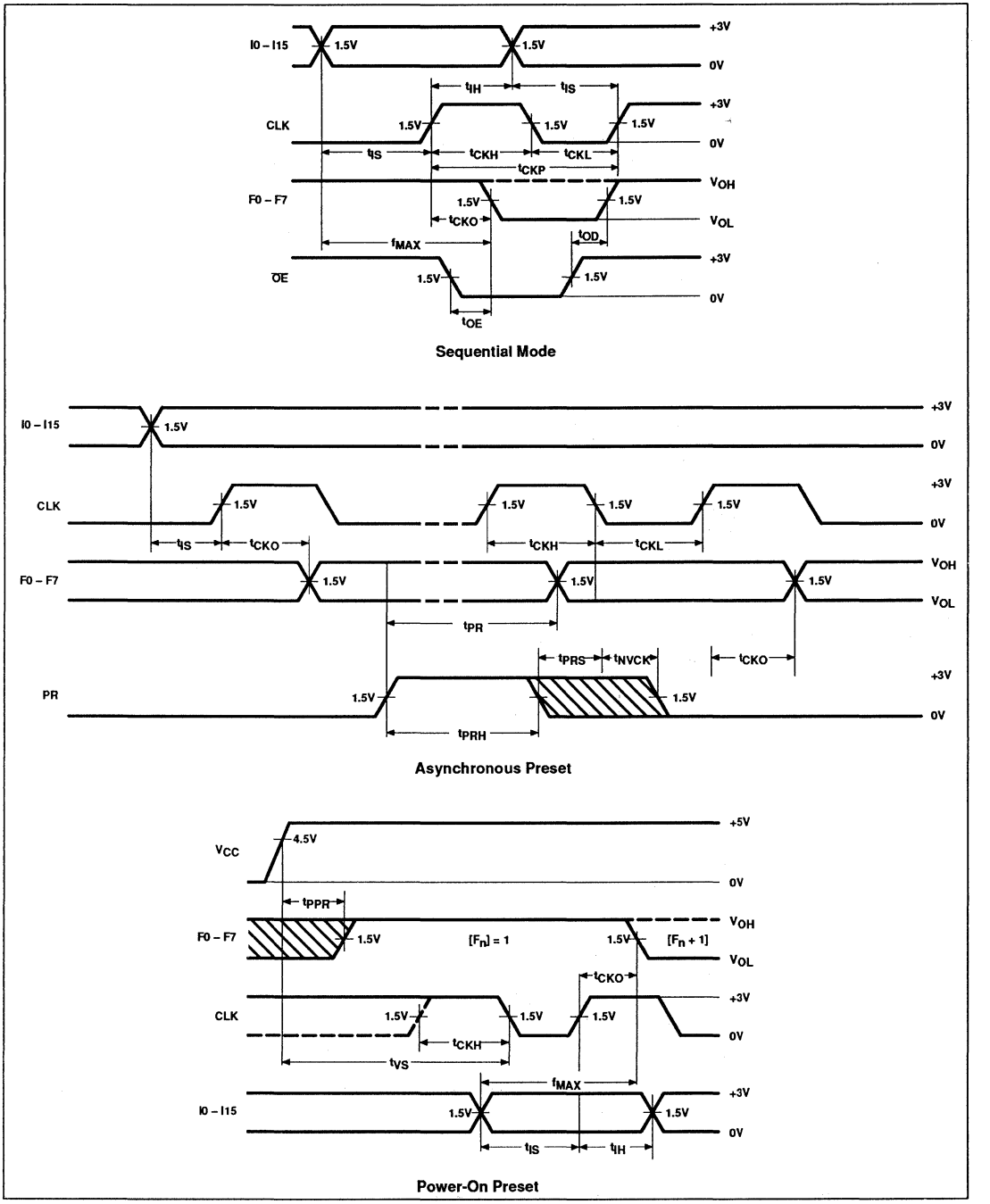
$R_1 = 470\Omega$, $R_2 = 1K\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Pulse Width							
t_{CKH}	Clock High	CK +	CK –	5			ns
t_{CKL}	Clock Low	CK –	CK +	5			ns
t_{CKP}	Clock Period	CK +	CK +	10			ns
t_{PRH}	Preset pulse	PR +	PR –	10			ns
Setup Time							
t_{IS1}	Input	Input ±	CK +	8			ns
t_{IS2}	Input (through Complement Array)	Input ±	CK +	15			ns
t_{VS}	Power-on preset	V_{CC} +	CK –	0			ns
t_{PRS}	Clock resume (after preset)	PR –	CK –	0			ns
t_{NVCK}	Clock lockout (before preset)	CK –	PR –	10			ns
Hold Time							
t_{IH}	Input	CK +	Input ±	0			ns
Diagnostic Mode							
t_{RJS}	Initialization of diagnostic mode	I10, I11 or I12 + (to 10V)	F_n as inputs	50			ns
t_{RJH}	Clock for diagnostic mode	CK +	Register input jam	50			ns
Propagation Delay[2]							
t_{CKO}	Clock	CK +	Output ±			6	ns
t_{OE}	Output enable[3]	OE –	Output –			6.5	ns
t_{OD}	Output disable[3]	OE +	Output +			6.5	ns
t_{PR}	Preset	PR +	Output +			12	ns
t_{PPR}	Power-on preset	V_{CC} +	Output +			10	ns
Frequency of Operation							
f_{MAX1}	Without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO}}\right)$	Input ±	Output ±	71.4			MHz
f_{MAX2}	With Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO}}\right)$	Input thru Complement Array ±	Output ±	47.6			MHz
f_{MAX3}	Internal feedback without Complement Array $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	100.0			MHz
f_{MAX4}	Internal feedback with Complement Array $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Input ±	66.7			MHz
f_{CLK}	Clock period	CK +	CK +	100.0			MHz

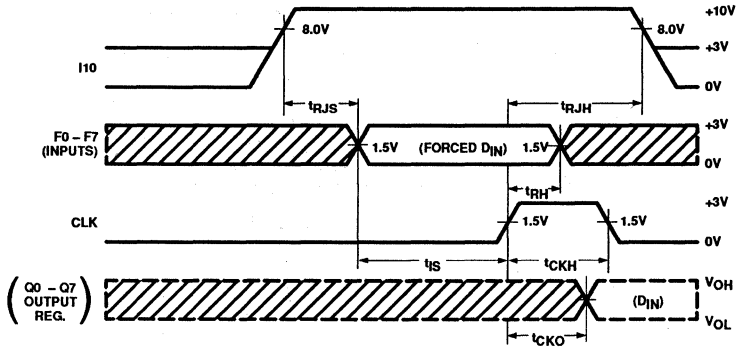
NOTES:

1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. All propagation delays and setup times are meausred and specified under worst case conditions.
3. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

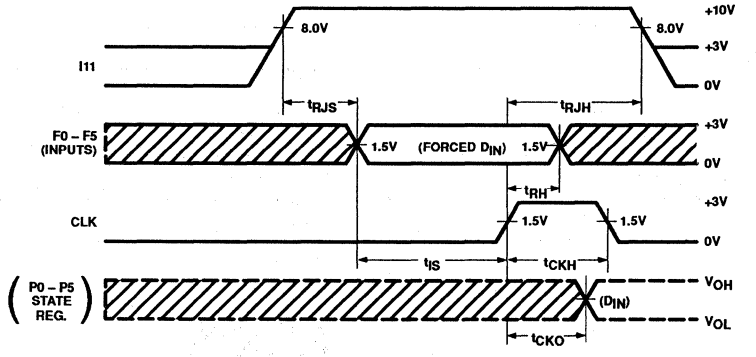
TIMING DIAGRAMS



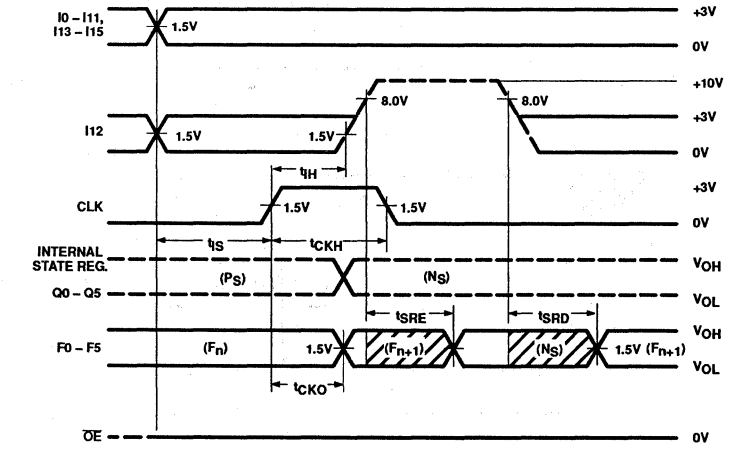
TIMING DIAGRAMS (Continued)



Diagnostic Mode—Output Register Input Jam



Diagnostic Mode—State Register Input Jam



Diagnostic Mode—State Register Outputs

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

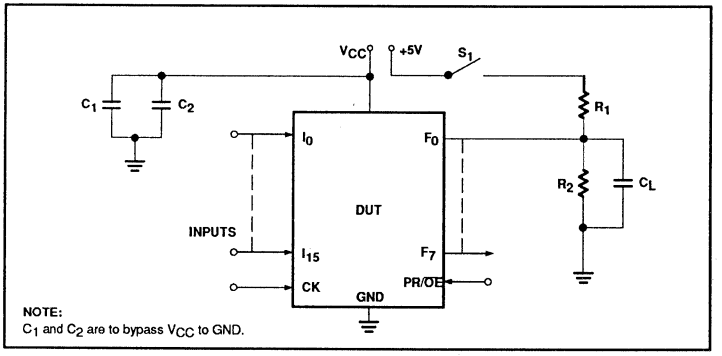
TIMING DEFINITIONS

SYMBOL	PARAMETER
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{CKH}	Width of input clock pulse
t_{CKL}	Interval between clock pulses.
t_{CKO}	Delay between positive transition of Clock and when Outputs become valid (with PR/$\overline{OE}$ Low).
t_{CKP}	Minimum guaranteed clock period.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous PRESET to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{PR}	Delay between positive transition of Preset and when Outputs become valid at "1".
t_{PRH}	Width of preset input pulse.
t_{PRS}	Required delay between negative transition of Asynchronous Preset and the first positive transition of Clock.
t_{RH}	Required delay between positive transition of clock and the end of valid input data (F0 – F7 as inputs), when jamming data into the State or Output registers in the Diagnostic Mode.

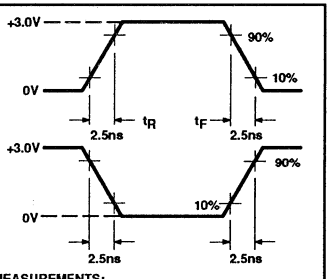
SYMBOL	PARAMETER
t_{RJH}	Required delay between positive transition of clock and return of input I10, I11 or I12 from Diagnostic Mode (10V).
t_{RJS}	Required delay between inputs I10, I11 or I12 transition to Diagnostic Mode (10V), and when the output pins become available as inputs.
t_{SRD}	Delay between input (I12) transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock puise.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency)
f_{MAX}	Minimum guaranteed operating frequency.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

LOGIC PROGRAMMING

PLUS105-70 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™, CUPL™ and PALASM® 90 design software packages also support the PLUS105-70 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

PLUS105-70 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP.

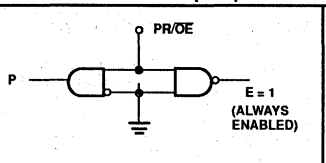
To implement the desired logic functions, the state of each logic variable from logic equations (I, B, O, P, etc.) is assigned a symbol. The symbols for TRUE, COMPLEMENT, INACTIVE, PRESET, etc., are defined below.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.
PALASM is a registered trademark of AMD Corp.

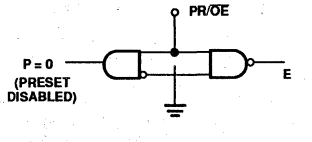
Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

PRESET/OE OPTION - (P/E)



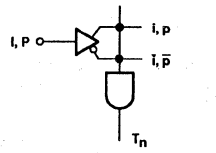
OPTION	CODE
PRESET[1]	H



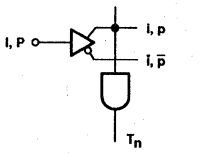
OPTION	CODE
$\overline{OE}$	L

PROGRAMMING THE PLUS105-70:
The PLUS105-70 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

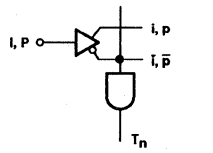
"AND" ARRAY - (I), (P)



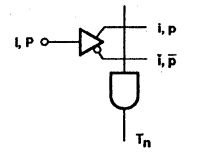
STATE	CODE
INACTIVE[1, 2]	0



STATE	CODE
I, P	H

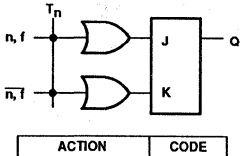


STATE	CODE
$\overline{I}$, $\overline{P}$	L

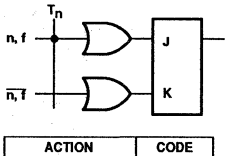


STATE	CODE
DON'T CARE	–

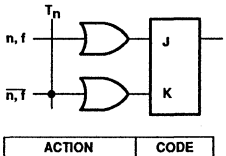
"OR" ARRAY - (N), (F)



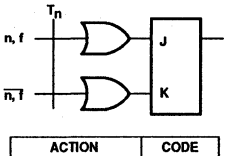
ACTION	CODE
TOGGLE[1]	0



ACTION	CODE
SET	H

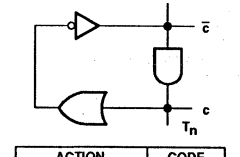


ACTION	CODE
RESET	L

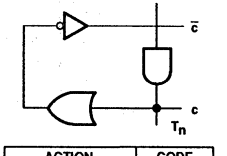


ACTION	CODE
NO CHANGE	–

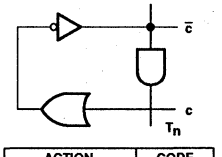
"COMPLEMENT" ARRAY - (C)



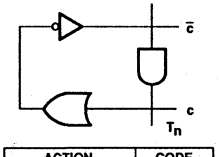
ACTION	CODE
INACTIVE[1, 3]	0



ACTION	CODE
GENERATE	A



ACTION	CODE
PROPAGATE	•



ACTION	CODE
TRANSPARENT	–

NOTES:
1. This is the initial unprogrammed state of all link pairs
2. Any gate T_n will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

PLUS105 PROGRAM TABLE

CUSTOMER NAME_____
PURCHASE ORDER #_____
PHILIPS DEVICE #_____ CF (XXXX)
CUSTOMER SYMBOLIZED PART #_____
TOTAL NUMBER OF PARTS_____
PROGRAM TABLE_____
REV_____
DATE_____

PROGRAM TABLE ENTRIES

AND

C_n	
INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

I_m, P_s	
INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

OR

N_s, Fr	
INACTIVE or TOGGLE	0
SET	H
RESET	L
NO CHANGE	—

OPTION

P/E	
PRESET	H
$\overline{OE}$	L

TERM	C_n	AND: INPUT (I_m) 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	AND: PRESENT STATE (P_s) 5 4 3 2 1 0	REMARKS	OPTION (P/E) OR: NEXT STATE (N_s) 5 4 3 2 1 0	OUTPUT (Fr) 7 6 5 4 3 2 1 0
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
PIN NO.		20 21 22 23 24 25 26 27 2 3 4 5 6 7 8 9				10 11 12 13 15 16 17 18
VARIABLE NAME						

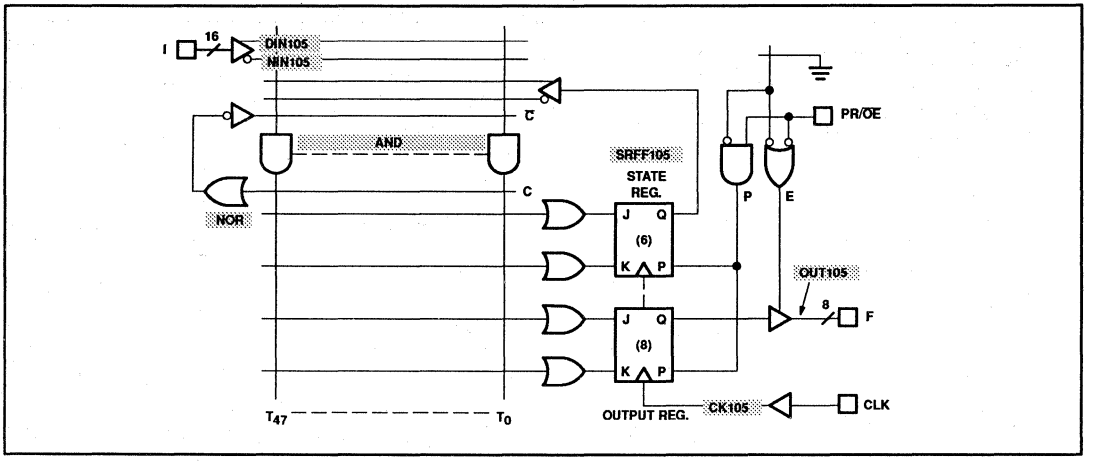
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the P/E option, exists in the table, shown BLANK instead for clarity.
2. Unused C_n, I_m, and P_s bits are normally programmed Don't Care (–).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (–) for maximum speed.
4. Letters in variable fields are used as identifiers by logic type programmers.

Programmable logic sequencer (16 × 48 × 8)

PLUS105-70

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

DESCRIPTION

The PLUS405 devices are bipolar, programmable state machines of the Mealy type. Both the AND and the OR array are user-programmable. All 64 AND gates are connected to the 16 external dedicated inputs (I0 - I15) and to the feedback paths of the 8 on-chip State Registers (Q_{P0} - Q_{P7}). Two complement arrays support complex IF-THEN-ELSE state transitions with a single product term (input variables C0, C1).

All state transition terms can include True, False and Don't Care states of the controlling state variables. All AND gates are merged into the programmable OR array to issue the next-state and next-output commands to their respective registers. Because the OR array is programmable, any one or all of the 64 transition terms can be connected to any or all of the State and Output Registers.

All state (Q_{P0} - Q_{P7}) and output (Q_{F0} - Q_{F7}) registers are edge-triggered, clocked J-K flip-flops, with Asynchronous Preset and Reset options. The PLUS405 architecture provides the added flexibility of the J-K toggle function which is indeterminate on S-R flip-flops. Each register may be individually programmed such that a specific Preset-Reset pattern is initialized when the initialization pin is raised to a logic level "1". This feature allows the state machine to be asynchronously initialized to known internal state and output conditions, prior to proceeding through a sequence of state transitions. Upon power-up, all registers are unconditionally preset to "1". If desired, the initialization input pin (INIT) can be converted to an $\overline{\text{Output Enable}}$ ($\overline{\text{OE}}$) function as an additional user-programmable feature.

Availability of two user-programmable clocks allows the user to design two independently clocked state machine functions consisting of four state and four output bits each.

Order codes are listed in the Ordering Information Table.

FEATURES

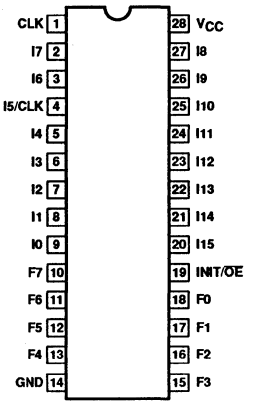
- PLUS405-37
  - f_{MAX} = 37MHz
  - 50MHz clock rate
- PLUS405-45
  - f_{MAX} = 45MHz
  - 58.8MHz clock rate
- Functional superset of PLS105/105A
- Field-programmable (Ti-W fusible link)
- 16 input variables
- 8 output functions
- 64 transition terms
- 8-bit State Register
- 8-bit Output Register
- 2 transition Complement Arrays
- Multiple clocks*
- Programmable Asynchronous Initialization or $\overline{\text{Output Enable}}$
- Power-on preset of all registers to "1"
- "On-chip" diagnostic test mode features for access to state and output registers
- 950mW power dissipation (typ.)
- TTL compatible
- J-K or S-R flip-flop functions
- Automatic "Hold" states
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator contollers
- Security locking systems
- Counters
- Shift registers

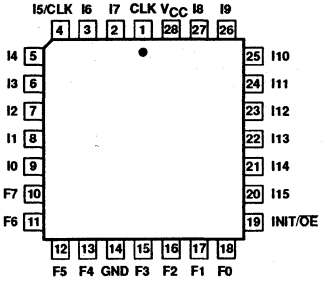
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE	DRAWING NUMBER
28-Pin Plastic DIP (600mil-wide)	45MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–45N	0413B
28-Pin Plastic DIP (600mil-wide)	37MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–37N	0413B
28-Pin Plastic Leaded Chip Carrier	45MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–45A	0401F
28-Pin Plastic Leaded Chip Carrier	37MHz ($t_{IS1} + t_{CKO1}$)	PLUS405–37A	0401F

*Refer to AC Specifications for clock and operating frequencies when using multiple clocks.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both state and output registers. Pin 1 only clocks P0–3 and F0–3 if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	I0–I4, I7, I6 I8–I9 I13–I15	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P4–7 and Output Registers F4–7, as above. Note that input buffer I_5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I12 is held at +10V, device outputs F0–F7 reflect the contents of State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0–F7 become direct inputs for State Register bits P0–P7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I_{10} is held at +10V, device outputs F0–F7 become direct inputs for Output Register bits Q0–Q7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the Output Register bits Q0–Q7. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	F0 – F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits Q0–Q7, when enabled. When I12 is held at +10V, F0–F7 = (P0–P7). When I11 is held at +10V, F0–F7 become inputs to State Register bits P0–P7. When I10 is held at +10V, F0–F7 become inputs to Output Register bits Q0–Q7.	Active-High (H)
19	INIT/$\overline{OE}$	**Initialization or $\overline{\text{Output Enable}}$ Input:** A user programmable function:	
		• **Initialization:** Provides an asynchronous preset to logic "1" or reset to logic "0" of all State and Output Register bits, determined individually for each register bit through user programming. INIT overrides Clock, and when held High, clocking is inhibited and F0–F7 and P0–P7 are in their initialization state. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after INIT goes Low. See timing definition for t_{NVCK} and t_{VCK}.	Active-High (H)
		• **$\overline{\text{Output Enable}}$:** Provides an output enable function to buffers F0–F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION INIT	OPTION $\overline{OE}$	I10	I11	I12	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H/L	H/L	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

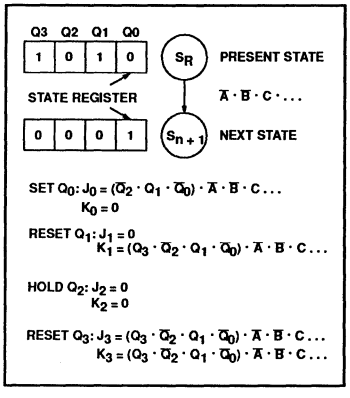
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   T_n = (C0, C1) (I0, I1, I2, . . .) (P0, P1, . . . P7)
2. Either Initialization (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤5.5V)
6. H/L implies that either a High or a Low can occur, depending upon user-programmed selection (each State and Output Register individually programmable).
7. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$ is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All S/R (or J/K) flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array preprogrammed with a standard test pattern.
5. Clock 2 is inactive.

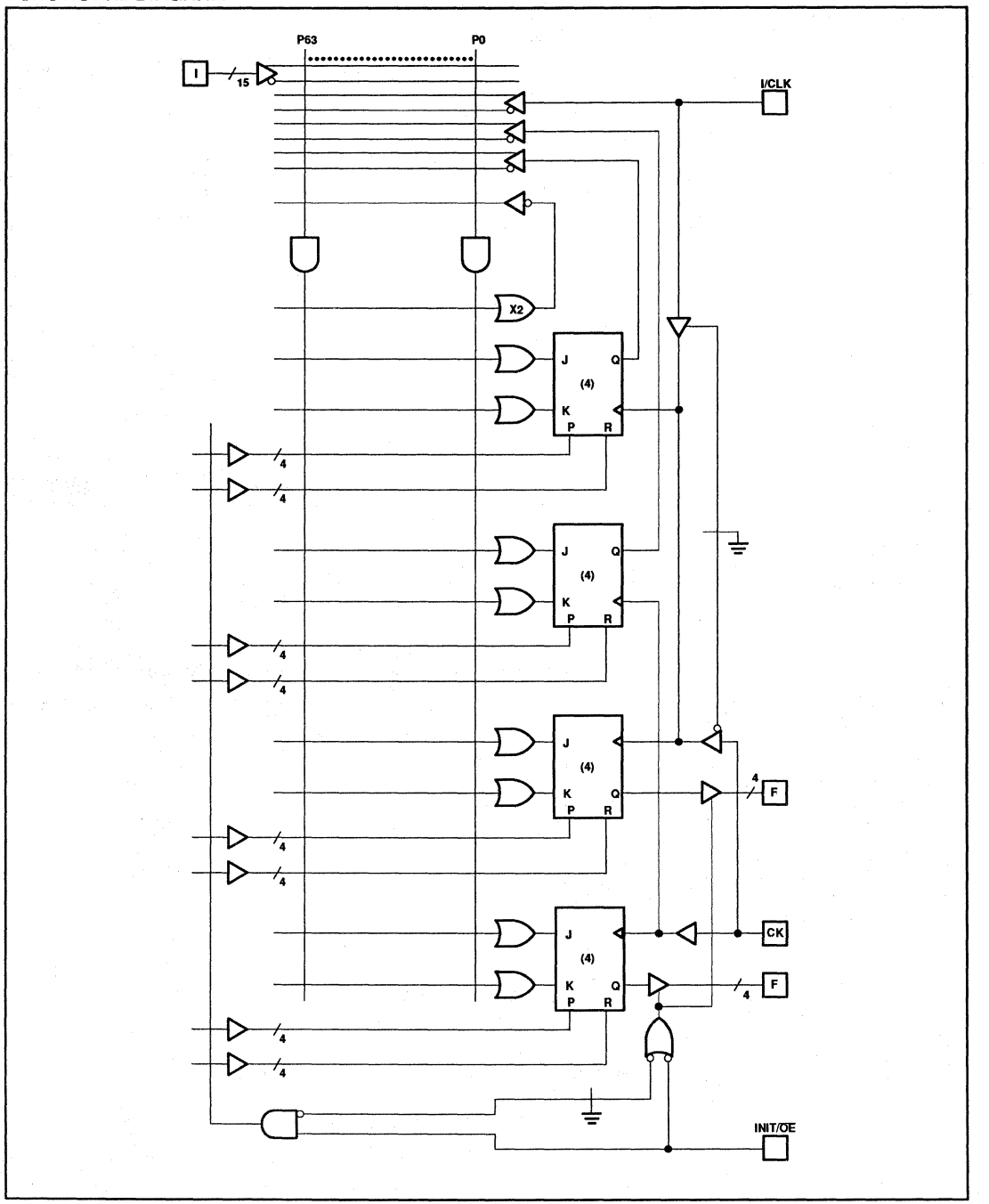
LOGIC FUNCTION



Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

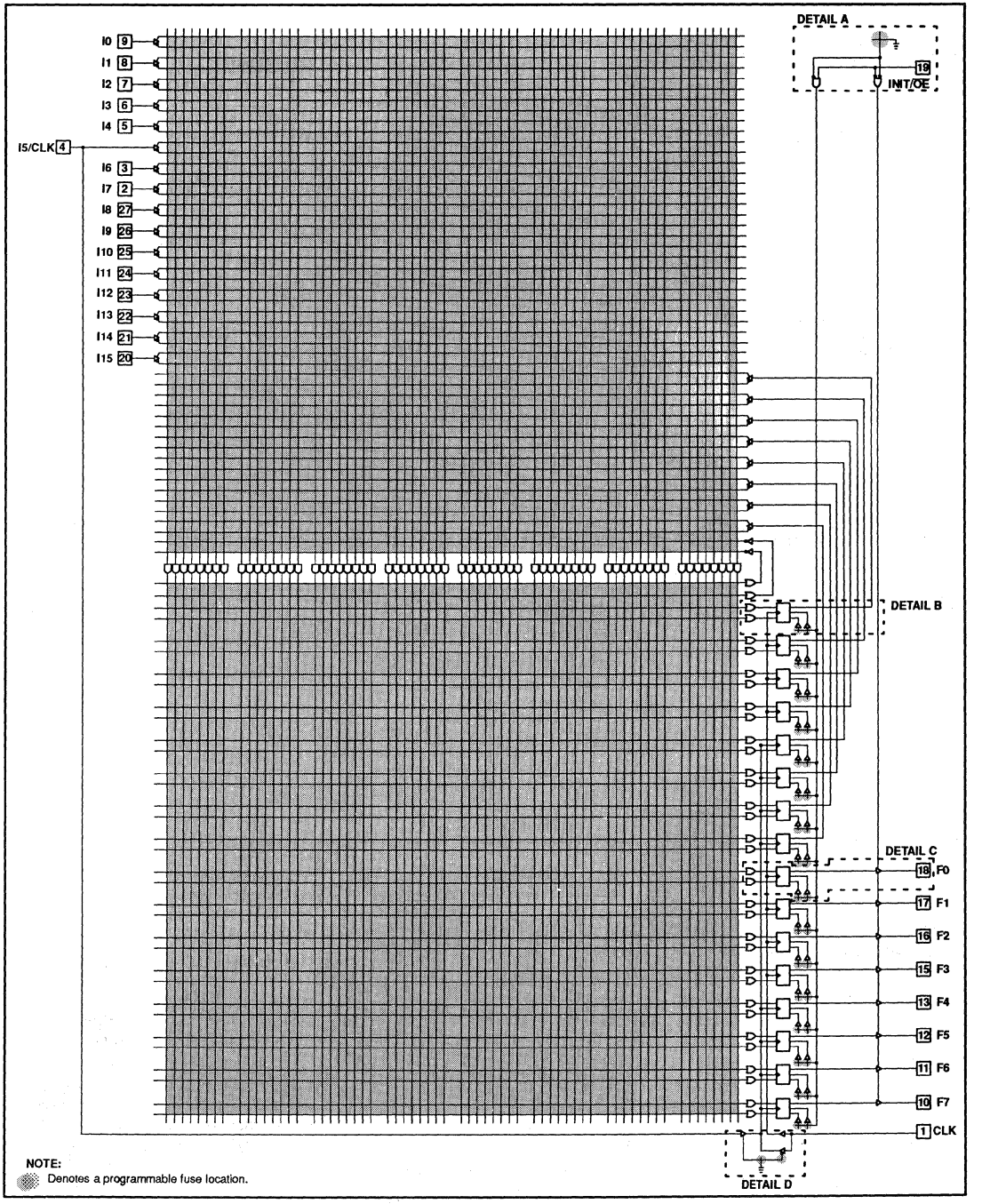
FUNCTIONAL DIAGRAM



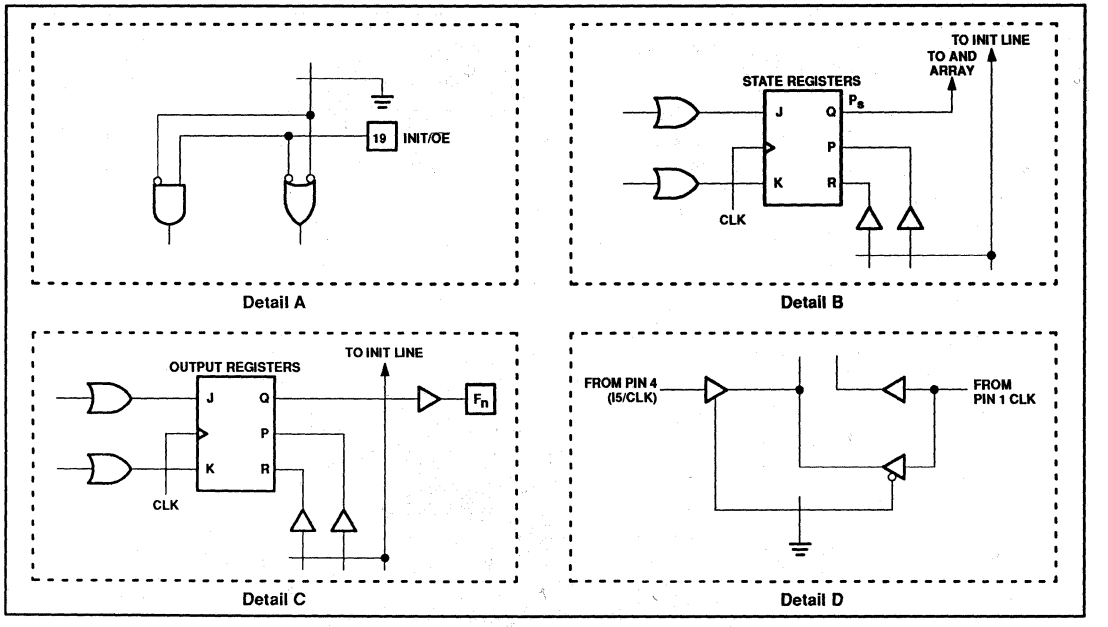
Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

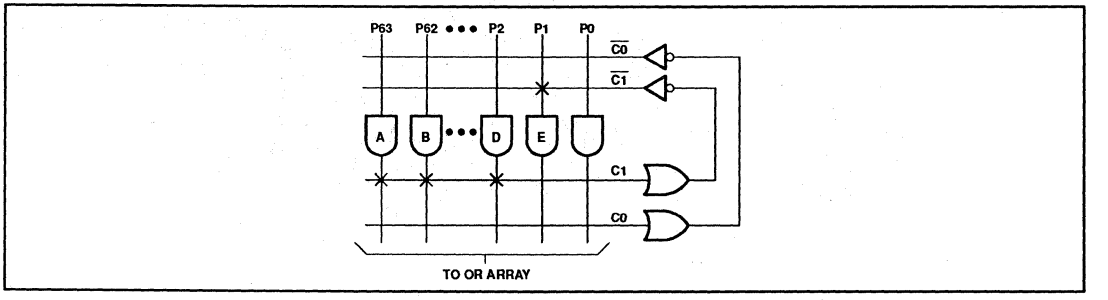
LOGIC DIAGRAM



DETAILS FOR REGISTERS FOR PLUS405



COMPLEMENT ARRAY DETAIL



The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLUS405 has 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{Sstg}	Storage temperature range	–65 to +150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = V_{CC}		<1	30	µA
I_{IL}	Low	V_{CC} = MAX, V_{IN} = 0.45V		–20	–250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{CC} = MAX, V_{OUT} = 2.7V		1	40	µA
		V_{CC} = MAX, V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit [3, 4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		190	225	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{CC} = 5.0V, V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short-circuit should not exceed one second.
5. I_{CC} is measured with the INIT/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLUS405-37			PLUS405-45			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Pulse width										
t_{CKH1}	Clock High; CLK1 (Pin 1)	CK+	CK–	10	8		8.5	7		ns
t_{CKL1}	Clock Low; CLK1 (Pin 1)	CK–	CK+	10	8		8.5	7		ns
t_{CKP1}	CLK1 Period	CK+	CK+	20	16		17	14		ns
t_{CKH2}	Clock High; CLK2 (Pin 4)	CK+	CK–	10	8		10	8		ns
t_{CKL2}	Clock Low; CLK2 (Pin 4)	CK–	CK+	10	8		10	8		ns
t_{CKP2}	CLK2 Period	CK+	CK+	20	16		20	16		ns
t_{INITH}	Initialization pulse	INIT–	INIT+	15	10		15	8		ns
Setup time										
t_{IS1}	Input	Input ±	CK+	15	12		12	10		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK+	25	20		22	18		ns
t_{VS}	Power-on preset	V_{CC}+	CK–	0	–10		0	–10		ns
t_{VCK}	Clock resume (after Initialization)	INIT–	CK–	0	–5		0	–5		ns
t_{NVCK}	Clock lockout (before Initialization)	CK–	INIT–	15	5		15	5		ns
Hold time										
t_{IH}	Input	CK+	Input ±	0	–5		0	–5		ns
Propagation delay										
t_{CKO1}	Clock1 (Pin 1)	CK1+	Output ±		10	12		8	10	ns
t_{CKO2}	Clock2 (Pin 4)	CK2+	Output ±		12	15		10	12	ns
t_{OE}[2]	Output Enable	OE–	Output –		12	15		12	15	ns
t_{OD}[2]	Output Disable	OE+	Output +		12	15		12	15	ns
t_{INIT}	Initialization	INIT+	Output +		15	20		15	20	ns
t_{PPR}	Power-on Preset	V_{CC} +	Output +		0	10		0	10	ns

Notes on following page

AC ELECTRICAL CHARACTERISTICS (Continued)

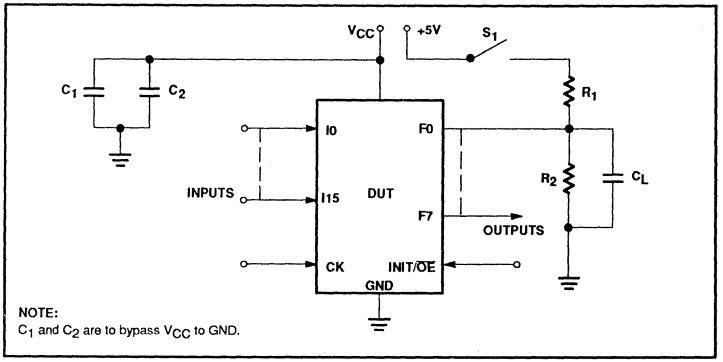
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS						UNIT
				PLUS405-37			PLUS405-45			
				MIN	TYP[1]	MAX	MIN	TYP[1]	MAX	
Frequency of operation										
f_{MAX1}	CLK1; without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO1}}\right)$	Input ±	Output ±	37.0	45.5		45.5	55.6		MHz
f_{MAX2}	CLK2; without Complement Array $\left(\frac{1}{t_{IS1} + t_{CKO2}}\right)$	Input ±	Output ±	33.0	41.7		41.7	50.0		MHz
f_{MAX3}	CLK1; with Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO1}}\right)$	Input thru Complement Array ±	Output ±	27.0	33.3		31.3	38.5		MHz
f_{MAX4}	CLK2; with Complement Array $\left(\frac{1}{t_{IS2} + t_{CKO2}}\right)$	Input thru Complement Array ±	Output ±	25.0	31.3		29.4	35.7		MHz
f_{MAX5}	Internal feedback without Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Intput ±	50.0	62.5		58.8	72.4		MHz
f_{MAX6}	Internal feedback with Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{IS2}}\right)$	Register Output thru Complement Array ±	Register Intput ±	40.0	50.0		45.5	55.6		MHz
f_{CLK}	Minimum guaranteed clock frequency	CK +	CK +	50.0	62.5		58.8	72.4		MHZ

NOTES:

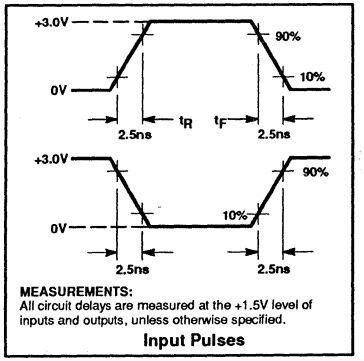
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times are measured and specified under worst case conditions.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

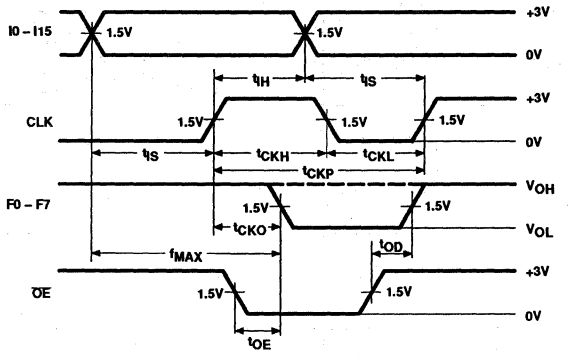
VOLTAGE WAVEFORMS



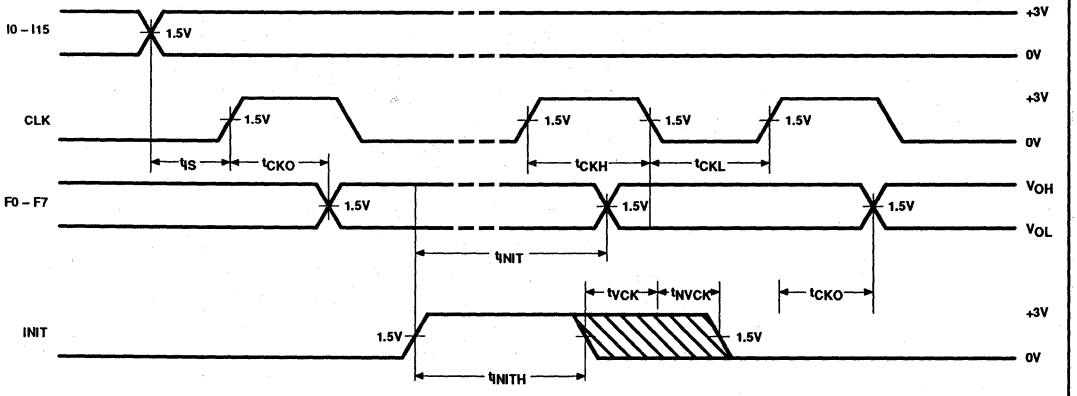
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

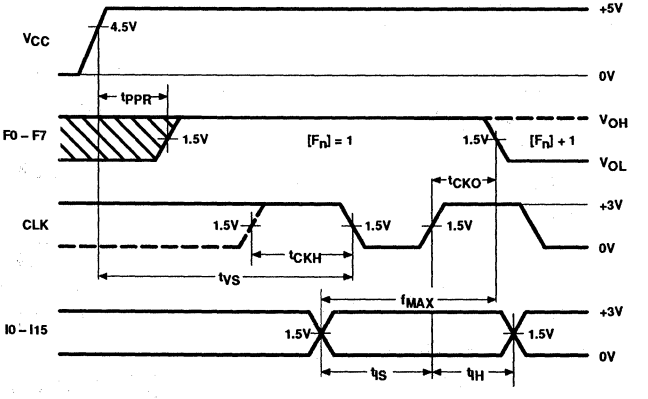
TIMING DIAGRAMS



Sequential Mode

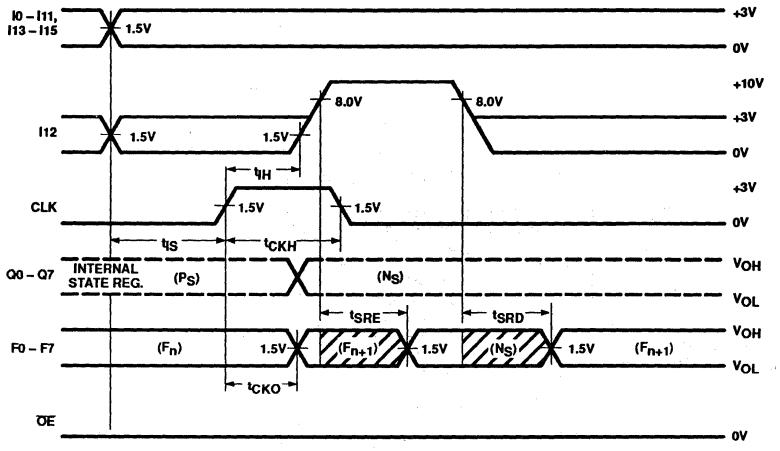


Asynchronous Initialization

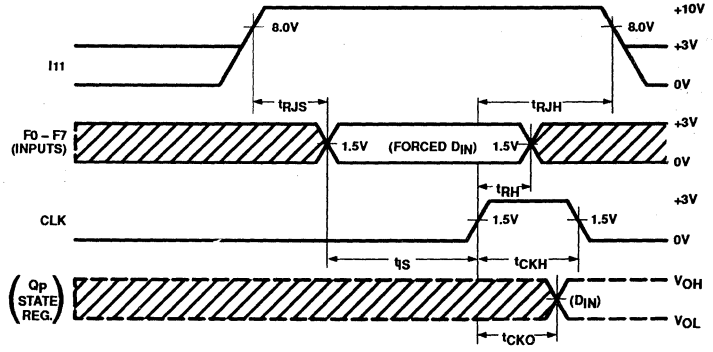


Power-On Preset

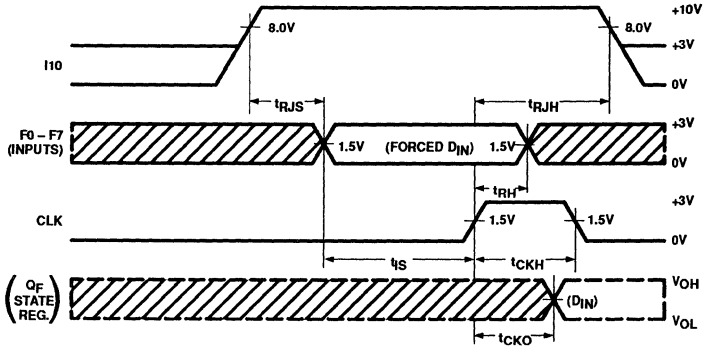
TIMING DIAGRAMS (Continued)



Diagnostic Mode – State Register Outputs



Diagnostic Mode – State Register Input Jam



Diagnostic Mode – Output Register Input Jam

TIMING DEFINITIONS

SYMBOL	PARAMETER
$t_{CKH1, 2}$	Width of input clock pulse.
$t_{CKP1, 2}$	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
$t_{CKO1, 2}$	Delay between positive transition of Clock and when Outputs become valid (with INIT/$\overline{OE}$ Low).
t_{PPR}	Delay between V_{CC} (after power–on) and when Outputs become preset at "1".
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{RJH}	Required delay between positive transition of clock, and return of input I10, I11 or I12 from Diagnostic Mode (10V).
$f_{MAX1, 2}$	Minimum guaranteed operating frequency; input to output (CLK1 and CLK2).
$f_{MAX3, 4}$	Minimum guaranteed operating frequency; input through Complement Array, to output (CLK1 and CLK2).
f_{MAX5}	Minimum guaranteed internal operating frequency; with internal feedback from state register to state register.

SYMBOL	PARAMETER
f_{MAX6}	Minimum guaranteed internal operating frequency with Complement Array, with internal feedback from state register through Complement Array, to state register.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency).
$t_{CKL1, 2}$	Interval between clock pulses.
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{RJS}	Required delay between inputs I11, I10 or I12 transition to Diagnostic Mode (10V), and when the output pins become available as inputs.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{INITH}	Width of initialization input pulse.
t_{VS}	Required delay between V_{CC} (after power–on) and negative transition of Clock preceding first reliable clock pulse.
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF–state.
t_{INIT}	Delay between positive transition of Initialization and when Outputs become valid.
t_{SRD}	Delay between input I12 transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{VCK}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding first reliable clock pulse.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

LOGIC PROGRAMMING

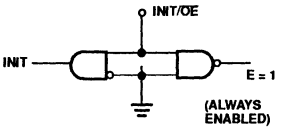
The PLUS405-37/-45 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software packages. ABEL™ and CUPL™ design software packages also support the PLUS405-37/-45 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

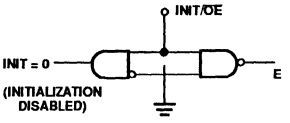
PLUS405-37/-45 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations is assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION/$\overline{OE}$ OPTION – (INIT/$\overline{OE}$)



OPTION	CODE
INITIALIZATION[1]	H

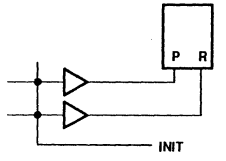


OPTION	CODE
$\overline{OE}$	L

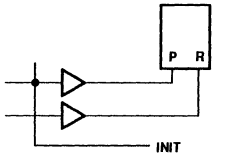
PROGRAMMING THE PLUS405:

The PLUS405 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

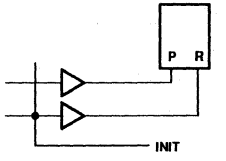
INITIALIZATION OPTION – (INIT)



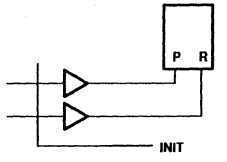
ACTION	CODE
INDETERMINATE[4]	0



ACTION	CODE
PRESET	H

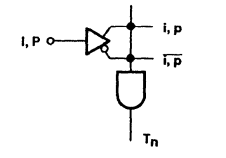


ACTION	CODE
RESET	L

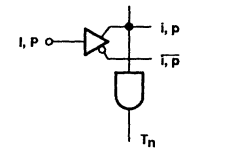


ACTION	CODE
INDETERMINATE[4]	–

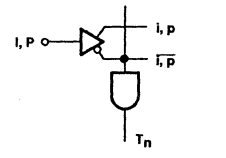
"AND" ARRAY – (I), (P)



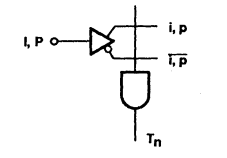
STATE	CODE
INACTIVE[1, 2]	0



STATE	CODE
I, P	H



STATE	CODE
$\overline{I, P}$	L



STATE	CODE
DON'T CARE	–

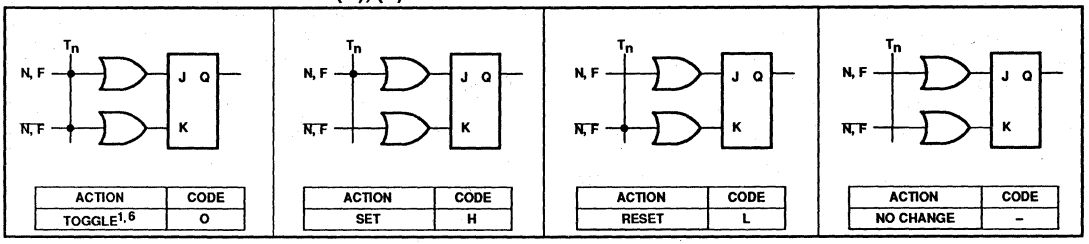
Notes are on next page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

"OR" ARRAY – J-K FUNCTION – (N), (F)



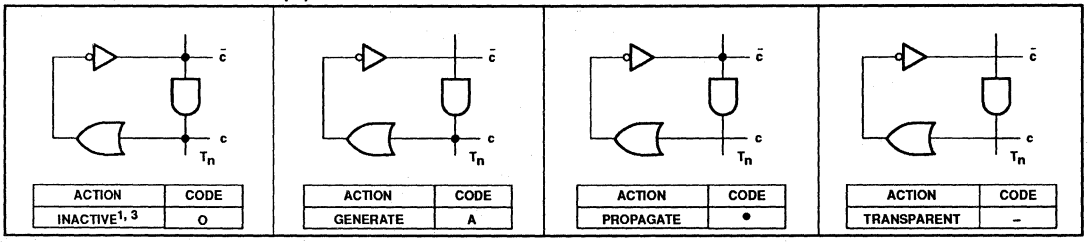
ACTION	CODE
TOGGLE[1,6]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



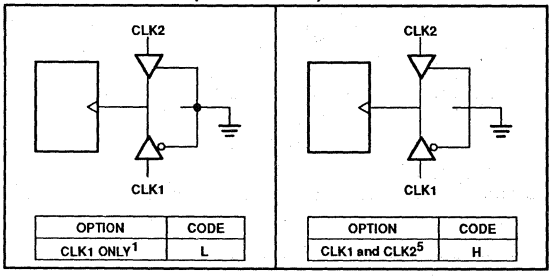
ACTION	CODE
INACTIVE[1,3]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L

OPTION	CODE
CLK1 and CLK2[5]	H

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using INITIALIZATION option.
5. Input buffer I5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. A single product term cannot drive more than 8 registers by itself when used in TOGGLE mode.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

PLUS405 PROGRAM TABLE

AND						OR			OPTIONS		
INACTIVE	0	Im, Ps	INACTIVE	0	Cn	INACTIVE OR TOGGLE	0	Ns, Fr	INIT	H	INIT/OE
I, P	H		GENERATE	A		SET	H		OE	L	
$\overline{I, P}$	L		PROPAGATE	•		RESET	L		CLK1 ONLY	L	CLK1/ CLK2
DON'T CARE	–		TRANSPARENT	–		NO CHANGE	–		CLK1 AND 2	H	

REV_____ DATE_____

PROGRAM TABLE_____

CUSTOMER NAME_____ CF (XXXX)_____

PHILIPS DEVICE #_____

CUSTOMER SYMBOLIZED PART #_____

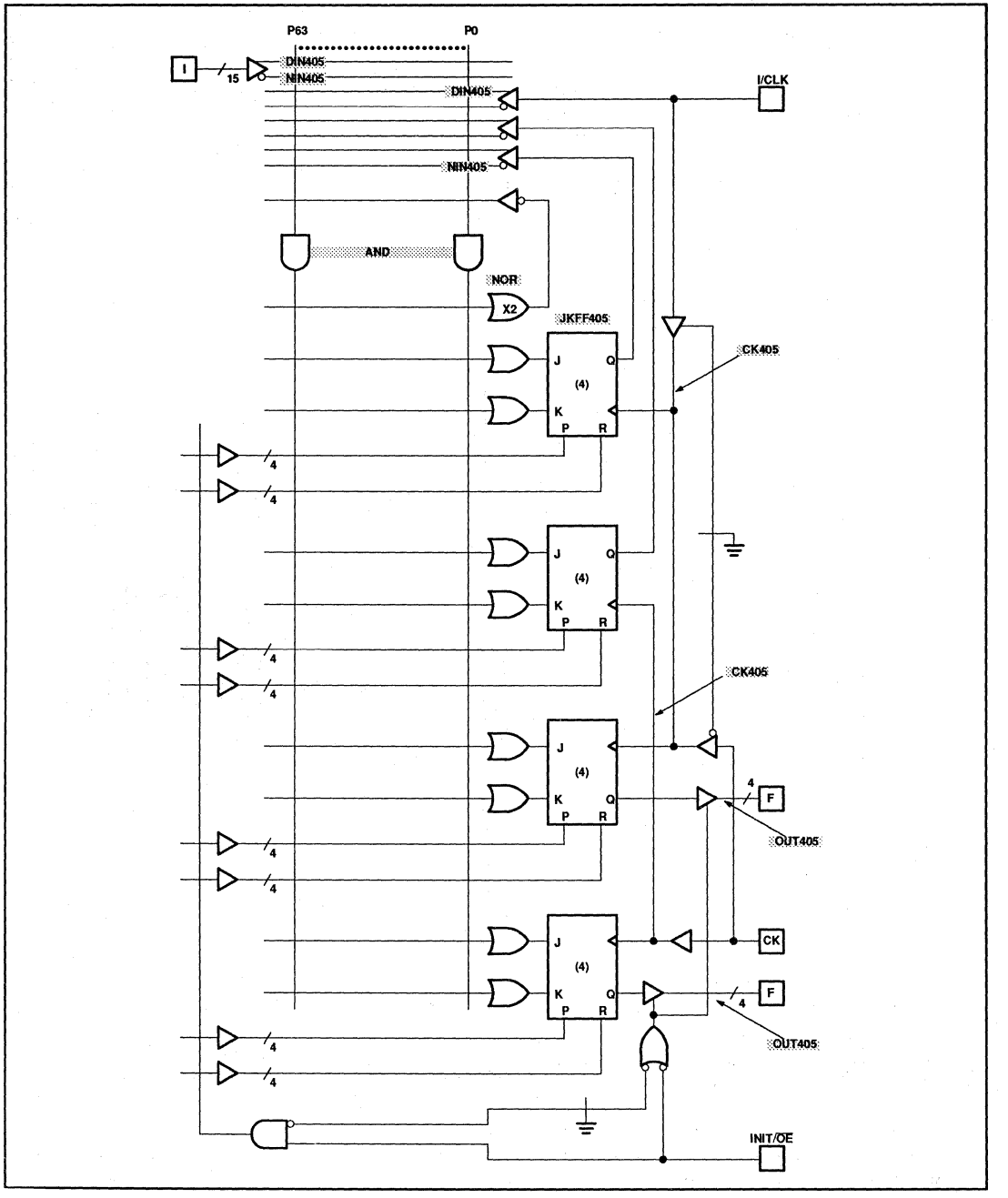
		CLOCK 1/2	INITIALIZATION/OUTPUT ENABLE
	AND		OR
	COMP. ARRAY	INPUT (Im)	PRESENT STATE (Ps)	NEXT STATE (Ns)	OUTPUT (Fr)
	C1 C0	I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0	P7 P6 P5 P4 P3 P2 P1 P0	N7 N6 N5 N4 N3 N2 N1 N0	F7 F6 F5 F4 F3 F2 F1 F0
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					
46					
47					
48					
49					
50					
51					
52					
53					
54					
55					
56					
57					
58					
59					
60					
61					
62					
63					
PIN NO.		20 21 22 23 24 25 26 27 1 2 3 4 5 6 7 8 9			10 11 12 13 15 16 17 18
PIN LABELS					

NOTES:
1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the IN/E and H for the clock option, exists in the table, shown BLANK instead for clarity.
2. Unused Cn Im, and Ps bits are normally programmed Don't Care (—).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (—) for maximum speed.

Programmable logic sequencers (16 × 64 × 8)

PLUS405-37/-45

SNAP RESOURCE SUMMARY DESIGNATIONS



Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

DESCRIPTION

The PLUS405-55 device is a bipolar, programmable state machine of the Mealy type. Both the AND and the OR array are user-programmable. All 64 AND gates are connected to the 16 external dedicated inputs (I0 - I15) and to the feedback paths of the 8 on-chip State Registers (Q_{P0} - Q_{P7}). Two complement arrays support complex IF-THEN-ELSE state transitions with a single product term (input variables C_0, C_1).

All state transition terms can include True, False and Don't Care states of the controlling state variables. All AND gates are merged into the programmable OR array to issue the next-state and next-output commands to their respective registers. Because the OR array is programmable, any one or all of the 64 transition terms can be connected to any or all of the State and Output Registers.

All state (Q_{P0} - Q_{P7}) and output (Q_{F0} - Q_{F7}) registers are edge-triggered, clocked J-K flip-flops, with Asynchronous Preset and Reset options. The PLUS405 architecture provides the added flexibility of the J-K toggle function which is indeterminate on S-R flip-flops. Each register may be individually programmed such that a specific Preset-Reset pattern is initialized when the initialization pin is raised to a logic level "1". This feature allows the state machine to be asynchronously initialized to known internal state and output conditions prior to proceeding through a sequence of state transitions. Upon power-up, all registers are unconditionally preset to "1". If desired, the initialization input pin (INIT) can be converted to an Output Enable ($\overline{OE}$) function as an additional user-programmable feature.

Availability of two user-programmable clocks allows the user to design two independently clocked state machine functions consisting of four state and four output bits each.

Order codes are listed in the Ordering Information Table below.

FEATURES

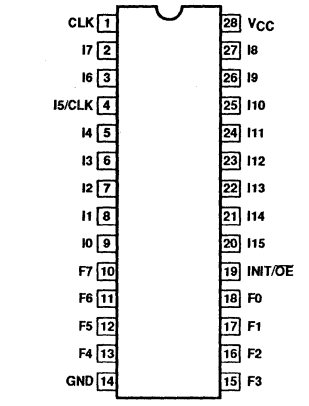
- 66.7MHz minimum guaranteed clock rate
- 55MHz minimum guaranteed operating frequency ($1/(t_{IS1} + t_{CKO1})$)
- Functional superset of PLS105/105A
- Field-programmable (Ti-W fusible link)
- 16 input variables
- 8 output functions
- 64 transition terms
- 8-bit State Register
- 8-bit Output Register
- 2 transition Complement Arrays
- Multiple clocks
- Programmable Asynchronous Initialization or Output Enable
- Power-on preset of all registers to "1"
- "On-chip" diagnostic test mode features for access to state and output registers
- 950mW power dissipation (typ.)
- TTL compatible
- J-K or S-R flip-flop functions
- Automatic "Hold" states
- 3-State outputs

APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator contollers
- Security locking systems
- Counters
- Shift registers

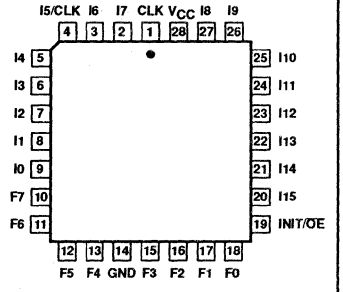
PIN CONFIGURATIONS

N Package



N = Plastic DIP (600mil-wide)

A Package



A = Plastic Leaded Chip Carrier

ORDERING INFORMATION

DESCRIPTION	OPERATING FREQUENCY	ORDER CODE	DRAWING NUMBER
28-Pin Plastic Dual In-Line (600mil-wide)	55MHz ($t_{IS} + t_{CKO}$)	PLUS405–55N	0413B
28-Pin Plastic Leaded Chip Carrier	55MHz ($t_{IS} + t_{CKO}$)	PLUS405–55A	0401F

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CLK1	**Clock:** The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both state and output registers. Pin 1 only clocks P0–3 and F0–3 if Pin 4 is also being used as a clock.	Active-High (H)
2, 3, 5–9, 26–27 20–22	I0 – I4, I7, I6 I8 – I9 I13 – I15	**Logic Inputs:** The 12 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence. True and complement signals are generated via use of "H" and "L".	Active-High/Low (H/L)
4	CLK2	**Logic Input/Clock:** A user programmable function:	
		• **Logic Input:** A 13th external logic input to the AND array, as above.	Active-High/Low (H/L)
		• **Clock:** A 2nd clock for the State Registers P4–7 and Output Registers F4–7, as above. Note that input buffer I5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using Pin 4 as a Clock.	Active-High (H)
23	I12	**Logic/Diagnostic Input:** A 14th external logic input to the AND array, as above, when exercising standard TTL or CMOS levels. When I12 is held at +10V, device outputs F0–F7 reflect the contents of State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
24	I11	**Logic/Diagnostic Input:** A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When I11 is held at +10V, device outputs F0–F7 become direct inputs for State Register bits P0–P7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the State Register bits P0–P7. The contents of each Output Register remains unaltered.	Active-High/Low (H/L)
25	I10	**Logic/Diagnostic Input:** A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When I10 is held at +10V, device outputs F0–F7 become direct inputs for Output Register bits Q0–Q7; a Low-to-High transition on the appropriate clock line loads the values on pins F0–F7 into the Output Register bits Q0–Q7. The contents of each State Register remains unaltered.	Active-High/Low (H/L)
10–13 15–18	F0 – F7	**Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:** Eight device outputs which normally reflect the contents of Output Register Bits Q0–Q7, when enabled. When I12 is held at +10V, F0–F7 = (P0–P7). When I11 is held at +10V, F0–F7 become inputs to State Register bits P0–P7. When I10 is held at +10V, F0–F7 become inputs to Output Register bits Q0–Q7.	Active-High (H)
19	INIT/$\overline{OE}$	**Initialization or Output Enable Input:** A user programmable function:	
		• **Initialization:** Provides an asynchronous preset to logic "1" or reset to logic "0" of all State and Output Register bits, determined individually for each register bit through user programming. INIT overrides Clock, and when held High, clocking is inhibited and F0–F7 and P0–P7 are in their initialization state. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after INIT goes Low. See timing definition for t_{NVCK} and t_{VCK}.	Active-High (H)
		• **Output Enable:** Provides an output enable function to buffers F0–F7 from the Output Registers.	Active-Low (L)

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

TRUTH TABLE [1, 2, 3, 4, 5, 6, 7]

V_{CC}	OPTION INIT	OPTION $\overline{OE}$	I10	I11	I12	CK	J	K	Q_P	Q_F	F
	H		*	*	*	X	X	X	H/L	H/L	Q_F
	L		+10V	X	X	↑	X	X	Q_P	L	L
	L		+10V	X	X	↑	X	X	Q_P	H	H
	L		X	+10V	X	↑	X	X	L	Q_F	L
	L		X	+10V	X	↑	X	X	H	Q_F	H
	L		X	X	+10V	X	X	X	Q_P	Q_F	Q_P
	L		X	X	X	X	X	X	Q_P	Q_F	Q_F
		H	X	X	*	X	X	X	Q_P	Q_F	Hi-Z
+5V		X	+10V	X	X	↑	X	X	Q_P	L	L
		X	+10V	X	X	↑	X	X	Q_P	H	H
		X	X	+10V	X	↑	X	X	L	Q_F	L
		X	X	+10V	X	↑	X	X	H	Q_F	H
		L	X	X	+10V	X	X	X	Q_P	Q_F	Q_P
		L	X	X	X	X	X	X	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	L	Q_P	Q_F	Q_F
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	$\overline{Q_P}$	$\overline{Q_F}$	$\overline{Q_F}$
↑	X	X	X	X	X	X	X	X	H	H	

NOTES:

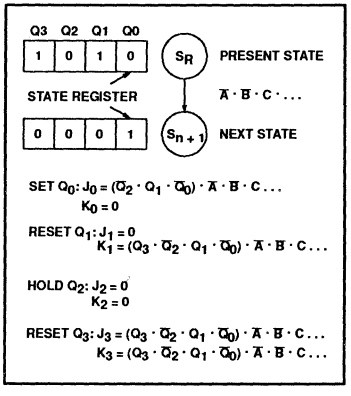
1. Positive Logic:
   S/R (or J/K) = $T_0 + T_1 + T_2 + \ldots T_{63}$
   T_n = (C0, C1) (I0, I1, I2, . . .) (P0, P1, . . . P7)
2. Either Initialization (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
3. ↑ denotes transition from Low-to-High level.
4. * = H or L or +10V
5. X = Don't Care (≤5.5V)
6. H/L implies that either a High or a Low can occur, depending upon user-programmed selection (each State and Output Register individually programmable).
7. When using the F_n pins as inputs to the State and Output Registers in diagnostic mode, the F buffers are 3-Stated and the indicated levels on the output pins are forced by the user.

VIRGIN STATE

A factory-shipped virgin device contains all fusible links intact, such that:

1. INIT/$\overline{OE}$ is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user-programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
2. All transition terms are inactive (0).
3. All S/R (or J/K) flip-flop inputs are disabled (0).
4. The device can be clocked via a Test Array preprogrammed with a standard test pattern.
5. Clock 2 is inactive.

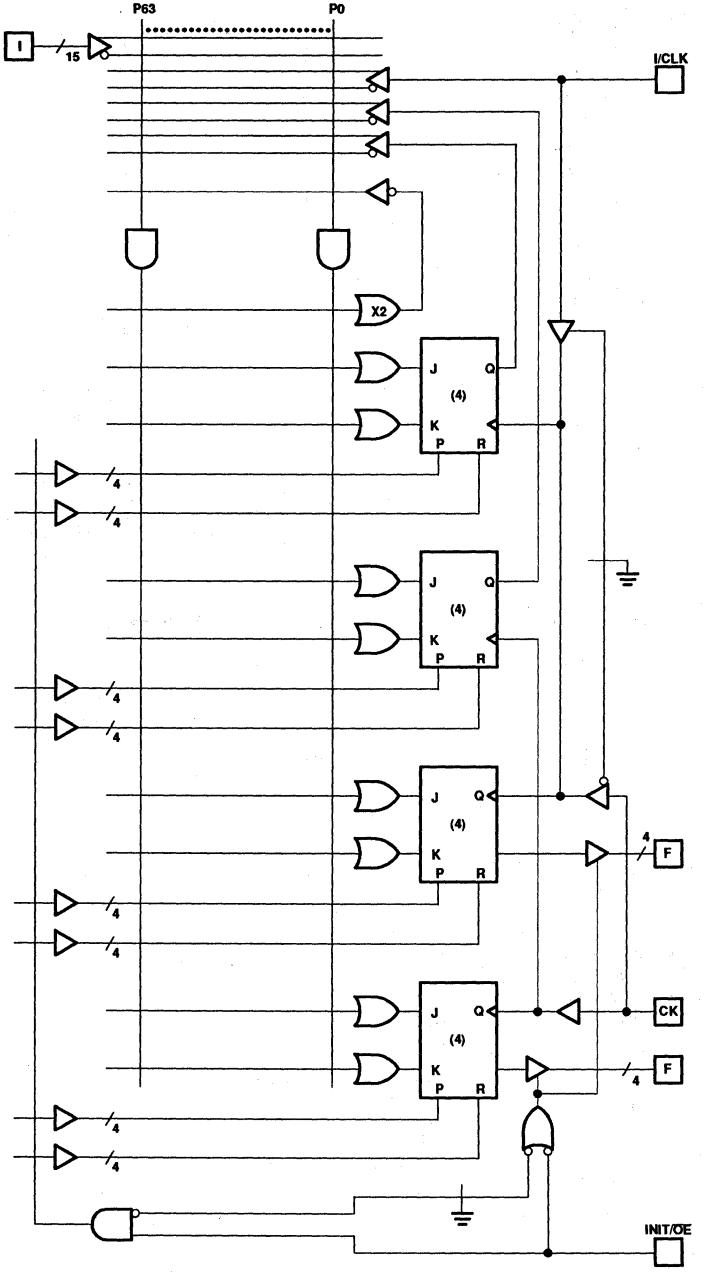
LOGIC FUNCTION



Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

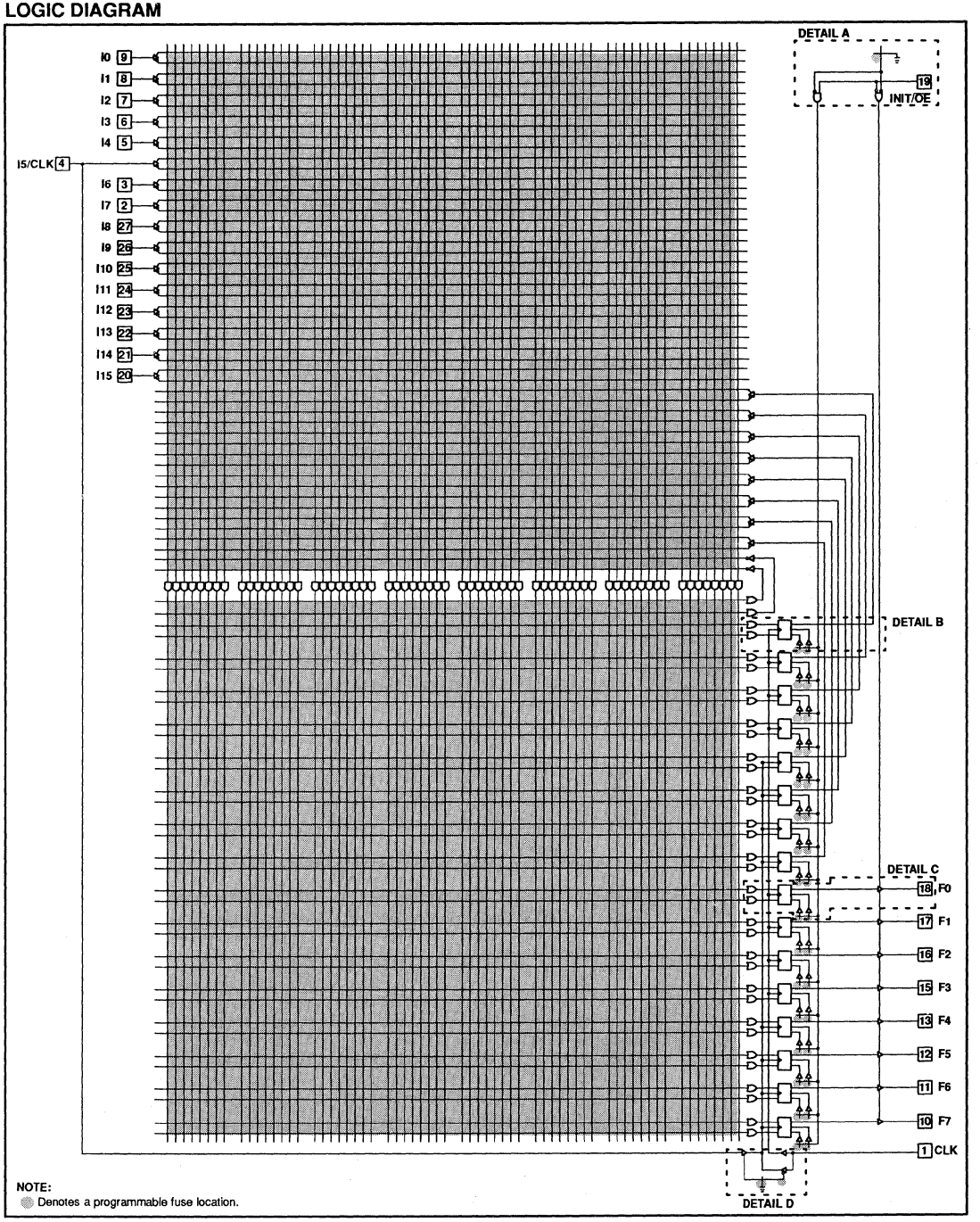
FUNCTIONAL DIAGRAM



Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

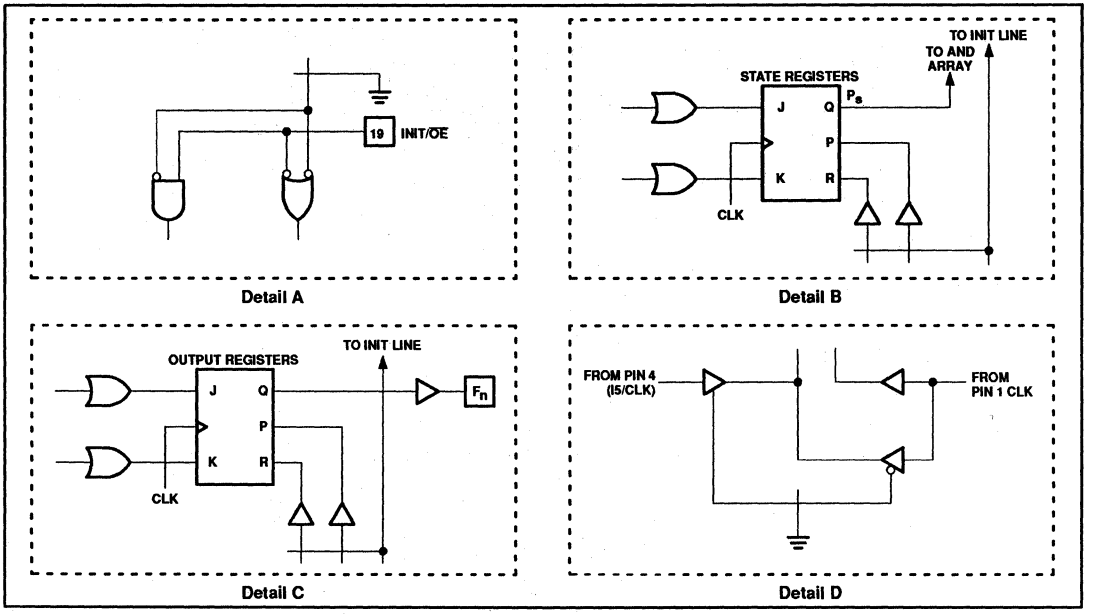
LOGIC DIAGRAM



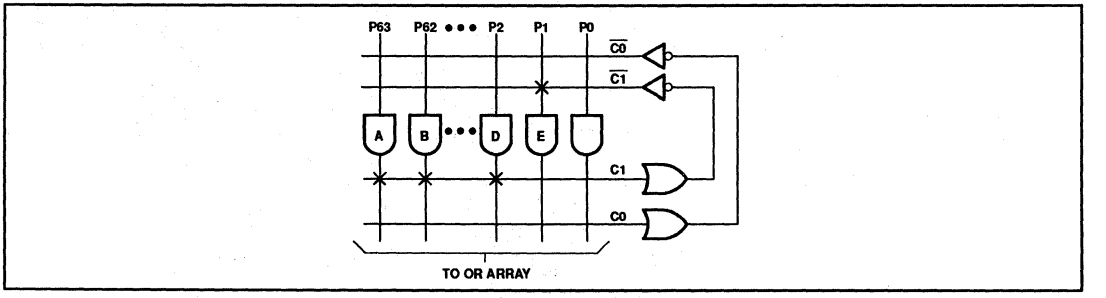
Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

DETAILS FOR REGISTERS FOR PLUS405



COMPLEMENT ARRAY DETAIL



The Complement Array is a special sequencer feature that is often used for detecting illegal states. It is also ideal for generating IF-THEN-ELSE logic statements with a minimum number of product terms.

The concept is deceptively simple. If you subscribe to the theory that the expressions (/A * /B * /C) and $(\overline{A + B + C})$ are equivalent, you will begin to see the value of this single term NOR array.

The Complement Array is a single OR gate with inputs from the AND array. The output of the Complement Array is inverted and fed back to the AND array (NOR). The output of the array will be Low if any one or more of the AND terms connected to it are active (High). If, however, all the connected terms are inactive (Low), which is a classic unknown state, the output of the Complement Array will be High.

Consider the Product Terms A, B and D that represent defined states. They are also connected to the input of the Complement Array. When the condition (not A and not B and not D) exists, the Complement Array will detect this and propagate an Active-High signal to the AND array. This signal can be connected to Product Term E, which could be used in turn to reset the state machine to a known state. Without the Complement Array, one would have to generate product terms for all unknown or illegal states. With very complex state machines, this approach can be prohibitive, both in terms of time and wasted resources.

Note that the PLUS405 sequencers have 2 Complement Arrays which allow the user to design 2 independent Complement functions. This is particularly useful if 2 independent state machines have been implemented on one device.

Note that use of the Complement Array adds an additional delay path through the device. Please refer to the AC Electrical Characteristics for details.

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage[2]						
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IC}	Clamp[3]	V_{CC} = MIN, I_{IN} = –12mA		–0.8	–1.2	V
Output voltage[2]						
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA	2.4			V
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 9.6mA		0.35	0.45	V
Input current						
I_{IH}	High	V_{CC} = MAX, V_{IN} = V_{CC}		<1	30	µA
I_{IL}	Low	V_{CC} = MAX, V_{IN} = 0.45V		–20	–250	µA
Output current						
$I_{O(OFF)}$	Hi-Z state	V_{CC} = MAX, V_{OUT} = 2.7V		1	40	µA
		V_{CC} = MAX, V_{OUT} = 0.45V		–1	–40	µA
I_{OS}	Short circuit [3, 4]	V_{OUT} = 0V	–15		–70	mA
I_{CC}	V_{CC} supply current[5]	V_{CC} = MAX		190	225	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5.0V, V_{IN} = 2.0V		8		pF
C_{OUT}	Output	V_{CC} = 5.0V, V_{OUT} = 2.0V		10		pF

NOTES:
1. All typical values are at V_{CC} = 5V. T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. Duration of short-circuit should not exceed one second.
5. I_{CC} is measured with the INIT/$\overline{OE}$ input grounded, all other inputs at 4.5V and the outputs open.

AC ELECTRICAL CHARACTERISTICS

$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS			UNIT
				MIN	TYP[1]	MAX	
Pulse width							
t_{CKH1}	Clock High; CLK1 (Pin 1)	CK+	CK–	7.5	6		ns
t_{CKL1}	Clock Low; CLK1 (Pin 1)	CK–	CK+	7.5	6		ns
t_{CKP1}	CLK1 Period	CK+	CK+	15	12		ns
t_{CKH2}	Clock High; CLK2 (Pin 4)	CK+	CK–	7.5	6		ns
t_{CKL2}	Clock Low; CLK2 (Pin 4)	CK–	CK+	7.5	6		ns
t_{CKP2}	CLK2 Period	CK +	CK +	15	12		ns
t_{INITH}	Initialization pulse	INIT–	INIT+	14	12		ns
Setup time							
t_{IS1}	Input	Input ±	CK+	10	9		ns
t_{IS2}	Input (through Complement Array)	Input ±	CK+	18	15		ns
t_{VS}	Power-on preset	V_{CC}+	CK–	0	–10		ns
t_{VCK}	Clock resume (after Initialization)	INIT–	CK–	0	–5		ns
t_{NVCK}	Clock lockout (before Initialization)	CK–	INIT–	12	5		ns
Hold time							
t_{IH}	Input	CK+	Input ±	0	–5		ns
Propagation delay							
t_{CKO1}	Clock1 (Pin 1)	CK1+	Output ±		6.5	8	ns
t_{CKO2}	Clock2 (Pin 4)	CK2+	Output ±		7.0	8	ns
t_{OE}[2]	Output Enable	OE–	Output –		6.5	8	ns
t_{OD}[2]	Output Disable	OE+	Output +		6.5	8	ns
t_{INIT}	Initialization	INIT+	Output +		12	18	ns
t_{PPR}	Power-on Preset	V_{CC} +	Output +		0	10	ns

Notes on following page

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

AC ELECTRICAL CHARACTERISTICS (Continued)

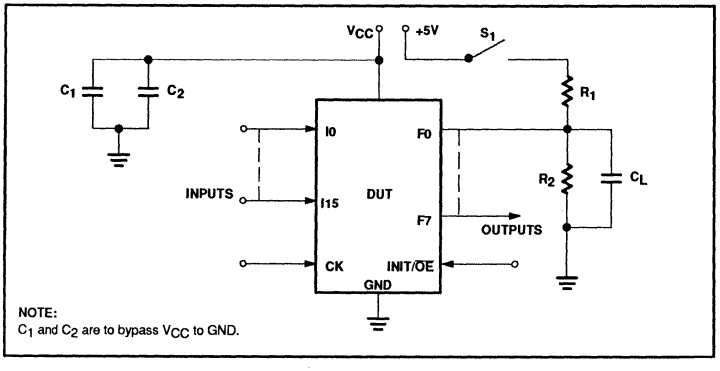
$R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	FROM	TO	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Frequency of operation							
f_{MAX1}	CLK1; (without Complement Array) $\left(\frac{1}{t_{IS1} + t_{CKO1}}\right)$	Input ±	Output ±	55.6	64.5		MHz
f_{MAX2}	CLK2; (without Complement Array) $\left(\frac{1}{t_{IS1} + t_{CKO2}}\right)$	Input ±	Output ±	55.6	62.5		MHz
f_{MAX3}	CLK1; (with Complement Array) $\left(\frac{1}{t_{IS2} + t_{CKO1}}\right)$	Input through Complement Array ±	Output ±	38.5	46.5		MHz
f_{MAX4}	CLK2; (with Complement Array) $\left(\frac{1}{t_{IS2} + t_{CKO2}}\right)$	Input through Complement Array ±	Output ±	38.5	45.5		MHz
f_{MAX5}	Internal feedback without Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{CKL} + t_{CKH}}\right)$	Register Output ±	Register Input ±	66.7	83.3		MHz
f_{MAX6}	Internal feedback with Complement Array (CLK1 or CLK2) $\left(\frac{1}{t_{IS2}}\right)$	Register Output through Complement Array ±	Register Input ±	55.6	66.7		MHz
f_{CLK}	Minimum guaranteed Clock frequency	CK +	CK +	66.7	83.3		MHz

NOTES:

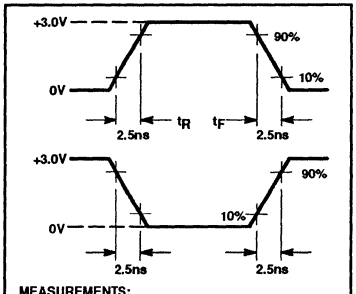
1. All typical values are at $V_{CC} = 5V$, $T_{amb} = +25°C$.
2. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.
3. All propagation delays and setup times are measured and specified under worst case conditions.

TEST LOAD CIRCUIT



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

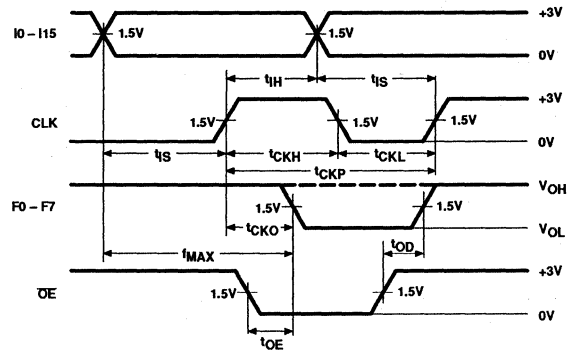
VOLTAGE WAVEFORMS



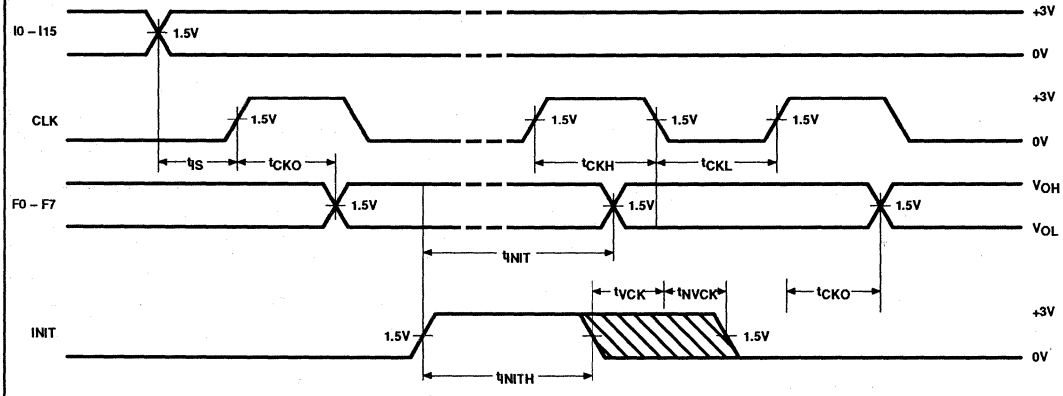
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

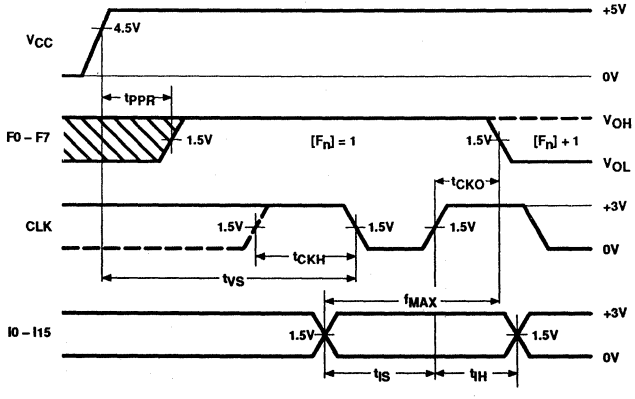
TIMING DIAGRAMS



Sequential Mode

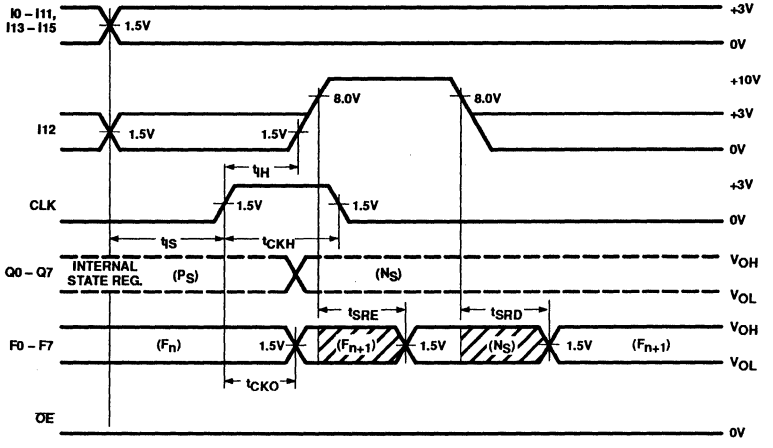


Asynchronous Initialization

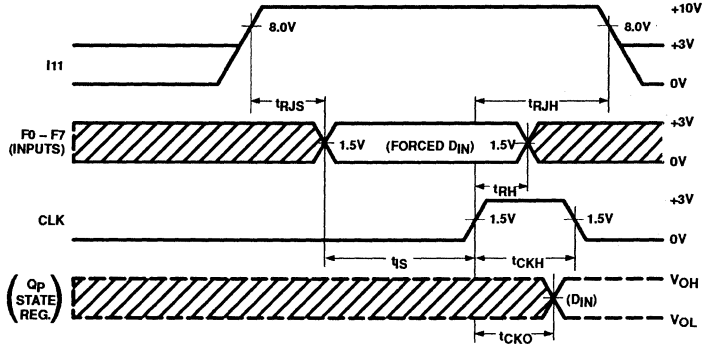


Power-On Preset

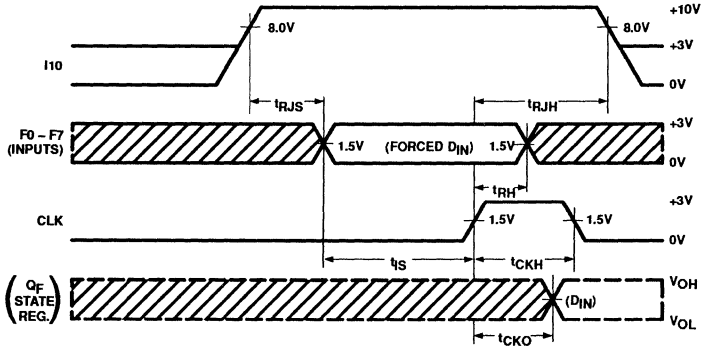
TIMING DIAGRAMS (Continued)



Diagnostic Mode – State Register Outputs



Diagnostic Mode – State Register Input Jam



Diagnostic Mode – Output Register Input Jam

TIMING DEFINITIONS

SYMBOL	PARAMETER
$t_{CKH1, 2}$	Width of input clock pulse.
$t_{CKP1, 2}$	Minimum guaranteed clock period.
t_{IS1}	Required delay between beginning of valid input and positive transition of Clock.
$t_{CKO1, 2}$	Delay between positive transition of Clock and when Outputs become valid (with INIT/$\overline{OE}$ Low).
t_{PPR}	Delay between V_{CC} (after power-on) and when Outputs become preset at "1".
t_{IS2}	Required delay between beginning of valid Input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
t_{RJH}	Required delay between positive transition of clock, and return of input I10, I11 or I12 from Diagnostic Mode (10V).
$f_{MAX1, 2}$	Minimum guaranteed operating frequency; input to output (CLK1 and CLK2).
$f_{MAX3, 4}$	Minimum guaranteed operating frequency; input through Complement Array, to output (CLK1 and CLK2).
f_{MAX5}	Minimum guaranteed internal operating frequency; with internal feedback from state register to state register.

SYMBOL	PARAMETER
f_{MAX6}	Minimum guaranteed internal operating frequency with Complement Array, with internal feedback from state register through Complement Array, to state register.
f_{CLK}	Minimum guaranteed clock frequency (register toggle frequency).
$t_{CKL1, 2}$	Interval between clock pulses.
t_{IH}	Required delay between positive transition of Clock and end of valid Input data.
t_{OE}	Delay between beginning of Output Enable Low and when Outputs become valid.
t_{SRE}	Delay between input I12 transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
t_{RJS}	Required delay between inputs I11, I10 or I12 transition to Diagnostic Mode (10V), and when the output pins become available as inputs.
t_{NVCK}	Required delay between the negative transition of the clock and the negative transition of the Asynchronous Initialization to guarantee that the clock edge is not detected as a valid negative transition.

SYMBOL	PARAMETER
t_{INITH}	Width of initialization input pulse.
t_{VS}	Required delay between V_{CC} (after power-on) and negative transition of Clock preceding first reliable clock pulse.
t_{OD}	Delay between beginning of Output Enable High and when Outputs are in the OFF-state.
t_{INIT}	Delay between positive transition of Initialization and when Outputs become valid.
t_{SRD}	Delay between input I12 transition to Logic mode and when the Outputs reflect the contents of the Output Register.
t_{RH}	Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
t_{VCK}	Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding first reliable clock pulse.

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

LOGIC PROGRAMMING

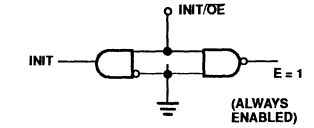
The PLUS405-55 is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors SNAP design software package. ABEL™ and CUPL™ design software packages also support the PLUS405-55 architecture.

All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

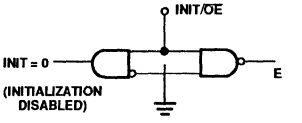
PLUS405-55 logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

To implement the desired logic functions, each logic variable (I, B, P, S, T, etc.) from the logic equations is assigned a symbol. TRUE, COMPLEMENT, PRESET, RESET, OUTPUT ENABLE, INACTIVE, etc., symbols are defined below.

INITIALIZATION/$\overline{OE}$ OPTION – (INIT/$\overline{OE}$)



OPTION	CODE
INITIALIZATION[1]	H

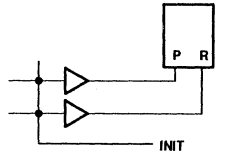


OPTION	CODE
$\overline{OE}$	L

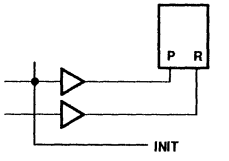
PROGRAMMING THE PLUS405:

The PLUS405 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

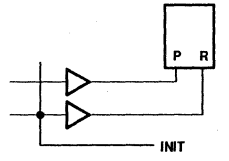
INITIALIZATION OPTION – (INIT)



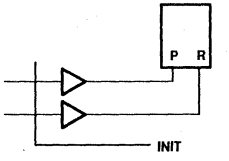
ACTION	CODE
INDETERMINATE[4]	O



ACTION	CODE
PRESET	H

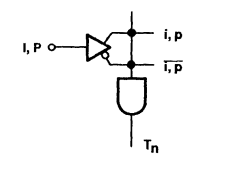


ACTION	CODE
RESET	L

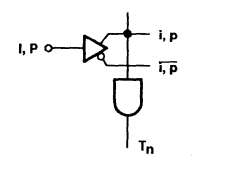


ACTION	CODE
INDETERMINATE[4]	–

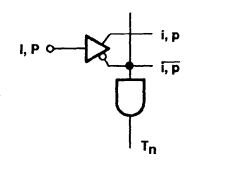
"AND" ARRAY – (I), (P)



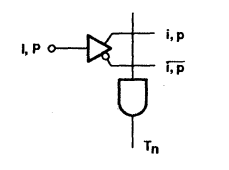
STATE	CODE
INACTIVE[1, 2]	O



STATE	CODE
I, P	H



STATE	CODE
$\overline{I, P}$	L



STATE	CODE
DON'T CARE	–

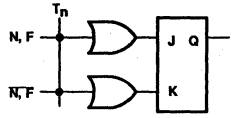
Notes are on next page.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

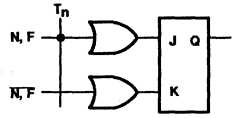
Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

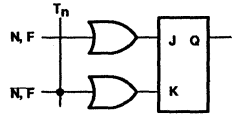
"OR" ARRAY – J-K FUNCTION – (N), (F)



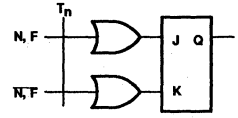
ACTION	CODE
TOGGLE[1,6]	O



ACTION	CODE
SET	H

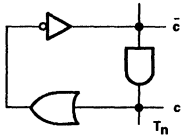


ACTION	CODE
RESET	L

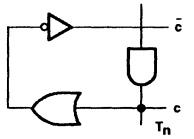


ACTION	CODE
NO CHANGE	–

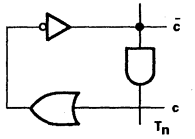
"COMPLEMENT" ARRAY – (C)



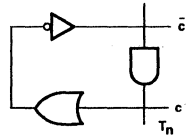
ACTION	CODE
INACTIVE[1,3]	O



ACTION	CODE
GENERATE	A

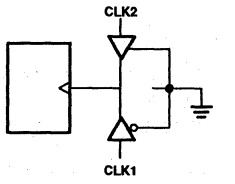


ACTION	CODE
PROPAGATE	•

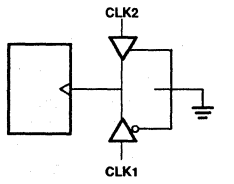


ACTION	CODE
TRANSPARENT	–

CLOCK OPTION – (CLK1/CLK2)



OPTION	CODE
CLK1 ONLY[1]	L



OPTION	CODE
CLK1 and CLK2[5]	H

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.
4. These states are not allowed when using INITIALIZATION option.
5. Input buffer I5 must be deleted from the AND array (i.e., all fuse locations "Don't Care") when using second clock option.
6. A single product term cannot drive more than 8 registers by itself when used in TOGGLE mode.

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

PLUS405 PROGRAM TABLE

AND

INACTIVE	0	INACTIVE	0
I, P	H	GENERATE	A
$\overline{I}$, P	L	PROPAGATE	•
DON'T CARE	–	TRANSPARENT	–

Im, Ps — Cn

OR

INACTIVE OR TOGGLE	0	Ns, Fr
SET	H	
RESET	L	
NO CHANGE	–	

OPTIONS

INIT	H	INIT/$\overline{OE}$
$\overline{OE}$	L	
CLK1 ONLY	L	CLK1/CLK2
CLK1 AND 2	H	

CUSTOMER NAME _____ REV _____ DATE _____

PHILIPS DEVICE # _____ CF (XXXX) _____ PROGRAM TABLE _____

CUSTOMER SYMBOLIZED PART # _____

	CLOCK 1/2	AND																OR															
		COMP. ARRAY		INPUT (Im)														PRESENT STATE (Ps)								INITIALIZATION/OUTPUT ENABLE									
																										NEXT STATE (Ns)							OUTPUT (Fr)			
		C1	C0	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0	P7	P6	P5	P4	P3	P2	P1	P0	N7	N6	N5	N4	N3	N2	N1	N0	F7	F6	F5	F4	F3	F2	F1	F0
0																																									
1																																									
2																																									
3																																									
4																																									
5																																									
6																																									
7																																									
8																																									
9																																									
10																																									
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									
31																																									
32																																									
33																																									
34																																									
35																																									
36																																									
37																																									
38																																									
39																																									
40																																									
41																																									
42																																									
43																																									
44																																									
45																																									
46																																									
47																																									
48																																									
49																																									
50																																									
51																																									
52																																									
53																																									
54																																									
55																																									
56																																									
57																																									
58																																									
59																																									
60																																									
61																																									
62																																									
63																																									
PIN NO.				20	21	22	23	24	25	26	27	2	3	4	5	6	7	8	9															10	11	12	13	15	16	17	18
PIN LABELS																																									

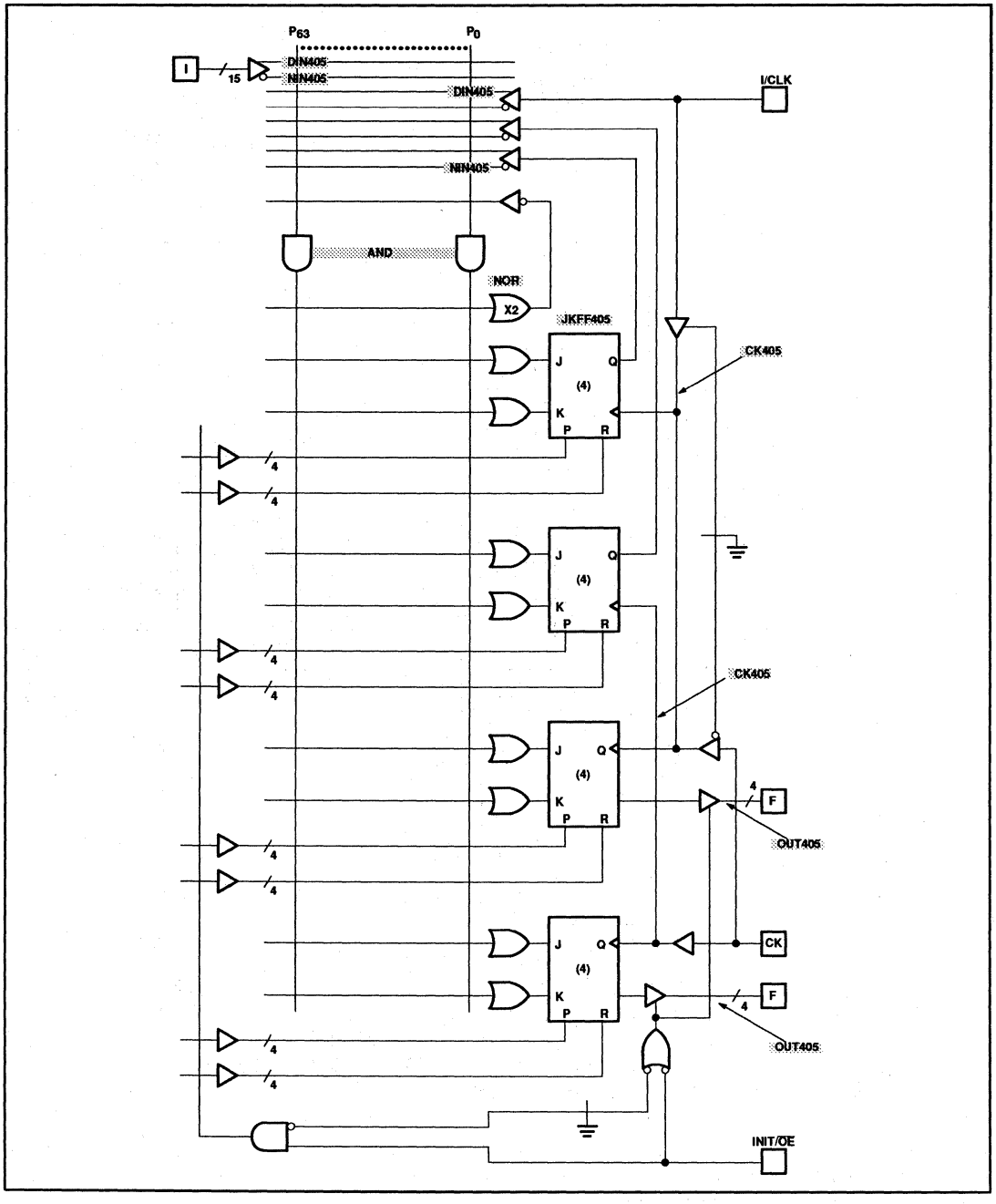
NOTES:

1. The device is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the IN/E and H for the clock option, exists in the table, shown BLANK instead for clarity.
2. Unused Cn Im, and Ps bits are normally programmed Don't Care (—).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (—) for maximum speed.

Programmable logic sequencer (16 × 64 × 8)

PLUS405-55

SNAP RESOURCE SUMMARY DESIGNATIONS



Section 6
Low Volt Devices

CONTENTS

3 Volt zero standby power universal PAL devices

P3C18V8Z35/P3C18V8ZI

DESCRIPTION

The P3C18V8Z is a universal PAL-type device designed to operate specifically in a low voltage environment (3.3V). Per JEDEC, the P3C18V8Z can support a regulated operating supply voltage, 3.0 to 3.6V and an unregulated (battery) operating supply voltage, 2.7 to 3.6V, at 21 and 18 MHz, respectively. The PAL device is available in the commercial temperature range, P3C18V8Z35, and the industrial temperature range, P3C18V8ZI.

These devices offer virtually zero standby power (20μA typical) as well as very low power consumption during operation (23mA worst case in combinatorial configuration). The P3C18V8Z automatically powers down when the inputs or the clock are idle for greater than one full clock cycle. The device will automatically power up from a standby mode once any input or the clock is activated. This input transition detection circuitry makes these devices ideal for power sensitive applications — especially those which are battery operated or backed up.

All the P3C18V8Z devices are available in plastic DIP, PLCC and Plastic Small Outline (SOL) packages. A ceramic DIP with a window for erasure is available for prototyping.

The P3C18V8Z is a two level logic element comprised of 10 inputs, 74 AND gates (logic and control product terms) and 8 Output Macro Cells (OMCs). Each OMC can be configured as a dedicated input, a combinatorial I/O or a registered output with internal feedback. Each OMC has individual direction control (from the AND array) and programmable output polarity. The dedicated clock and OE pins can be configured as inputs for strictly combinatorial applications. Two product terms control the asynchronous Reset and the synchronous Preset functions.

Power up Reset and Register Preload functions have also been incorporated into the P3C18V8Z to facilitate state machine design and testing.

The Output Macro Cell feature of the P3C18V8Z devices provides the flexibility to emulate all 20 pin common PAL and GAL functions, thus providing reduced documentation, inventory and manufacturing costs. The P3C18V8Z is also pin and fuse map compatible with all the Philips 5 Volt P3C18V8Z devices.

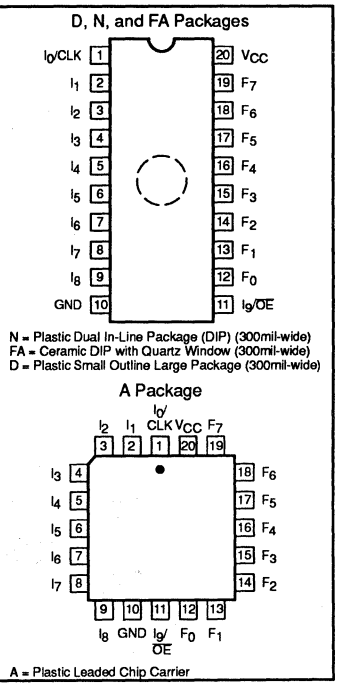
FEATURES

- 20-pin Universal Programmable Array Logic (PAL), operational over low voltage ranges
  - 3.0 to 3.6V (35ns T_{PD}/21 MHz f_{MAX})
  - 2.7 to 3.6V (40ns T_{PD}/18 MHz f_{MAX})
- Virtually zero-standby-power and very low dynamic power
  - 20μA standby (typ.)
  - 0.8 mA/MHz (worst case)
- Functional replacement for Series 16 PALs and GALs
  - Highly flexible Output Macro Cell
- Available in DIP, PLCC and SOL (Small Outline) packages
- High performance EPROM CMOS cell technology
  - 100% testable prior to programming
  - Low cost OTP plastic packages
  - Erasable/reconfigurable (ceramic package)
- Design support provided by most popular third party programmable Logic CAD tools

APPLICATIONS

- Laptop, notebook and palm top computers
- Portable communications equipment
- Battery power/backed instruments
- Industrial automation/control

PIN CONFIGURATIONS



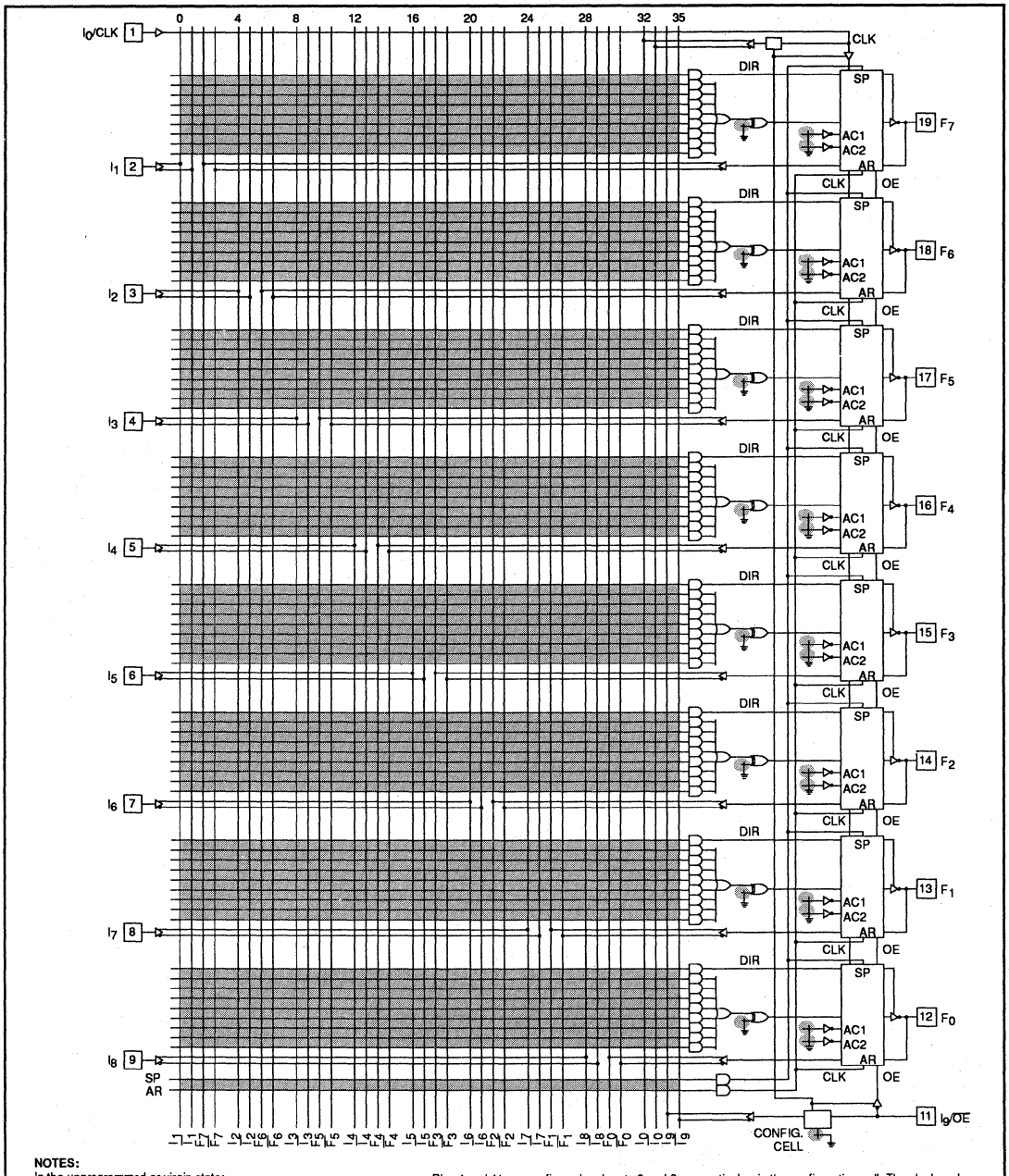
PIN DESCRIPTIONS

I	Dedicated Input
F	Output/Input Macrocell
CLK	Clock Input
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage
GND	Ground

ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DRAWING NUMBER
20-Pin (300mil-wide) Plastic Dual In-Line Package	Commercial	P3C18V8Z35N	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window		P3C18V8Z35FA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier Package		P3C18V8Z35A	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package		P3C18V8Z35D	0172D
20-Pin (300mil-wide) Plastic Dual In-Line Package	Industrial	P3C18V8ZIN	0408B
20-Pin (300mil-wide) Ceramic Dual In-Line Package with quartz window		P3C18V8ZIFA	0584B
20-Pin (350mil square) Plastic Leaded Chip Carrier Package		P3C18V8ZIA	0400E
20-Pin (300mil-wide) Plastic Small Outline Large Package		P3C18V8ZID	0172D

LOGIC DIAGRAM



NOTES:

In the unprogrammed or virgin state:

All cells are in a conductive state.
All AND gate locations are pulled to a logic "0" (Low).
Output polarity is inverting.

Pins 1 and 11 are configured as Inputs 0 and 9, respectively, via the configuration cell. The clock and $\overline{OE}$ functions are disabled.
All output macro cells (OMC) are configured as bidirectional I/O, with the outputs disabled via the direction term.
Denotes a programmable cell location.

3 Volt zero standby power universal PAL devices

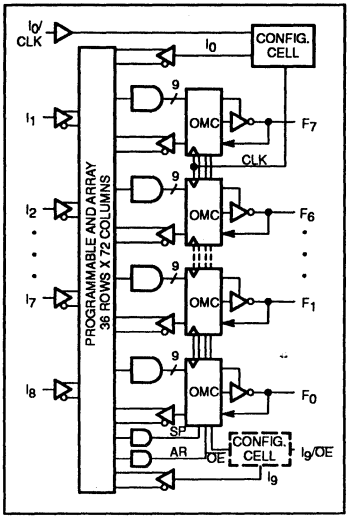
P3C18V8Z35/P3C18V8ZI

PAL DEVICE TO P3C18V8Z OUTPUT PIN CONFIGURATION CROSS REFERENCE

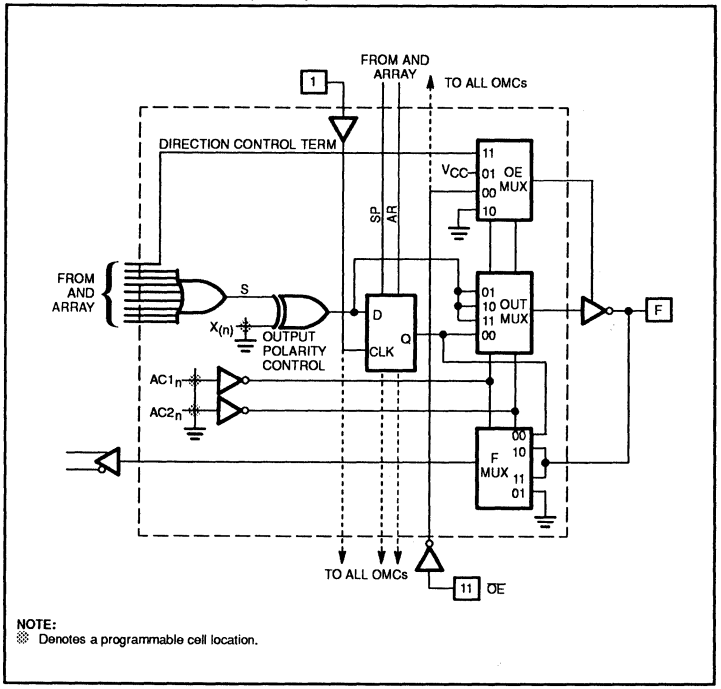
PIN NO.	P3C 18V8Z	16L8 16H8 16P8 16P8	16R4 16RP4	16R6 16RP6	16R8 16RP8	16L2 16H2 16P2	14L4 14H4 14P4	12L6 12H6 12P6	10L8 10H8 10P8
1	I_0/CLK	I	CLK	CLK	CLK	I	I	I	I
19	F7	B	B	B	D	I	I	I	O
18	F6	B	B	D	D	I	I	O	O
17	F5	B	D	D	D	I	O	O	O
16	F4	B	D	D	D	O	O	O	O
15	F3	B	D	D	D	O	O	O	O
14	F2	B	D	D	D	I	O	O	O
13	F1	B	B	D	D	I	I	O	O
12	F0	B	B	B	D	I	I	I	O
11	I_9/$\overline{OE}$	I	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	I	I	I	I

The Philips Semiconductors' state-of-the-art Floating-Gate CMOS EPROM process yields bipolar equivalent performance at less than one-quarter the power consumption. The erasable nature of the EPROM process enables Philips Semiconductors to functionally test the devices prior to shipment to the customer. Additionally, this allows Philips Semiconductors to extensively stress test, as well as ensure the threshold voltage of each individual EPROM cell. 100% programming yield is subsequently guaranteed.

FUNCTIONAL DIAGRAM



OUTPUT MACRO CELL (OMC)



NOTE:
▒ Denotes a programmable cell location.

THE OUTPUT MACRO CELL (OMC)

The P3C18V8Z series devices have 8 individually programmable Output Macro Cells. The 72 AND inputs (or product terms) from the programmable AND array are connected to the 8 OMCs in groups of 9. Eight of the AND terms are dedicated to logic functions; the ninth is for asynchronous direction control, which enables/disables the respective bidirectional I/O pin. Two product terms are dedicated for the Synchronous Preset and Asynchronous Reset functions.

Each OMC can be independently programmed via 16 architecture control bits, $AC1_n$ and $AC2_n$ (one pair per macro cell). Similarly, each OMC has a programmable output polarity control bit (Xn). By configuring the pair of architecture control bits according to the configuration cell table, 4 different configurations may be implemented. Note that the configuration cell is automatically programmed based on the OMC configuration.

DESIGN SECURITY

The P3C18V8Z series devices have a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

CONFIGURATION CELL

A single configuration cell controls the functions of Pins 1 and 11. Refer to Functional Diagram. When the configuration cell is programmed, Pin 1 is a dedicated clock and Pin 11 is dedicated for output enable. When the configuration cell is unprogrammed, Pins 1 and 11 are both dedicated inputs. Note that the output enable for all registered OMCs is common—from Pin 11 only. Output enable control of the bidirectional I/O OMCs is provided from the AND array via the direction product term.

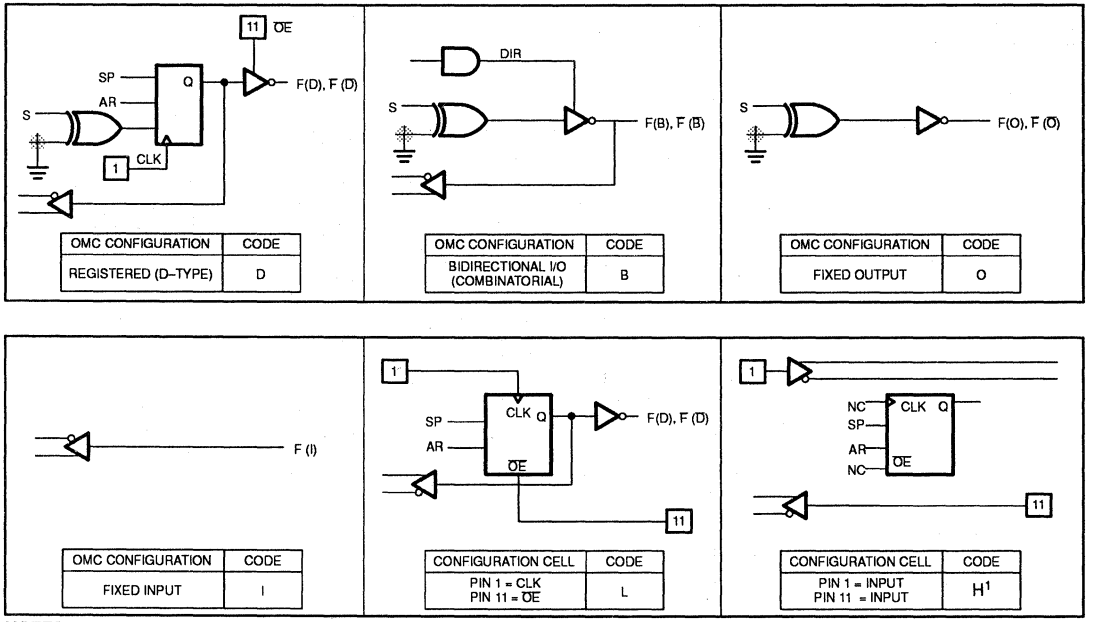
If any one OMC is configured as registered, the configuration cell will be automatically configured (via the design software) to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively. If none of the OMCs are registered, the configuration cell will be programmed such that Pins 1 and 11 are dedicated inputs. The programming codes are as follows:

Pin 1 = CLK, Pin 11 = $\overline{OE}$	L
Pin 1 and Pin 11 = Input	H

FUNCTION	CONTROL CELL CONFIGURATIONS			COMMENTS
	$AC1_1$	$AC2_N$	CONFIG. CELL	
Registered mode	Programmed	Programmed	Programmed	Dedicated clock from Pin 1. $\overline{OE}$ Control for all registerd OMCs from Pin 11 only.
Bidirectional I/O mode	Unprogrammed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. 3-State control from AND array only.
Fixed input mode	Unprogrammed	Programmed	Unprogrammed	Pins 1 and 11 are dedicated inputs.
Fixed output mode	Programmed	Unprogrammed	Unprogrammed	Pins 1 and 11 are dedicated inputs. The feedback path (via F_{MUX}) is disabled.

NOTE:
4. This is the virgin state as shipped from the factory.

ARCHITECTURE CONTROL—AC1 and AC2



OMC CONFIGURATION	CODE
REGISTERED (D–TYPE)	D

OMC CONFIGURATION	CODE
BIDIRECTIONAL I/O (COMBINATORIAL)	B

OMC CONFIGURATION	CODE
FIXED OUTPUT	O

OMC CONFIGURATION	CODE
FIXED INPUT	I

CONFIGURATION CELL	CODE
PIN 1 = CLK PIN 11 = $\overline{OE}$	L

CONFIGURATION CELL	CODE
PIN 1 = INPUT PIN 11 = INPUT	H[1]

NOTES:
A factory shipped unprogrammed device is configured such that:
1. This configuration cannot be used if any OMCs are configured as registered (Code = D). The configuration cell will be automatically configured to ensure that the clock and output enable functions are enabled on Pins 1 and 11, respectively, if any one OMC is programmed as registered.
   * All AND gates are pulled to a logic "0" (Low).
   * Output polarity is inverting.
   * Pins 1 and 11 are configured as inputs 0 and 9. The clock and OE functions are disabled.
   * All Output Macro Cells (OMCs) are configured as bidirectional I/O, with the outputs disabled via the direction term.

ABSOLUTE MAXIMUM RATINGS[1]

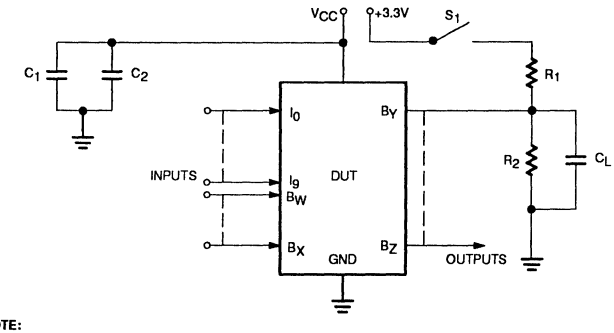
SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	–0.5 to +6	V_{DC}
V_{CC}	Operating supply voltage	2.7 to 3.6	V_{DC}
V_{IN}	Input voltage	–0.5 to V_{CC} +0.5	V_{DC}
V_{OUT}	Output voltage	–0.5 to V_{CC} +0.5	V_{DC}
$\Delta t/\Delta V$	Input/clock transition rise or fall[2]	200	ns/V maximum
I_{IN}	Input currents	–10 to +10	mA
I_{OUT}	Output currents	+24	mA
T_{amb}	Operating temperature range	–40 to +85 (Industrial) 0 to +75 (Commercial)	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTES:
1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
2. All digital circuits can oscillate or trigger prematurely when input rise and fall times are very long. When the input signal to a device is at or near the switching threshold, noise on the line will be amplified and can cause oscillation which, if the frequency is low enough, can cause subsequent stages to switch and give erroneous results. For this reason, external Schmitt-triggers are recommended if rise/fall times are likely to exceed 200ns at V_{CC} = 3.6V.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

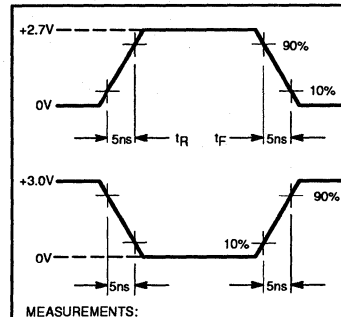
AC TEST CONDITIONS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

R1 = 200Ω R2 = 390Ω

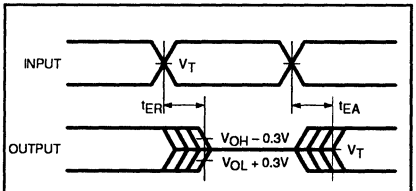
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

SWITCHING WAVEFORM



Input to Output Disable/Enable

DC ELECTRICAL CHARACTERISTICS

$2.7V \leq V_{CC} \leq 3.6$ and $3.0V \leq V_{CC} \leq 3.6$ ranges
Commercial = $0°C \leq T_{amb} \leq +75°C$
Industrial = $-40°C \leq T_{amb} \leq +85°C$

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			MIN	TYP[1]	MAX	
Input voltage						
V_{IL}	Low	V_{CC} = MIN	–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX	2.0		V_{CC} + 0.3	V
Output voltage[2]						
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 20µA V_{CC} = MIN, I_{OL} = 24mA			0.100 0.500	V V
V_{OH}	High	V_{CC} = 3.0, I_{OH} = –3.2mA V_{CC} = 3.0, I_{OH} = –20µA V_{CC} = 2.7, I_{OH} = –1.6µA	V_{CC} – 0.6 V_{CC} – 0.3 V_{CC} – 0.3			V V
Input current						
I_{IL}	Low[5]	V_{IN} = GND			–5	µA
I_{IH}	High	V_{IN} = V_{CC}			5	µA
Output current						
$I_{O(OFF)}$	Hi–Z state	V_{OUT} = V_{CC} V_{OUT} = GND			10 –10	µA µA
I_{OS}	Short-circuit[3]	V_{OUT} = GND			–130	mA
I_{CC}	V_{CC} supply current (Standby)	V_{CC} = MAX, V_{IN} = 0 or V_{CC}[6]		20	60	µA
I_{CC}/f	V_{CC} supply current (Active)[4]	V_{CC} = MAX			0.8	mA/MHz
Capacitance						
C_I	Input	V_{CC} = 5V V_{IN} = 2.0V		12		pF
C_B	I/O	V_B = 2.0V		15		pF

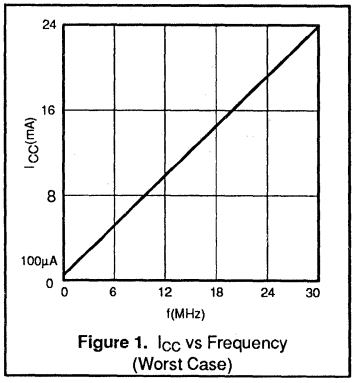


Figure 1. I_{CC} vs Frequency (Worst Case)

NOTES:
1. All typical values are at V_{CC} = 3.3V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Duration of short–circuit should not exceed one second. Test one at a time.
4. Measured with all outputs swtiching.
5. I_{IL} for Pin 1 (I_O/CLK) is ± 10µA with V_{IN} = 0.4V.
6. V_{IN} includes CLK and $\overline{OE}$ if applicable.

AC ELECTRICAL CHARACTERISTICS

3.0V ≤ V_{CC} ≤ 3.6V range; R_2 = 390Ω
Commercial = 0°C ≤ T_{amb} ≤ +75°C
Industrial = −40°C ≤ T_{amb} ≤ +85°C

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION[1] C_L (pF)	P3C18V8Z35 (Commercial) MIN	P3C18V8Z35 (Commercial) MAX	P3C18V8ZI (Industrial) MIN	P3C18V8ZI (Industrial) MAX	UNIT
Pulse width									
t_{CKP}	Clock period (Minimum t_{IS} + t_{CKO})	CLK +	CLK +	50	47		57		ns
t_{CKH}	Clock width High	CLK +	CLK −	50	20		25		ns
t_{CKL}	Clock width Low	CLK −	CLK +	50	20		25		ns
t_{ARW}	Async reset pulse width	I ±, F±	I ∓, F ∓		35		40		ns
Hold time									
t_{IH}	Input or feedback data hold time	CLK +	Input ±	50	0		0		ns
Setup time									
t_{IS}	Input or feedback data setup time	I ±, F±	CLK +	50	25		30		ns
Propagation delay									
t_{PD}	Delay from input to active output	I ±, F±	F±	50		35		40	ns
t_{CKO}	Clock High to output valid access Time	CLK +	F±	50		15		20	ns
t_{OE1}	Product term enable to outputs off	I ±, F±	F±	50		40		45	ns
t_{OD1}	Product term disable to outputs off	I ±, F±	F±	5		35		40	ns
t_{OD2}	Pin 11 output disable High to outputs off	OE −	F±	5		25		30	ns
t_{OE2}	Pin 11 output enable to active output	OE +	F±	50		30		35	ns
t_{ARD}	Async reset delay	I ±, F±	F +			35		40	ns
t_{ARR}	Async reset recovery time	I ±, F±	CLK +		25		30		ns
t_{SPR}	Sync preset recovery time	I ±, F±	CLK +		25		30		ns
t_{PPR}	Power-up reset	V_{CC} +	F +			35		40	ns
Frequency of operation									
f_{MAX}	Maximum frequency	$1/(t_{IS} + t_{CKO})$		50		25		20	MHz

NOTES:
1. Refer also to AC Test Conditions. (Test Load Circuit)

AC ELECTRICAL CHARACTERISTICS

$2.7V \leq V_{CC} \leq 3.6V$ range; $R_2 = 390\Omega$
Commercial = $0°C \leq T_{amb} \leq +75°C$
Industrial = $-40°C \leq T_{amb} \leq +85°C$

SYMBOL	PARAMETER	FROM	TO	TEST CONDITION[1] C_L (pF)	P3C18V8Z35 (Commercial) MIN	P3C18V8Z35 (Commercial) MAX	P3C18V8ZI (Industrial) MIN	P3C18V8ZI (Industrial) MAX	UNIT
Pulse width									
t_{CKP}	Clock period (Minimum $t_{IS} + t_{CKO}$)	CLK +	CLK +	50	57		57		ns
t_{CKH}	Clock width High	CLK +	CLK –	50	25		30		ns
t_{CKL}	Clock width Low	CLK –	CLK +	50	25		30		ns
t_{ARW}	Async reset pulse width	I ±, F±	I $\overline{+}$, F $\overline{+}$		40		45		ns
Hold time									
t_{IH}	Input or feedback data hold time	CLK +	Input ±	50	0		0		ns
Setup time									
t_{IS}	Input or feedback data setup time	I ±, F±	CLK +	50	30		35		ns
Propagation delay									
t_{PD}	Delay from input to active output	I ±, F±	F±	50		40		45	ns
t_{CKO}	Clock High to output valid access Time	CLK +	F±	50		17		22	ns
t_{OE1}	Product term enable to outputs off	I ±, F±	F±	50		45		50	ns
t_{OD1}	Product term disable to outputs off	I ±, F±	F±	5		40		45	ns
t_{OD2}	Pin 11 output disable High to outputs off	OE –	F±	5		30		35	ns
t_{OE2}	Pin 11 output enable to active output	OE +	F±	50		35		40	ns
t_{ARD}	Async reset delay	I ±, F±	F +			40		45	ns
t_{ARR}	Async reset recovery time	I ±, F±	CLK +		30		35		ns
t_{SPR}	Sync preset recovery time	I ±, F±	CLK +		30		35		ns
t_{PPR}	Power-up reset	V_{CC} +	F +			40		45	ns
Frequency of operation									
f_{MAX}	Maximum frequency	$1/(t_{IS} + t_{CKO})$		50		21		17	MHz

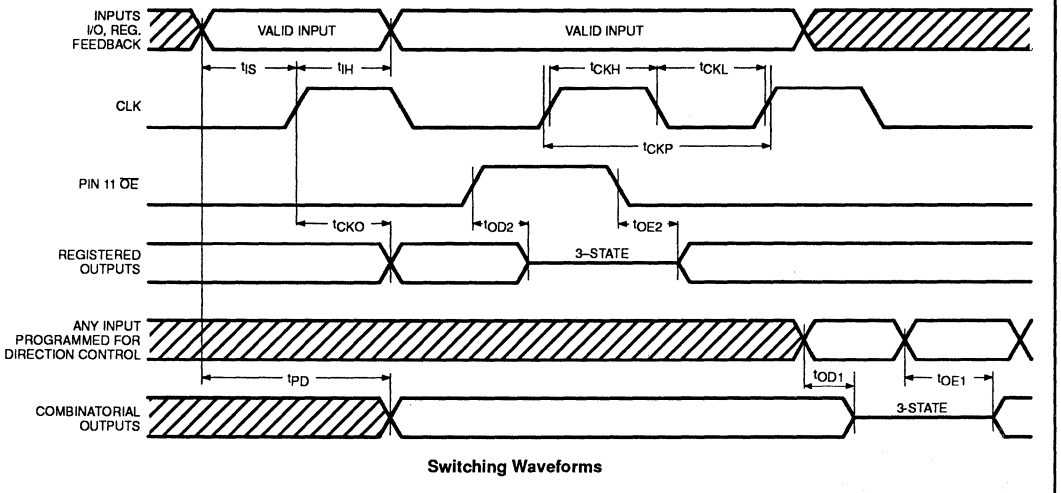
NOTES:
1. Refer also to AC Test Conditions. (Test Load Circuit)

POWER-UP RESET

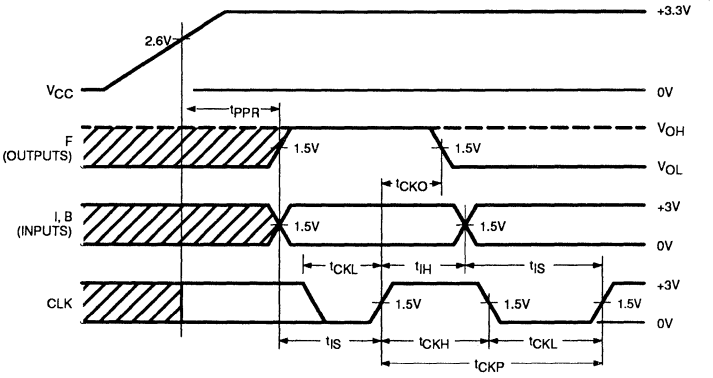
In order to facilitate state machine design and testing, a power-up reset function has been incorporated in the P3C18V8Z. All internal registers will reset to Active-Low (logical "0") after a specified period of time (t_{PPR}).

Therefore, any OMC that has been configured as a registered output will always produce an Active-High on the associated output pin because of the inverted output buffer. The internal feedback (Q) of a registered OMC will also be set Low. The programmed polarity of OMC will not affect the Active-High output condition during a system power-up condition.

TIMING DIAGRAMS



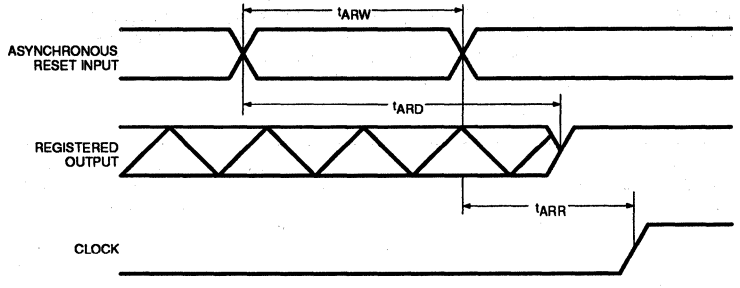
Switching Waveforms



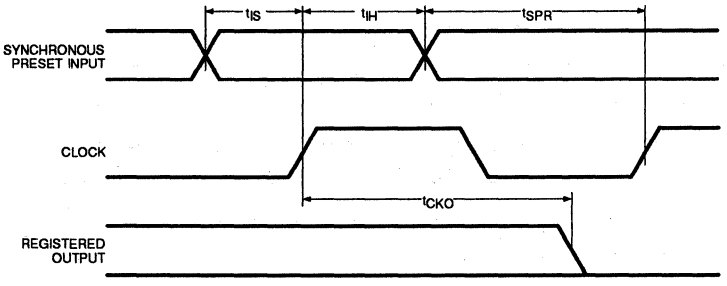
NOTE:
Diagram presupposes that the outputs (F) are enabled. The reset occurs regardless of the output condition (enabled or disabled).

Power-Up Reset

TIMING DIAGRAMS (Continued)



Asynchronous Reset



Synchronous Preset

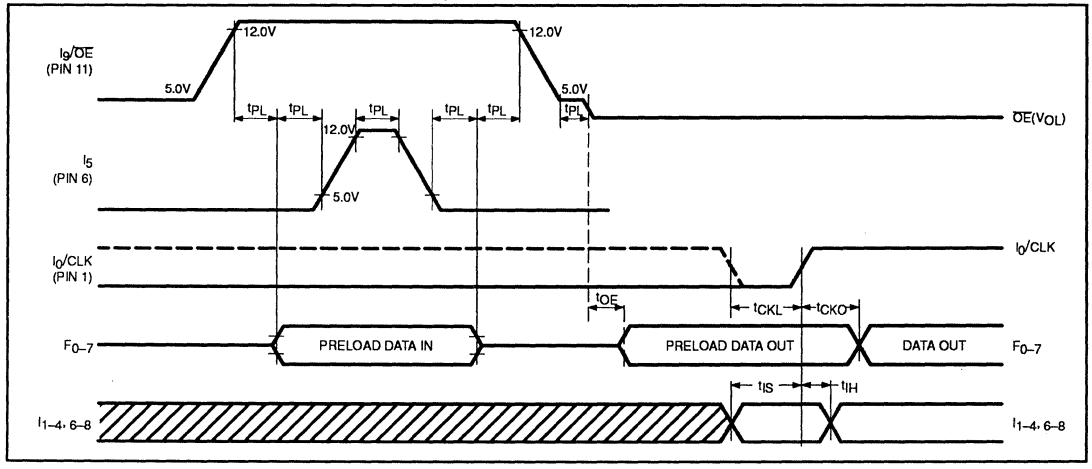
REGISTER PRELOAD FUNCTION (DIAGNOSTIC MODE ONLY)

In order to facilitate the testing of state machine/controller designs, a diagnostic mode register preload feature has been incorporated into the P3C18V8Z series device. This feature enables the user to load the registers with predetermined states while a super voltage is applied to Pins 11 and 6 ($I_9/\overline{OE}$ and I_5). (See diagram for timing and sequence.)

To read the data out, Pins 11 and 6 must be returned to normal TTL levels. The outputs, F_{0-7}, must be enabled in order to read data out. The Q outputs of the registers will reflect data in as input via F_{0-7} during preload. Subsequently, the register Q output via the feedback path will reflect the data in as input via F_{0-7}.

Refer to the voltage waveform for timing and voltage references. t_{PL} = 10μsec.

REGISTER PRELOAD (DIAGNOSTIC MODE)



LOGIC PROGRAMMING

The P3C18V8Z series is fully supported by industry standard (JEDEC compatible) PLD CAD tools, including Philips Semiconductors' SNAP design software package. ABEL™ and CUPL™ 90 design software packages also support the P3C18V8Z architecture.

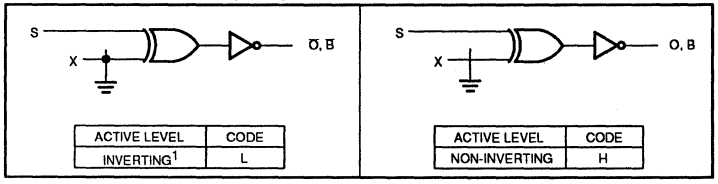
All packages allow Boolean and state equation entry formats. SNAP, ABEL and CUPL also accept, as input, schematic capture format.

P3C18V8Z logic designs can also be generated using the program table entry format, which is detailed on the following pages. This program table entry format is supported by SNAP only.

With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table. Similarly, various OMC configurations are implemented by programming the Architecture Control bits AC1 and AC2. Note that the configuration cell is automatically programmed based on the OMC configuration.

In this table, the logic state of variables I, P and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

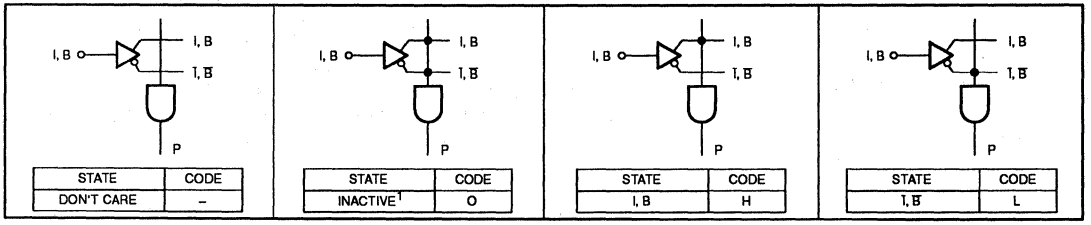
OUTPUT POLARITY – (O, B)



ACTIVE LEVEL	CODE
INVERTING[1]	L

ACTIVE LEVEL	CODE
NON-INVERTING	H

"AND" ARRAY – (I, B)



STATE	CODE
DON'T CARE	–

STATE	CODE
INACTIVE[1]	O

STATE	CODE
I, B	H

STATE	CODE
$\bar{I}, \bar{B}$	L

NOTE:
1. A factory shipped unprogrammed device is configured such that all cells are in a conductive state.

ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the P3C18V8Z Series devices are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lighting could erase a typical P3C18V8Z in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the P3C18V8Z is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the P3C18V8Z is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

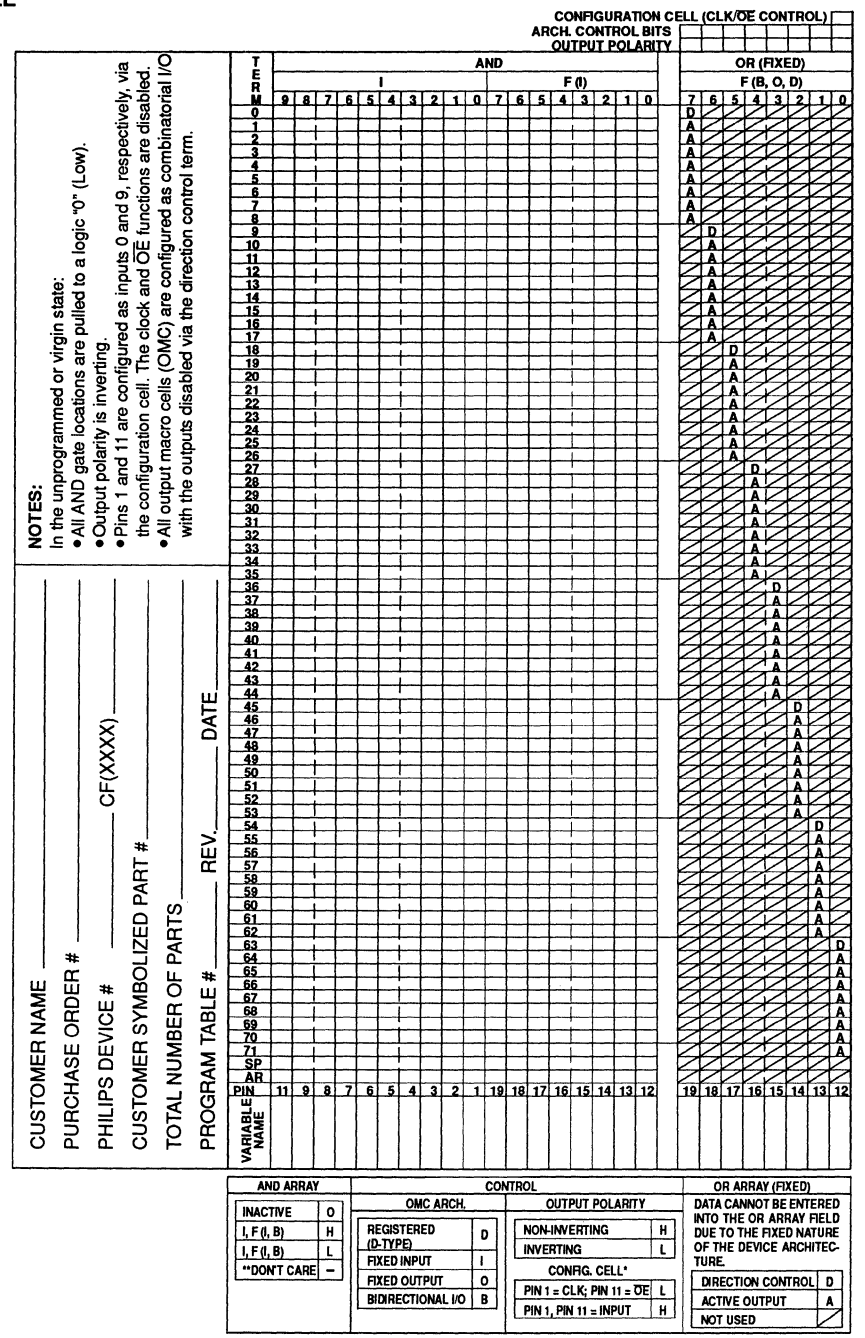
The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

ABEL is a trademark of Data I/O Corp.
CUPL is a trademark of Logical Devices, Inc.

3 Volt zero standby power universal PAL devices

P3C18V8Z35/P3C18V8ZI

PROGRAM TABLE



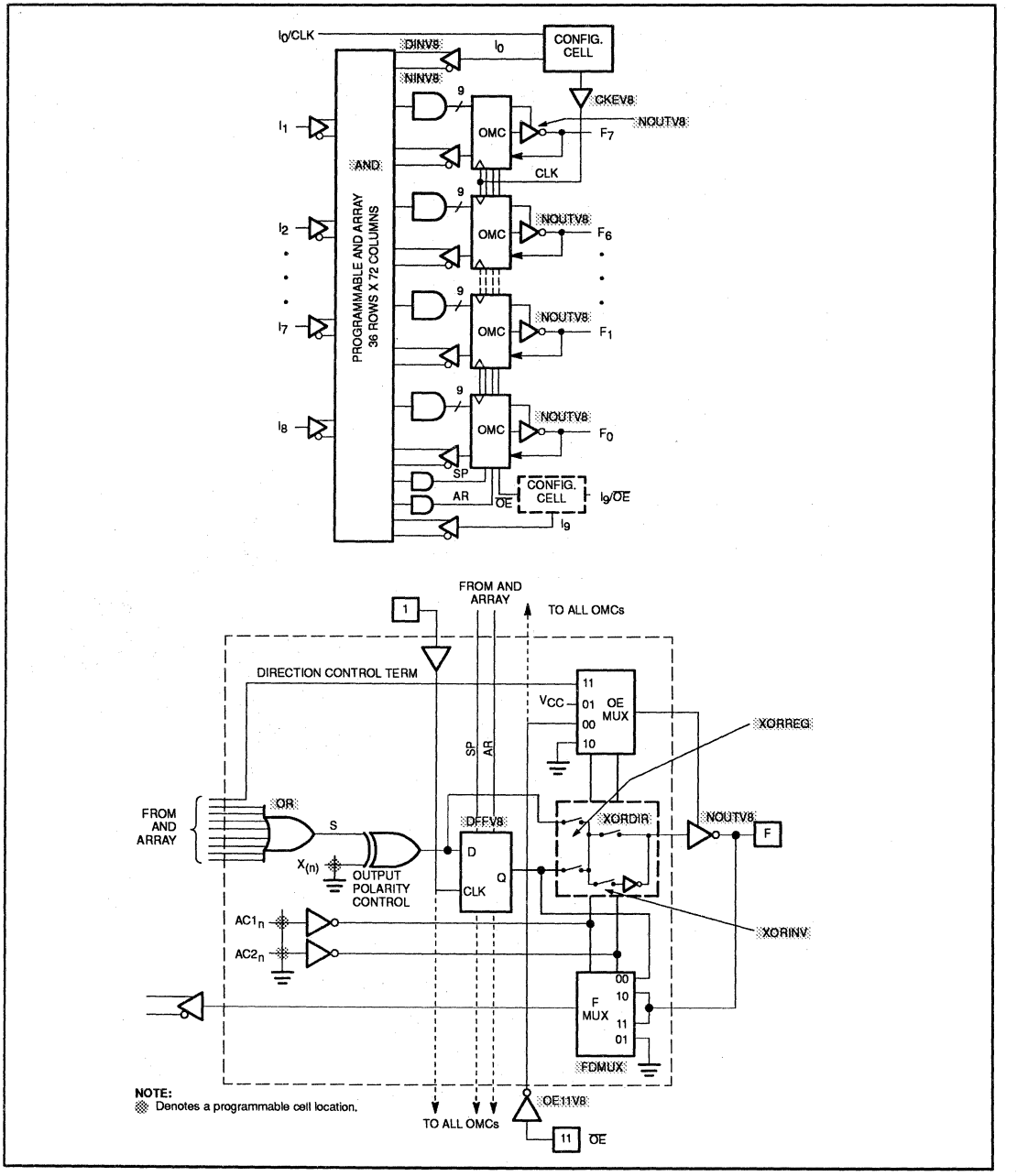
AND ARRAY	
INACTIVE	0
I, F (I, B)	H
I, F (I, B)	L
"DON'T CARE	–

CONTROL			
OMC ARCH.		OUTPUT POLARITY	
REGISTERED (D-TYPE)	D	NON-INVERTING	H
FIXED INPUT	I	INVERTING	L
FIXED OUTPUT	O	CONFIG. CELL*	
BIDIRECTIONAL I/O	B	PIN 1 = CLK; PIN 11 = OE	L
		PIN 1, PIN 11 = INPUT	H

OR ARRAY (FIXED)	
DATA CANNOT BE ENTERED INTO THE OR ARRAY FIELD DUE TO THE FIXED NATURE OF THE DEVICE ARCHITECTURE.	
DIRECTION CONTROL	D
ACTIVE OUTPUT	A
NOT USED	

* THE CONFIGURATION CELL IS AUTOMATICALLY PROGRAMMED BASED ON THE OMC ARCHITECTURE.
** FOR SP, AR: "–" IS NOT ALLOWED.

SNAP RESOURCE SUMMARY DESIGNATIONS



3 Volt BiCMOS Versatile GAL–type PLD — LVT16V8–7

DESCRIPTION

The LVT16V8–7 is a V–type GAL device designed to operate over the 3.0 to 3.6 volt range. This versatile device is fabricated using the BiCMOS process which produces superior performance, low noise and reduced ground bounce. The reduction from 5V to 3.3V also dramatically reduces power to less than 0.5 watts (worst case).

This industry standard device is ideal for high performance systems which have been designed to operate with 3.3V ± 0.3V power supplies, as well as systems which are operating with dual supplies (5.0V and 3.3V). The LVT16V8–7 can accept both 3.3 and 5.0V input levels without the need for level translators. Both the inputs and I/O have high state reverse current flow protection to insure that the outputs are not damaged if the 3V LVT16V8 is interfaced with 5V devices.

The LVT16V8–7 is designed with metastable hardened flip–flops so that the outputs can never display a metastable state due to set up or hold time violations. If set up or hold times are violated, the outputs will not glitch or display a metastable state (however propagation delays may extend).

Active bus-hold circuitry is provided to eliminate the need for external resistors to hold unused or floating inputs at valid logic levels.

The LVT16V8's flexible architecture supports a wide variety of high performance applications: counters, shift registers, address decoders, state machines, multiplexers and random logic collection.

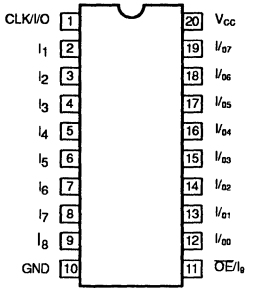
The LVT16V8–7 is identical, in function and fuse map, to other industry standard EEPROM and EPROM 16V8 devices. Development and programming support are offered by Philips and other third party vendors.

FEATURES

- Advanced low voltage BiCMOS process technology
- Ultra high performance over the 3.0 to 3.6 voltage range
  - 7.5ns T_{PD}
  - 5.0ns T_{IS}
  - 5.0ns T_{CKO}
  - 110 MH_Z F_{MAX} (internal feedback)
  - 143 MH_Z clock rate
- Low power dissipation
  - 300mW typical
- 5V compatible inputs and I/O
- Exceptional noise immunity and low ground bounce
- Live insertion/extraction
- Bus-hold data inputs eliminate the need for external pull up resistors.
- Wide package availability; DIP, PLCC, SOL
- Metastable hardened Flip-Flops
- Architectural Flexibility
  - Emulates all 20 pin PAL devices
  - Up to 16 inputs and 8 outputs
  - Independently programmable I/O macrocells (4 configurations)
  - Independently programmable output polarity
  - Product term output enable for combinatorial functions
  - Register Preload and Power Up reset of all registers
- Development and programming support
  - Third party software and programmers
  - Philips SNAP development software

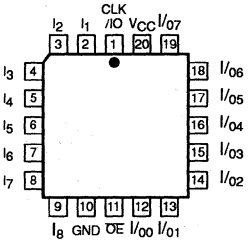
PIN CONFIGURATIONS

D and N Packages



N = Plastic DIP (300mil-wide)
D = Plastic SOL (300mil-wide)

A Package



A = Plastic Leaded Chip Carrier

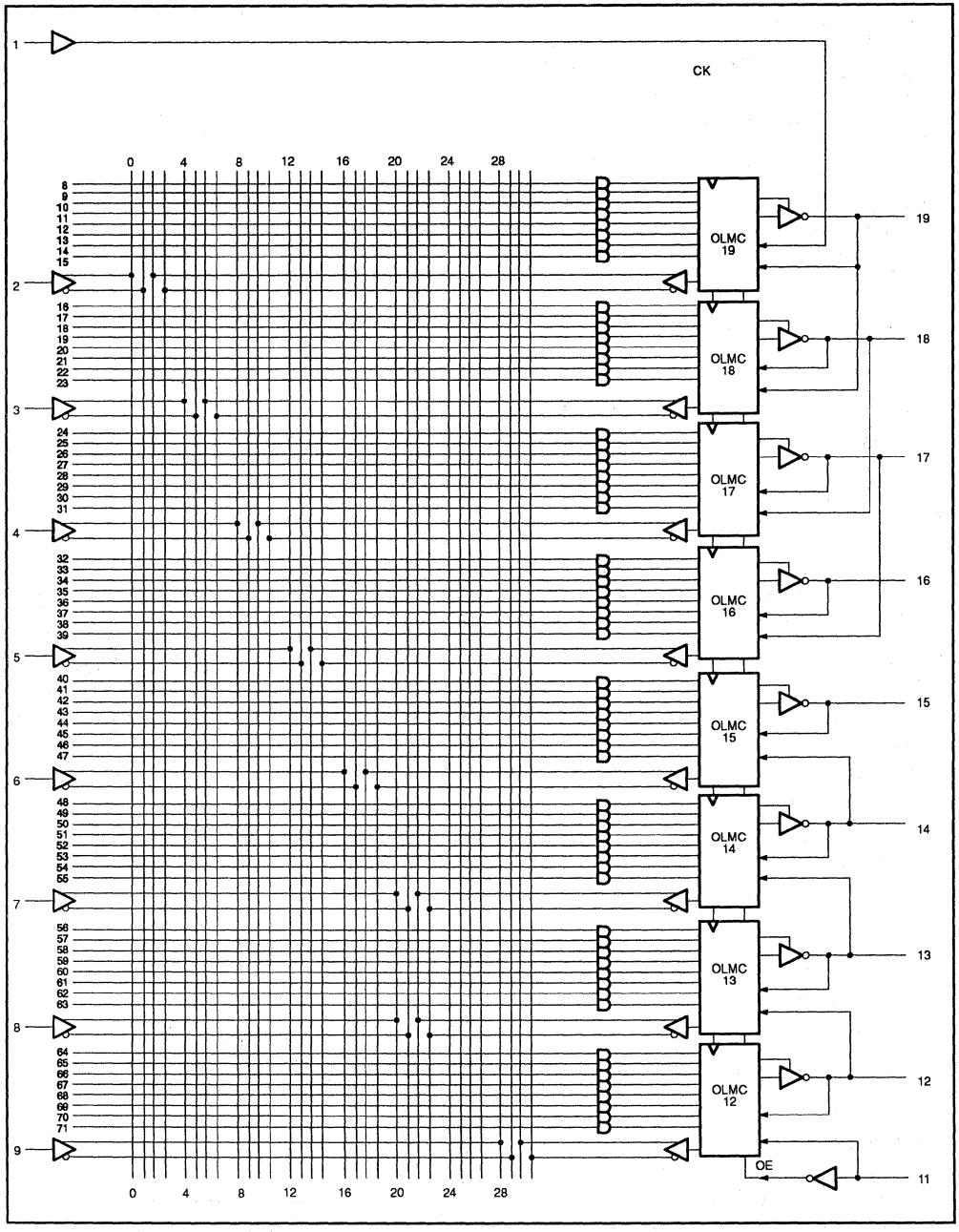
PIN TABLE DESCRIPTIONS

CLK	Clock
GND	Ground
I	Input
I/O	Input/Output Macro Cell
NC	No connect
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
20–Pin Plastic Dual In–Line Package	LVT16V8–7N	0408B
20–Pin Plastic Leaded Chip Carrier	LVT16V8–7A	0400E
20–Pin Plastic Small Outline Large Package	LVT16V8–7D	0172D

LOGIC DIAGRAM



3 Volt BiCMOS Versatile PAL — LVT22V10–7

DESCRIPTION

The LVT22V10–7 is a V–type PAL device designed to operate over the 3.0 to 3.6 volt range. This versatile device is fabricated using the BiCMOS process which produces superior performance, low noise and reduced ground bounce. The reduction from 5V to 3.3V also dramatically reduces the power consumption to less than 100mA (worst case).

This industry standard device is ideal for high performance systems which have been designed to operate with 3.3V ± 0.3V power supplies, as well as systems which are operating with dual supplies (5.0V and 3.3V). The LVT22V10–7 can accept both 3.3 and 5.0V input levels without the need for level translators. Both the inputs and I/O have high state reverse current flow protection to insure that the outputs are not damaged if the 3V LVT22V10 is interfaced with 5V devices.

The LVT22V10–7 is designed with metastable hardened flip–flops so that the outputs can never display a metastable state due to set up or hold time violations. If set up or hold times are violated, the outputs will not glitch or display a metastable state however (propagation delays may extend).

The LVT22V10's flexible architecture supports a wide variety of high performance applications: counters, shift registers, address decoders, state machines multiplexers and random logic collection.

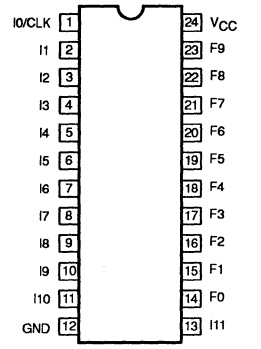
The LVT22V10–7 is identical, in function and fuse map, to other industry standard EEPROM and EPROM 22V10 devices. Development and programming support are offered by Philips and other third party vendors.

FEATURES

- Advanced low voltage BiCMOS process technology
- Ultra high performance over the 3.0 to 3.6 voltage range
  - 7.5ns T_{PD}
  - 5.0ns T_{IS}
  - 6.0ns T_{CKO}
  - 110 MH_Z F_{MAX} (internal feedback)
  - 143 MH_Z clock rate
- Low power dissipation
  - 300mW typical
- 5V compatible inputs and I/O
- Exceptional noise immunity and low ground bounce
- Live insertion/extraction
- Metastable hardened Flip-Flops
- Wide package availability; DIP, PLCC, SOL
- Architectural Flexibility
  - Up to 22 inputs and 10 outputs
  - Variable product term distribution for greater logic flexibility
  - Synchronous preset; asynchronous clear
  - Independently programmable output polarity and output enable
  - Register preload and power up reset of all registers
  - Register Preload and Power Up reset of all registers
- Development and programming support
  - Third party software and programmers
  - Philips SNAP development software

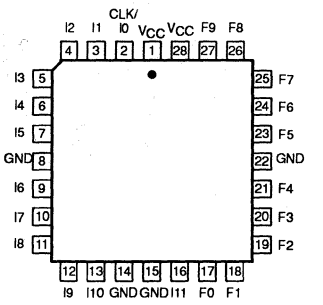
PIN CONFIGURATIONS

D and N Packages



N = Plastic Dual In-Line Package (300mil-wide)
D = Plastic Small Outline Large (300mil-wide) Package

A Package



A = Plastic Leaded Chip Carrier

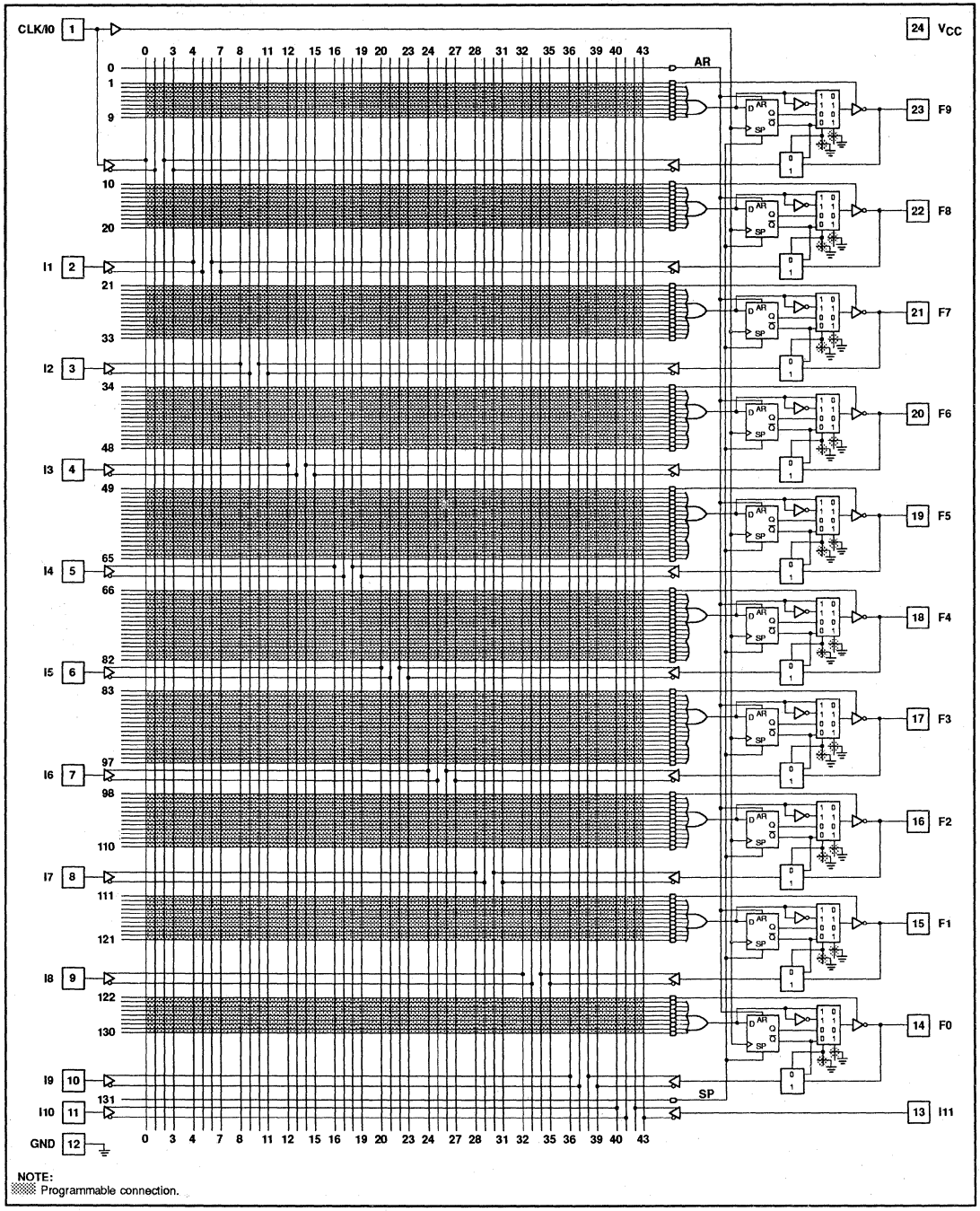
PIN LABEL DESCRIPTIONS

I1 – I11	Dedicated Input
NC	Not Connected
F0 – F9	Macro Cell Input/Output
CLK/I0	Clock Input/Dedicated Input
V_{CC}	Supply Voltage
GND	Ground

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24–Pin Plastic Dual In-Line Package	LVT22V10–7N	0410D
28–Pin Leaded Chip Carrier	LVT22V10–7A	0401F
24–Pin Plastic Small Outline Large Package	LVT22V10–7D	0173D

LOGIC DIAGRAM



NOTE:
▒ Programmable connection.

3 Volt BiCMOS Versatile GAL–type PLD — LVT20V8–7

DESCRIPTION

The LVT20V8–7 is a V–type GAL device designed to operate over the 3.0 to 3.6 volt range. This versatile device is fabricated using the BiCMOS process which produces superior performance, low noise and reduced ground bounce. The reduction from 5V to 3.3V also dramatically reduces power to less than 0.5 watts (worst case).

This industry standard device is ideal for high performance systems which have been designed to operate with 3.3V ± 0.3V power supplies, as well as systems which are operating with dual supplies (5.0V and 3.3V). The LVT20V8–7 can accept both 3.3 and 5.0V input levels without the need for level translators. Both the inputs and I/O have high state reverse current flow protection to insure that the outputs are not damaged if the 3V LVT20V8 is interfaced with 5V devices.

The LVT20V8–7 is designed with metastable hardened flip–flops so the outputs can never display a metastable state due to set up or hold time violations. If set up or hold times are violated, the outputs will not glitch or display a metastable state (however propagation delays may extend).

Active bus-hold circuitry is provided to eliminate the need for external pull up resistors to hold unused or floating inputs at valid logic levels.

The LVT20V8's flexible architecture supports a wide variety of high performance applications: counters, shift registers, address decoders, state machines, multiplexers and random logic collection.

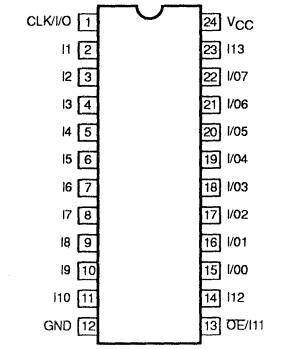
The LVT20V8–7 is identical, in function and fuse map, to other industry standard EEPROM and EPROM 20V8 devices. Development and programming support are offered by Philips and other third party vendors.

FEATURES

- Advanced low voltage BiCMOS process technology
- Ultra high performance over the 3.0 to 3.6 voltage range
  - 7.5ns T_{PD}
  - 5.0ns T_{IS}
  - 6.0ns T_{CKO}
  - 110MHz F_{MAX} (internal feedback)
  - 143MHz clock rate
- Low power dissipation
  - 300mW typical
- 5V compatible inputs and I/O
- Exceptional noise immunity and low ground bounce
- Live insertion/extraction
- Metastable hardened Flip–Flops
- Wide package availability; DIP, PLCC, SOL
- Bus-hold data inputs eliminate the need for external pull up resistors
- Architectural Flexibility
  - Emulates all 24 pin PAL devices
  - Up to 20 inputs and 8 outputs
  - Independently programmable I/O macrocells (4 configurations)
  - Independently programmable output polarity
  - Product term output enable for combinatorial functions
  - Register Preload and Power Up reset of all registers
- Development and programming support
  - Third party software and programmers
  - Philips SNAP development software

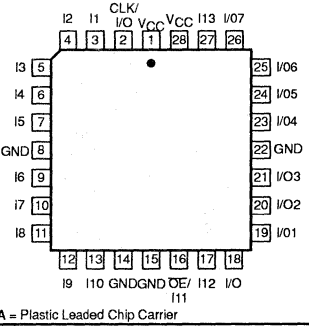
PIN CONFIGURATIONS

D and N Packages



N = Plastic DIP (300mil-wide)
D = Plastic SOL (300 mil–wide)

A Package



A = Plastic Leaded Chip Carrier

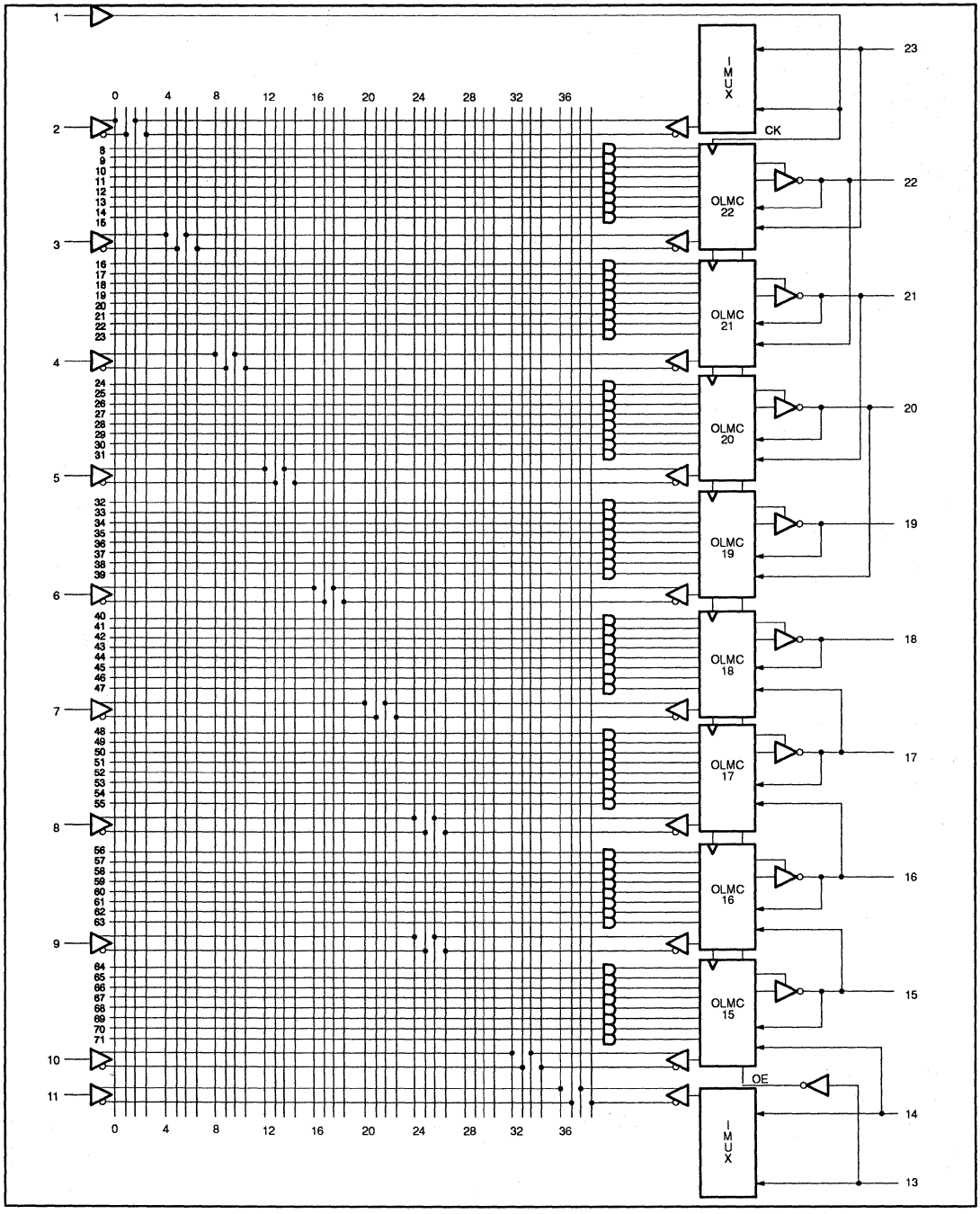
PIN TABLE DESCRIPTIONS

CLK	Clock
GND	Ground
I	Input
I/O	Input/Output Macro Cell
NC	No connect
$\overline{OE}$	Output Enable
V_{CC}	Supply Voltage

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
24–Pin Plastic Dual In–Line Package	LVT20V8–7N	0410D
28–Pin Plastic Leaded Chip Carrier	LVT20V8–7A	0401F
24–Pin Plastic Small Outline Large Package	LVT20V8–7D	0173D

LOGIC DIAGRAM



Section 7
Programmable Macro Logic Devices

CONTENTS

Programmable macro logic PML™

PLHS501/PLHS501I

FEATURES

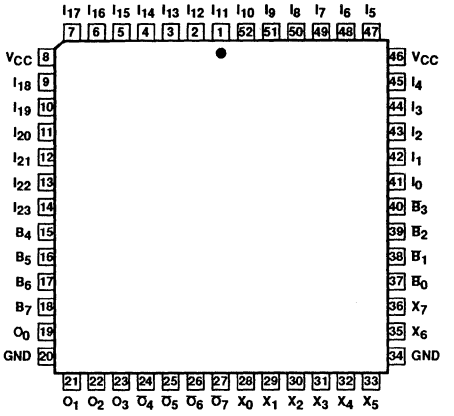
- Programmable Macro Logic device
- Full connectivity
- TTL compatible
- SNAP development system:
  - Supports third-party schematic entry formats
  - Macro library
  - Versatile netlist format for design portability
  - Logic, timing, and fault simulation
- Delay per internal NAND function = 6.5ns (typ)
- Testable in unprogrammed state
- Security fuse allows protection of proprietary designs

STRUCTURE

- NAND gate based architecture
  - 72 foldback NAND terms
- 136 input-wide logic terms
- 44 additional logic terms
- 24 dedicated inputs ($I_0 - I_{23}$)
- 8 bidirectional I/Os with individual 3-State enable:
  - 4 Active-High ($B_4 - B_7$)
  - 4 Active-Low ($\overline{B}_0 - \overline{B}_3$)
- 16 dedicated outputs:
  - 4 Active-High outputs
    O_0, O_1 with common 3-State enable
    O_2, O_3 with common 3-State enable
  - 4 Active-Low outputs:
    $\overline{O}_4$, $\overline{O}_5$ with common 3-State enable
    $\overline{O}_6$, $\overline{O}_7$ with common 3-State enable
  - 8 Exclusive-OR outputs:
    X_0, X_1 with common 3-State enable
    X_2, X_3 with common 3-State enable
    X_4, X_5 with common 3-State enable
    X_6, X_7 with common 3-State enable

PIN CONFIGURATION

A Package
(52-pin PLCC)



DESCRIPTION

The PLHS501 is a high-density Bipolar Programmable Macro Logic device. PML incorporates a programmable NAND structure. The NAND architecture is an efficient method for implementing any logic function. The SNAP software development system provides a user friendly environment for design entry. SNAP eliminates the need for a detailed understanding of the PLHS501 architecture and makes it transparent to the user. PLHS501 is also supported on the Philips Semiconductors SNAP software development systems.

The PLHS501 is ideal for a wide range of microprocessor support functions, including bus interface and control applications.

The PLHS501 is also processed to industrial requirements for operation over an extended temperature range of –40°C to +85°C and supply voltage of 4.5V to 5.5V.

ARCHITECTURE

The core of the PLHS501 is a programmable fuse array of 72 NAND gates. The output of each gate folds back upon itself and all other NAND gates. In this manner, full connectivity of all logic functions is achieved in the PLHS501. Any logic function can be created within the core of the device without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers.

PML is a trademark of Philips Semiconductors

Programmable macro logic PML™

PLHS501/PLHS501I

ORDERING INFORMATION

DESCRIPTION	OPERATING CONDITIONS	ORDER CODE	DRAWING NUMBER
52-Pin Plastic Leaded Chip Carrier	Commercial Temperature Range ±5% Power Supply	PLHS501A	0397E
52-Pin Plastic Leaded Chip Carrier	Industrial Temperature Range ±10% Power Supply	PLHS501IA	0397E

DESIGN DEVELOPMENT TOOLS

SNAP

The SNAP Software Development System provides the necessary tools for designing with PML. SNAP provides the following:

- Schematic entry netlist generation from third-party schematic design packages such as OrCAD/SDT III™ and FutureNet™.
- Macro library for standard TTL functions and user defined functions
- Boolean equation entry
- State equation entry
- Syntax and design entry checking
- Simulator includes logic simulation, fault simulation and timing simulation.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. The minimum system configuration for SNAP is 640K bytes of RAM and a hard disk.

SNAP provides primitive PML function libraries for third-party schematic design packages. Custom macro function libraries can be defined in schematic or equation form.

After the completion of a design, the software compiles the design for syntax and completeness. Complete simulation can be carried out using the different simulation tools available.

The programming data is generated in JEDEC format. Using the Device Programmer Interface (DPI) module of SNAP, the JEDEC fusemap is sent from the host computer to the device programmer.

DESIGN SECURITY

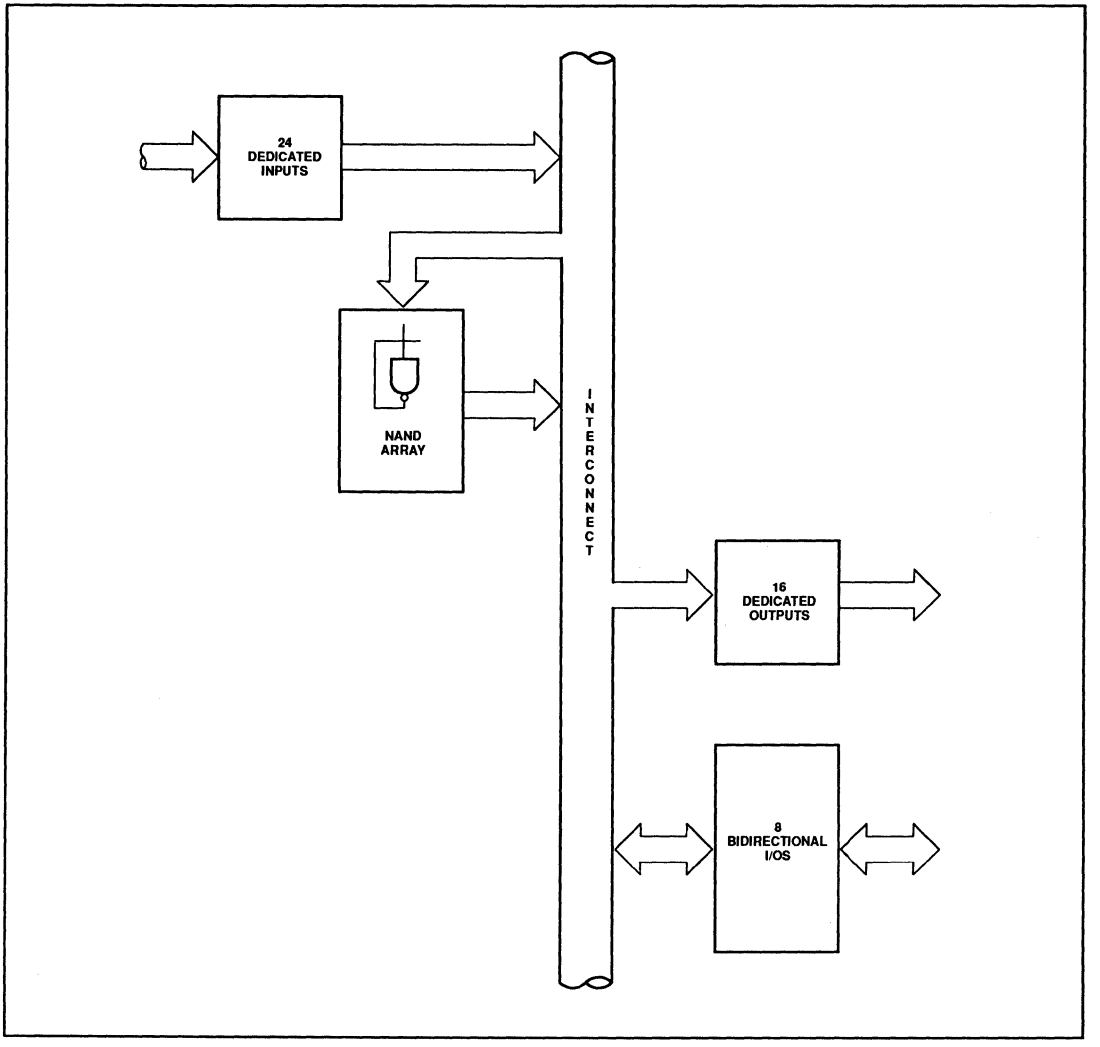
The PLHS501 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary designs implemented in the device cannot be copied or retrieved.

PROGRAMMING/SOFTWARE SUPPORT

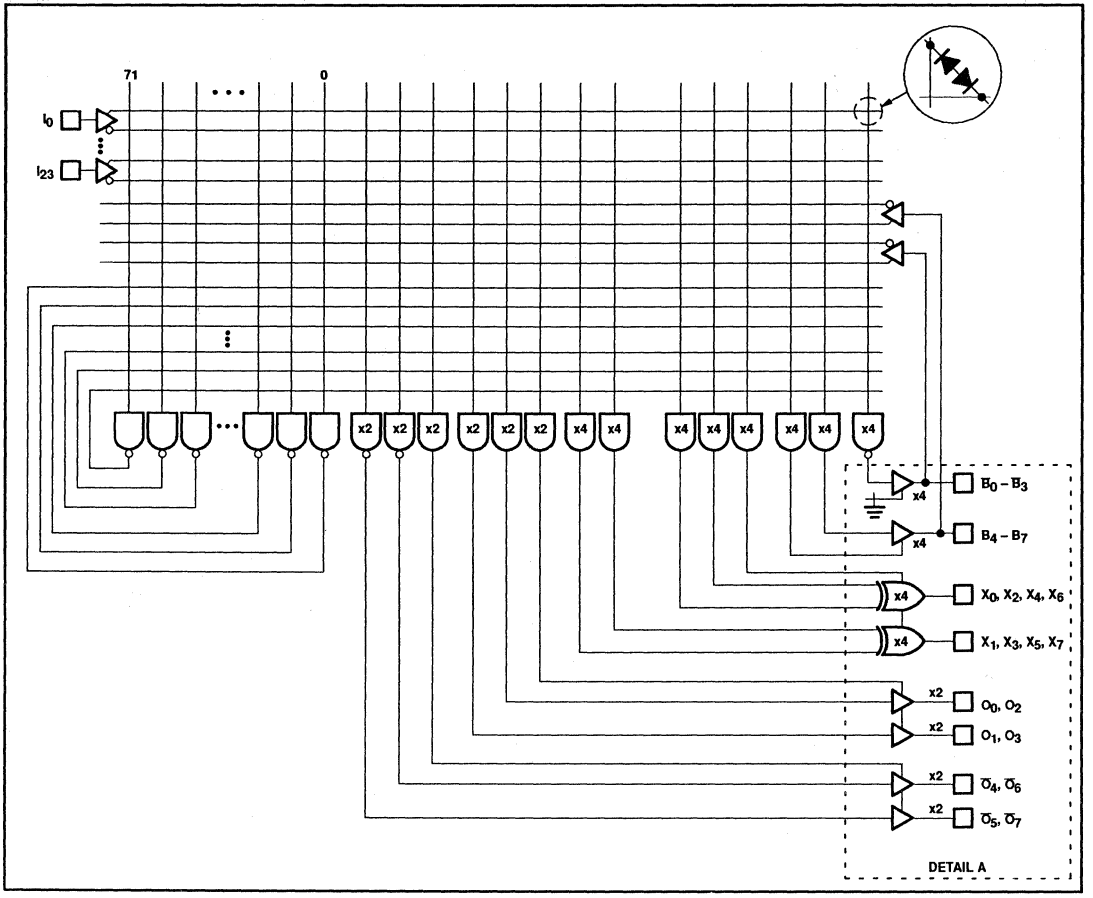
Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

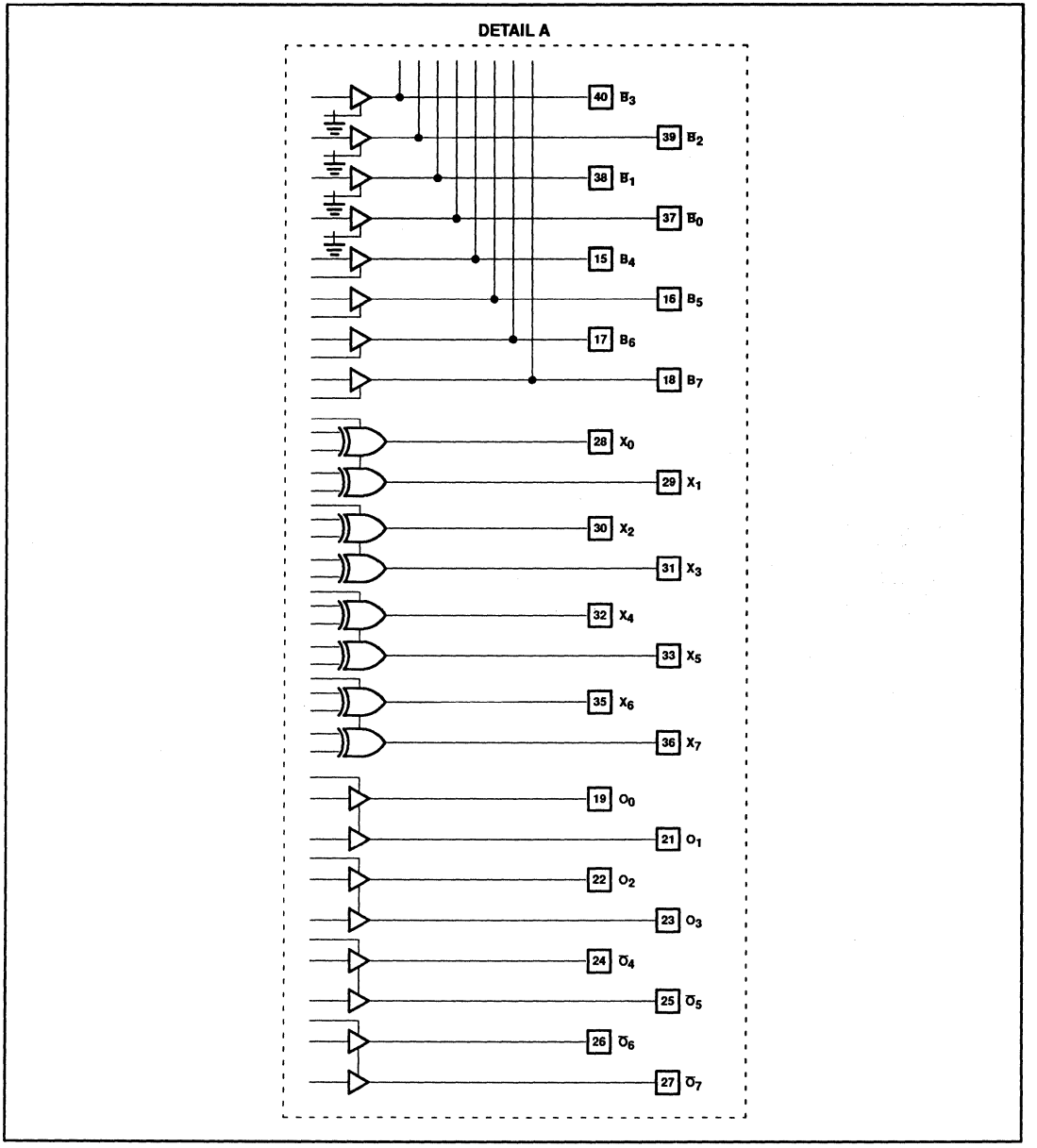
FutureNet is a trademark of FutureNet Corporation.
OrCAD/SDT is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

PLHS501 FUNCTIONAL BLOCK DIAGRAM



FUNCTIONAL DIAGRAM




DETAIL A
40 B3
39 B2
38 B1
37 B0
15 B4
16 B5
17 B6
18 B7
28 X0
29 X1
30 X2
31 X3
32 X4
33 X5
35 X6
36 X7
19 O0
21 O1
22 O2
23 O3
24 O4
25 O5
26 O6
27 O7

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS		UNIT
		MIN	MAX	
V_{CC}	Supply voltage		+7	V_{DC}
V_{IN}	Input voltage		+5.5	V_{DC}
V_{OUT}	Output voltage		+5.5	V_{DC}
I_{IN}	Input currents	–30	+30	mA
I_{OUT}	Output currents		+100	mA
T_{amb}	Operating temperature range	0	+75	°C
T_{stg}	Storage temperature range	–65	+150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All product terms are enabled.
2. All bidirectional (B) pins are outputs.
3. All outputs are enabled.
4. All outputs are Active-High **except** $\overline{B}_0 - \overline{B}_3$ (fusible I/O) and $\overline{O}_4 - \overline{O}_7$ which are Active-Low.

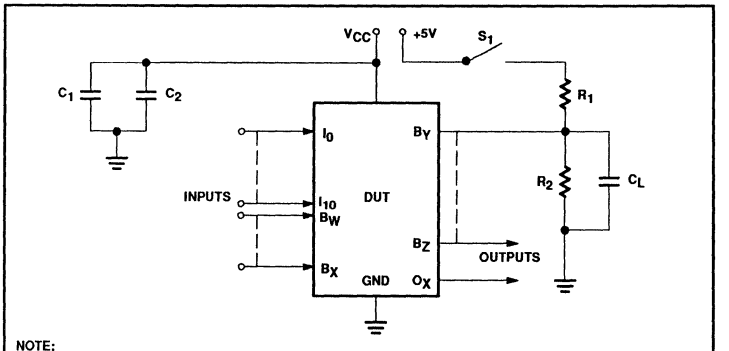
DC ELECTRICAL CHARACTERISTICS

Commercial= 0°C ≤ T_{amb} ≤ +75°C, 4.75V ≤ V_{CC} ≤ 5.25V
Industrial = −40°C ≤ T_{amb} ≤ +85°C, 4.5V ≤ V_{CC} ≤ 5.5V

SYMBOL	PARAMETER	TEST CONDITION	LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage[2]						
V_{IL}	Low	V_{CC} = MIN			0.8	V
V_{IH}	High	V_{CC} = MAX	2.0			V
V_{IC}	Clamp[2, 3]	V_{CC} = MIN, I_{IN} = −12mA		−0.8	−1.2	V
Output voltage						
V_{OL}	Low[2, 4]	V_{CC} = MIN I_{OL} = 10mA			0.45	V
V_{OH}	High[2, 5]	I_{OH} = −2mA	2.4			V
Input current						
I_{IL}	Low	V_{CC} = MAX V_{IN} = 0.45V			−100	µA
I_{IH}	High	V_{IN} = 5.5V			40	µA
Output current						
$I_{O(OFF)}$	Hi-Z state[9]	V_{CC} = MAX V_{OUT} = 5.5V V_{OUT} = 0.45V			80 −140	µA
I_{OS}	Short circuit[3, 5, 6]	V_{OUT} = 0V	−15		−70	mA
I_{CC}	V_{CC} supply current[8]	V_{CC} = MAX		225	295	mA
Capacitance						
C_{IN}	Input	V_{CC} = 5V V_{IN} = 2.0V		8		pF
C_B	I/O	V_{OUT} = 2.0V		15		pF

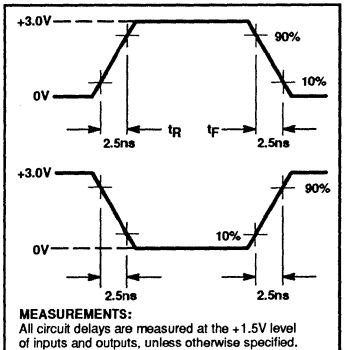
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. All voltage values are with respect to network ground terminal.
3. Test one at a time.
4. For Pins 15 – 19, 21 – 27 and 37 – 40, V_{OL} is measured with Pins 5 and 41 = 8,75V, Pin 43 = 0V and Pins 42 and 44 = 4.5V. For Pins 28 – 33 and 35 – 36, V_{OL} is measured under same conditions EXCEPT Pin 44 = 0V.
5. V_{OH} is measured with Pins 5 and 41 = 8.75V, Pins 42 and 43 = 4.5V and Pin 44 = 0V.
6. Duration of short circuit should not exceed 1 second.
7. I_{CC} is measured with all dedicated inputs at 0V and bidirectional and output pins open.
8. Measured at $V_T = V_{OL} + 0.5V$.
9. Leakage values are a combination of input and output leakage.

TEST LOAD CIRCUITS



NOTE:
C_1 and C_2 are to bypass V_{CC} to GND.

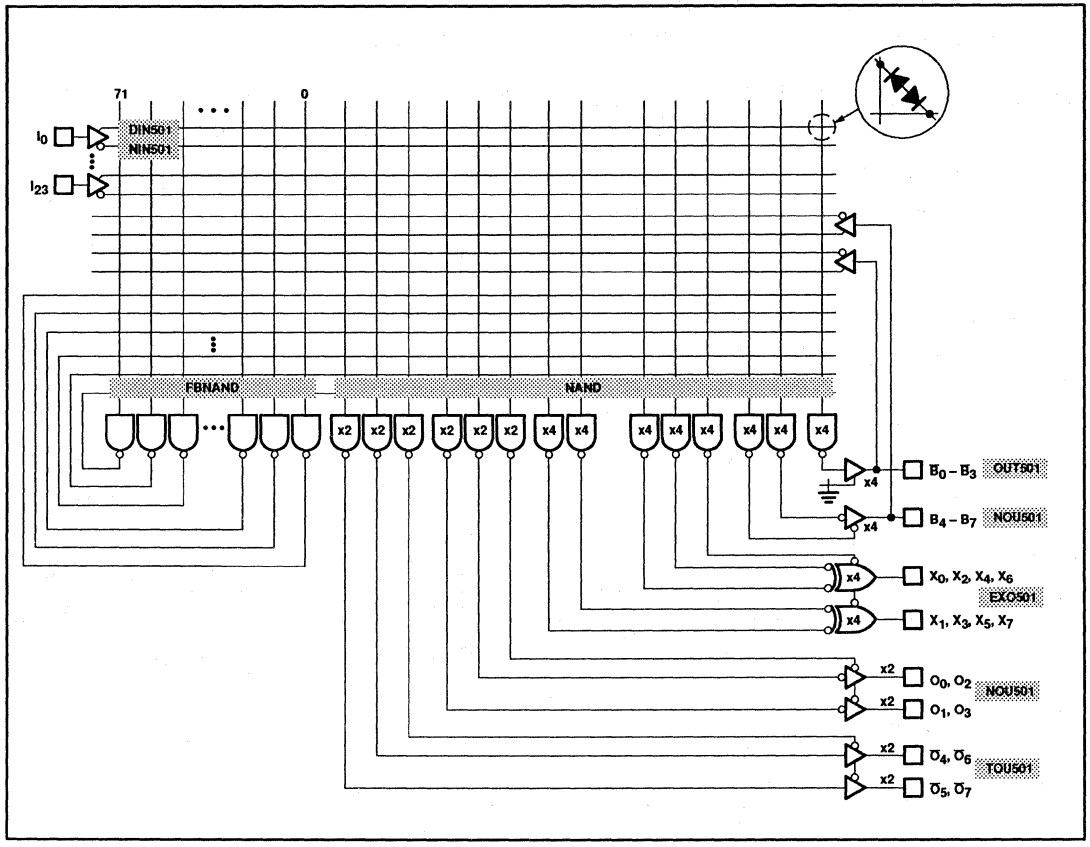
VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

SNAP RESOURCE SUMMARY DESIGNATIONS



MACRO CELL SPECIFICATIONS[1] (SNAP Resource Summary Designations in Parantheses)

Commercial: T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω
Industrial: T_{amb} = –40°C to +85°C, 4.5V ≤ V_{CC} ≤ 5.5V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω

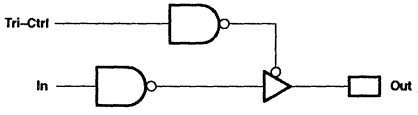
Input Buffer
(DIN501 [Non-Inverting], NIN501 [Inverting])

SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{HL}	0.05	0.1	0.15	ns/p-term
Δt_{LH}	–0.02	–0.05	–0.08	ns/p-term

SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL}	X	I	4.5	5.5	6.5	ns	With 0 p-terms load
t_{PLH}	X	I	5	6	7.5	ns	
t_{PHL}	Y	I	2.5	3	3.5	ns	With 0 p-terms load
t_{PLH}	Y	I	4	4	4.5	ns	

Input Pins: 1 – 7, 9 – 14, 41 – 45, 48 – 52.
Bidirectional Pins: 15 – 18, 37 – 40.
Maximum internal fan-out: 16 p-terms on X or Y.

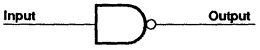
NAND Output Buffer with 3-State Control
(TOU501)



SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT
t_{PHL}	Out	In	8.5	14.0	17.5	ns
t_{PLH}	Out	In	8.5	14.0	16	ns
t_{OE}[2]	Out	Tri-Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri-Ctrl	8.5	12.5	17.0	ns

Output Pins: 24 – 27.

Internal Foldback NAND
(FBNAND)



SYMBOL	LIMITS MIN	TYP	MAX	UNIT
Δt_{PHL}	0.05	0.1	0.15	ns/p-term
Δt_{PLH}	–0.0	–0.05	–0.1	ns/p-term

SYMBOL	PARAMETER To (Output)	From (Input)	LIMITS MIN	TYP	MAX	UNIT	NOTES
t_{PHL}	Out	Any	4.0	4.5	6.8	ns	With 0 p-terms load
t_{PLH}			5.5	6.5	8	ns	

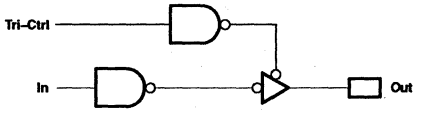
Maximum internal loading of 16 terms.

Notes are on following page.

MACRO CELL SPECIFICATIONS[1] (Continued) (SNAP Resource Summary Designations in Parantheses)

Commercial: T_{amb} = 0°C to +75°C, 4.75V ≤ V_{CC} ≤ 5.25V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω
Industrial: T_{amb} = –40°C to +85°C, 4.5V ≤ V_{CC} ≤ 5.5V, C_L = 30pF, R_2 = 1000Ω, R_1 = 470Ω

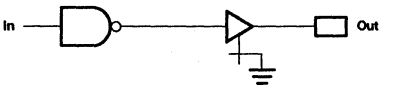
AND Output Buffer with 3-State Control (NOU501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Output	In	8.0	11	13	ns
t_{PLH}	Output	In	8.0	11	13	ns
t_{OE}[2]	Out	Tri-Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri-Ctrl	8.5	12.5	17.0	ns

Bidirectional and Output Pins: 19, 21, 22, 23, 15 – 18.

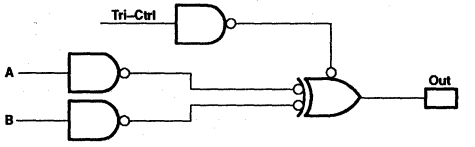
NAND Output Buffer (OUT501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	In	8.5	14	17.5	ns
t_{PLH}	Out	In	8.5	14	16.0	ns

Bidirectional Pins: 37 – 40.

Ex–OR Output Buffer (EXO501)



SYMBOL	PARAMETER		LIMITS			UNIT
	To (Output)	From (Input)	MIN	TYP	MAX	
t_{PHL}	Out	A or B	8.5	14	17.5	ns
t_{PLH}	Out	A or B	8.5	14	16.0	ns
t_{OE}[2]	Out	Tri-Ctrl	8.5	15	18.5	ns
t_{OD}[2]	Out	Tri-Ctrl	8.5	12.5	17.0	ns

Ex-OR Output Pins: 28 – 33.

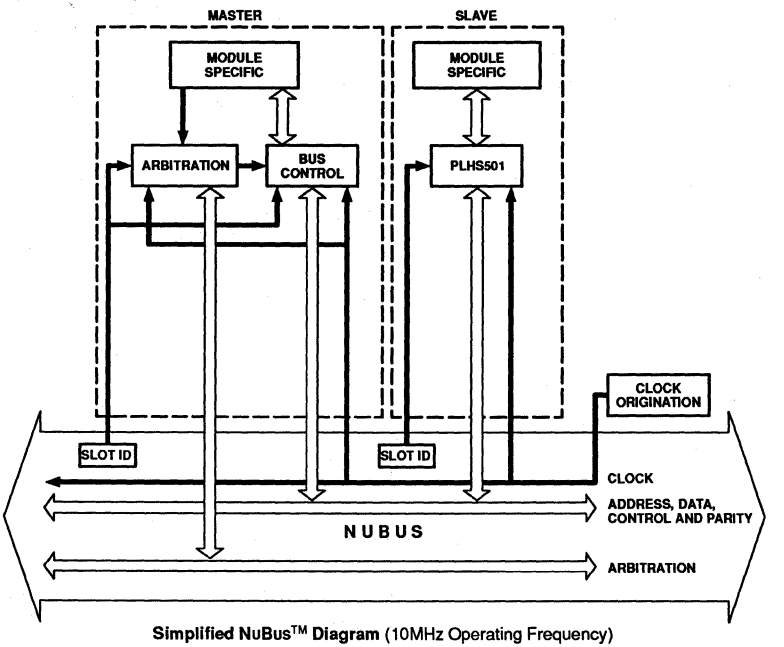
NOTES:

1. Limits are guaranteed with internal feedback buffers simultaneously switching cumulative maximum of eight outputs.
2. For 3-State output; output enable times are tested with C_L = 30pF to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with C_L = 5pF. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

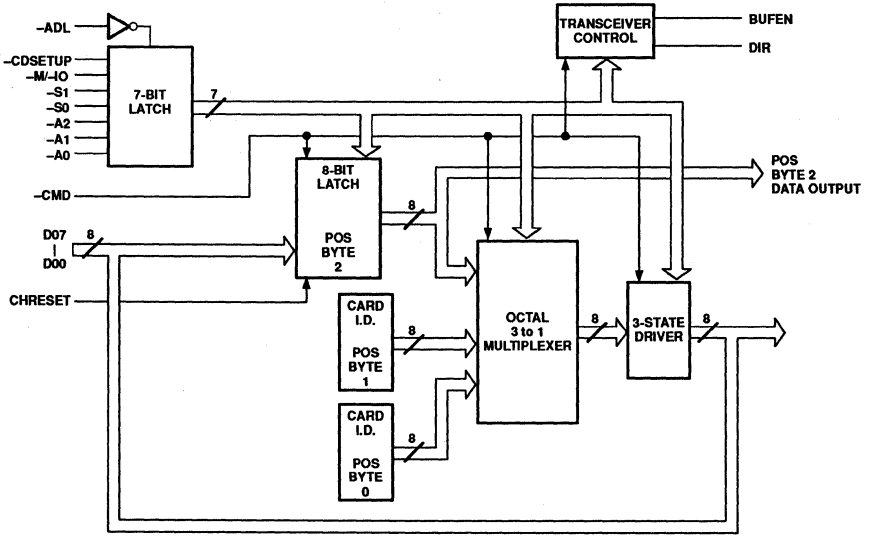
PLHS501 GATE AND SPEED ESTIMATE TABLE

FUNCTION	INTERNAL NAND EQUVALENT	TYPICAL t_{PD}	f_{MAX}	COMMENTS
Gates				
NANDs	1	6.5ns		For 1 to 32 input variables
ANDs	1	6.5ns		For 1 to 32 input variables
NORs	1	6.5ns		For 1 to 32 input variables
ORs	1	6.5ns		For 1 to 32 input variables
Decoders				
3-to-8	8	11ns		Inverted inputs available
4-to-16	16	11ns		Inverted inputs available
5-to-32	32	11ns		Inverted inputs available (24 chip outputs only)
Encoders				
8-to-3	15	11ns		Inverted inputs, 2 logic levels
16-to-4	32	11ns		Inverted inputs, 2 logic levels
32-to-5	41	11ns		Inverted inputs, 2 logic levels, factored solution.
Multiplexers				
4-to-1	5	11ns		Inverted inputs available
8-to-1	9	11ns		
16-to-1	17	11ns		
27-to-1	28	11ns		Can address only 27 external inputs - more if internal
Flip-Flops				
D-type Flip-Flop	6		30MHz	With asynchronous S-R
T-type Flip-Flop	6		30MHz	With asynchronous S-R
J-K-type Flip-Flop	10		30MHz	With asynchronous S-R
Adders				
8-bit	45	15.5ns		Full carry-lookahead (four levels of logic)
Barrel Shifters				
8-bit	72	11ns		2 levels of logic
Latches				
D-latch	3			2 levels of logic with one shared gate

APPLICATIONS



Simplified NuBus™ Diagram (10MHz Operating Frequency)



Block Diagram of Basic POS Implementation in PLHS501

NuBus is a trademark of Texas Instruments, Inc.

CMOS high density programmable macro logic — PML2552

FEATURES

- Full connectivity
- Erasable version and one time programmable version available
- Scan test
- Power down mode
- Power on reset
- 100% testable
- SNAP development system
  - Supports third-party schematic entry formats
  - TTL Macro library
  - Versatile netlist format for design portability
- Power dissipation (TTL) = 630mW
- Power dissipation (CMOS) = 525mW
- Power dissipation (Power-Down mode) = 52mW
- Security fuse for copy protection
- Reprogrammable

PROPAGATION DELAYS

- Delay per internal NAND gate = 15ns (typ)
- 50MHz flip-flop toggle rate

APPLICATIONS

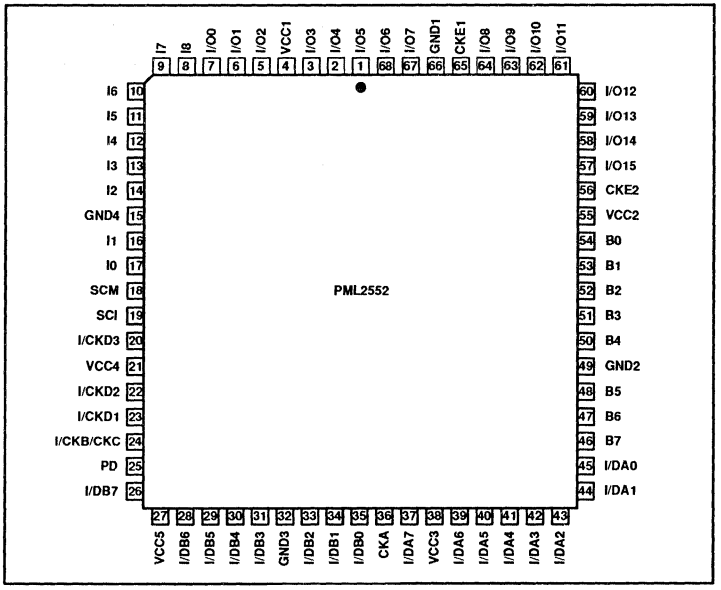
- Low-end gate array replacement
- Instrumentation
- Bus arbitration functions
- Wide multiplexers and decoders
- Multiple independent state machines
- General purpose logic integration and microprocessor support logic
- PAL® and glue logic replacement

DESCRIPTION

The Philips Semiconductors PML family of PLDs provides "instant gate array" capabilities for general purpose logic integration applications. The PML2552 is the first high density CMOS-PML product. Fabricated with the Philips Semiconductors high-performance EPROM process, it is an ideal way to reduce NRE costs, inventory problems and quality concerns. The PML2552 incorporates the PML folded NAND array architecture which provides 100% connectivity to eliminate routing restrictions. What distinguishes the PML2552 from the "classic" PLD architectures is its flexibility and the potent flip-flop building blocks. The device utilizes a folded NAND architecture, which enables the designer to implement multiple levels of logic on a single chip. The PML2552 eliminates the NRE costs, risks, and hard to use design tools associated with semicustom and full custom approaches. It allows the system designer to manage reliable functionality, in less time and space plus a faster time to market. The PML2552 is ideal in todays instrumentation, industrial control, EISA, NuBus™, bus interface and dense state machine applications in conjunction with the state-of-the-art CMOS processors. It is capable of replacing large amounts of TTL, SSI and MSI logic and literally allows the designer to build a system on the chip.

The SNAP development software gives easy access to the density and flexibility of the PML2552 through a variety of design entry formats, including schematic, logic equations, and state equations in any combination.

PIN CONFIGURATION



ORDERING INFORMATION

DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
68-pin Plastic Leaded Chip Carrier	35ns	PML2552-35A	0398E
68-pin "J" Leaded Ceramic Cerquad Package	35ns	PML2552-35KA	1473A
68-pin Plastic Leaded Chip Carrier	50ns	PML2552-50A	0398E
68-pin "J" Leaded Ceramic Cerquad Package	50ns	PML2552-50KA	1473A

PAL is a registered trademark of Advanced Micro Devices, Inc.
NuBus is a trademark of Texas Instruments, Inc.

FUNCTIONAL BLOCK DIAGRAM

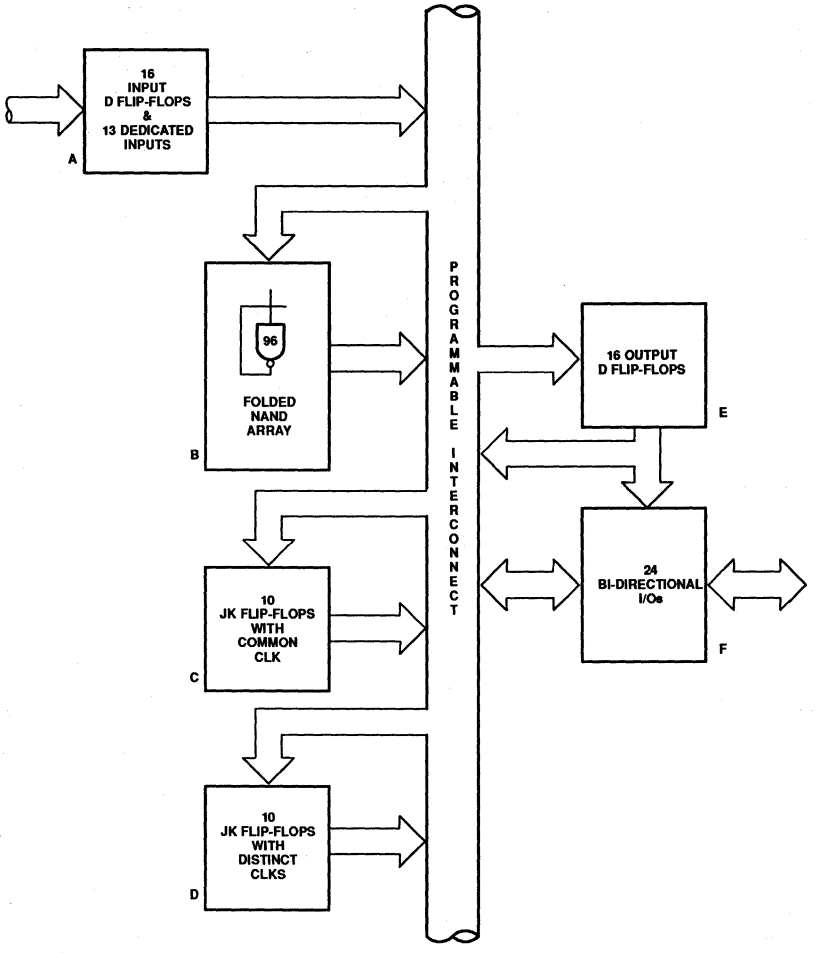
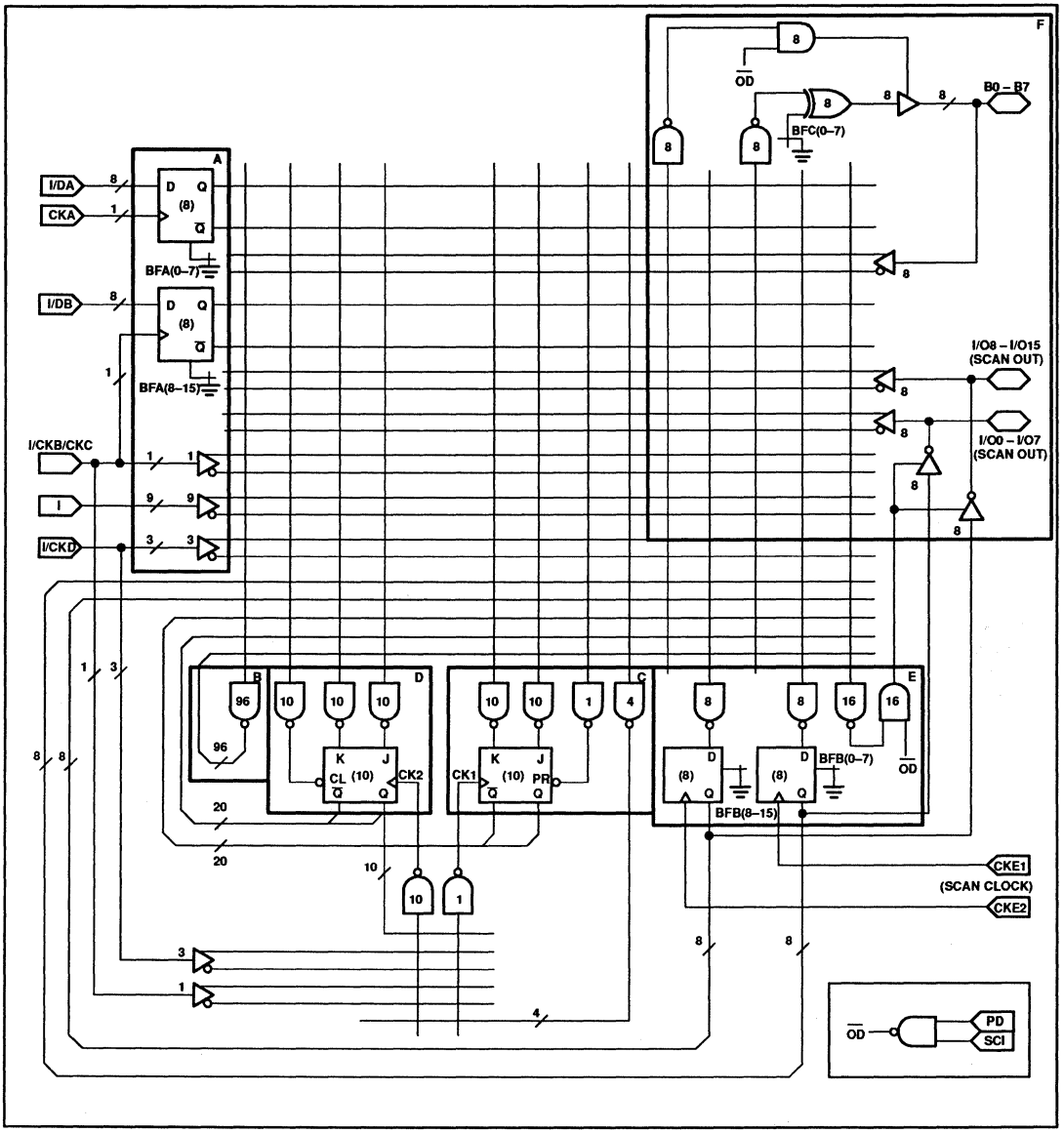


Figure 1.

LOGIC DIAGRAM



CMOS high density programmable macro logic PML2552

STRUCTURE

- 112 possible foldback NAND gates:
  - 96 internal NAND
  - 16 from the I/O macros
- 114 additional logic terms
- 53 possible inputs (with programmable polarity)
  - 29 dedicated inputs
  - 24 bidirectional I/Os
- 24 bidirectional pins
- 52 flip-flops
- 24 possible outputs with individual Output Enable control (8 with programmable polarity)
- Multiple independent clocks
- 20 Buried JK-type flip-flops with foldback (JKFFs):
  - 10 JKFFs with one shared preset signal and one shared clocked signal originating from the clock array.
  - 10 JKFFs with 10 independent clock signals originating from the clock array and 10 independent clear signals
- 258 inputs per NAND gate
- Bypassable Input D-type flip-flop (DFFs)/Combinatorial Inputs:
  - 16 DFFs/combinatorial inputs
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
- Inputs/bypassable D-type flip-flop outputs/foldback NAND gates:
  - 16 output DFFs/combinatorial inputs/outputs with individual Output Enable control
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
  - The DFF can be used as an internal DFF or an internal foldback NAND gate.
- Combinatorial inputs:
  - 9 dedicated inputs to the NAND array
  - 3 inputs optional to NAND array and/or clock array
  - 1 input optional to NAND array and/or clock array, and/or clock of Input D Flip-Flops (Group B)
- Separate clock array:
  - Separate clock array for JKFFs clock inputs
  - 4 inputs to clock array originated from NAND array
  - 4 inputs (with programmable polarity) directly from input pins
  - 10 inputs from Q outputs of JKFFs with clear
- Dedicated clocks:
  - One dedicated clock for input DFFs (Group A)
  - Two dedicated clocks for output DFFs
- Scan test feature:
  - Scan chain is implemented through the 20 buried JKFFs and 16 output DFFs
  - Pins SCI, SCM, and CKE1 are used to operate the scan test
- Power down mode
  - Dedicated pin (PD) freezes the circuit when brought to logic "1". The circuit remains in the same state prior to the logic "0" to logic "1" transition of the "PD" pin.
  - When in the power down mode, the SCI pin acts as the 3-State pin for the 24 outputs.
- Power on reset:
  - All flip-flops (16 input DFFs, 20 buried JKFFs, and 16 output DFFs) are reset to logic "0" after V_{CC} power on.

ARCHITECTURE

The core of the PML2552 is a programmable NAND array of 96 NAND gates and 20 buried JKFFs. The output of each NAND gate folds back upon itself and all other NAND gates and flip-flops. The 'Q' and '$\overline{Q}$' output of each flip-flop also folds back in the same manner. Thus, total connectivity of all logic functions is achieved in the PML2552. Any logic function can be created within the core without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers. Figure 1 shows the functional block diagram of the PML2552.

Macro Cells

There are 16 bypassable DFFs on the input to the NAND array. These flip-flops are split in two banks of 8 (Bank A and Bank B). Each bank of flip-flops has a common clock. In the unprogrammed state of the device the flip-flops are active. In order to bypass any DFF, its respective bypass fuse (BFA_X) must be programmed.

The 16 I/O pins (IO_0 - IO_{15}) and their respective D flip-flop macros can be used in any one of the following configurations:

1. As combinatorial input(s).
   Each of the 16 3-State outputs can be individually disabled by the associated NAND term and the pin is used as an inverting or non-inverting input.
2. As registered DFF outputs.
   These DFFs are split into two banks of 8, and each bank is clocked separately. The bypass fuse BFB_X (see PML2552 Logic Diagram) is used to bypass any one of these DFFs. The flip-flops are all active in an unprogrammed device.
3. As combinatorial outputs.
   By programming the bypass (BFB_X) fuse of any one of the DFFs, the flip-flop(s) is bypassed. The I/O pin can then be used as a combinatorial output.
4. As Internal foldback DFFs or foldback NAND gates.
   When the I/O pin is used as an input, the output macro can be used as an internal DFF or a foldback NAND term. If the bypass fuse is programmed, the macro will act as a foldback NAND term. Otherwise it will act as an internal DFF.

The 8 bidirectional pins (B0-B7) can be used as either combinatorial inputs or outputs with programmable polarity. The Exclusive-OR polarity gates are non-inverting in the unprogrammed state.

The NAND signal labeled '$\overline{OD}$' (Output Disable) shown on the PML2552 logic diagram is used for the Power Down mode operation. This signal disables the outputs when the device enters the Power Down mode and SCI is high.

Clock Array

The 20 buried JKFFs can be clocked through the 'Clock Array'. The Clock Array consists of 11 NAND terms. Ten of these terms are connected to the clock inputs of the Bank A flip-flops that can be clocked individually. One NAND gate is connected to Bank B flip-flops that have a common clock. There are 18 inputs to the clock array. Four come directly from the input pins (with programmable polarity), 4 inputs are from 4 NAND gates connected directly to the folded NAND array. 10 inputs are from the Q outputs of the JKFFs with clear.

SCAN TEST FEATURE

With the rise in the ratio of devices on a chip to the number of I/O pins, Design For Testability is becoming an essential factor in logic design methodology. The PML2552 incorporates a variable length scan test feature which permits access to the internal flip-flop nodes without requiring a separate external I/O pin for each node accessed. Figure 2 (Scan Mode Operation) shows how a scan chain is implemented through the 20 buried JKFFs and 16 output DFFs. Two dedicated pins, SCI (Scan In) and SCM (Scan Mode), are used to operate the scan test. The SCM pin is used to put the circuit in scan mode. When this pin is brought to a logic "1", the circuit enters the scan mode.

In this mode it is possible to shift an arbitrary test pattern into the flip-flops. The SCI pin is used to input the pattern. The inverted outputs of flip-flops D0 - D15 are observable on pins I/O0 - I/O15.

The following are features and characteristics of the device when in Scan Mode:

1. CKE1 is the common scan-clock for all the flip-flops when in scan mode. CKE1 overrides all clock resources of normal operational mode.
2. The Preset (PR) and Clear (CL) functions of the flip-flops are disabled.
3. Scan overrides the bypass fuse of the flip-flops. This means that all the bypassable DFFs remain intact during scan operation even though they may have been bypassed during normal operation.
4. To observe the SCAN data, the output buffers must be enabled by the Output Enable (tri-ctrl) terms.
5. The outputs of the flip-flops are complemented on pins I/O0 - I/O15.
6. All external inputs to flip-flops in the scan chain are disabled when the device enters the scan mode.
7. Blowing the security fuse does not disable the Scan Test feature.

SCAN MODE OPERATION

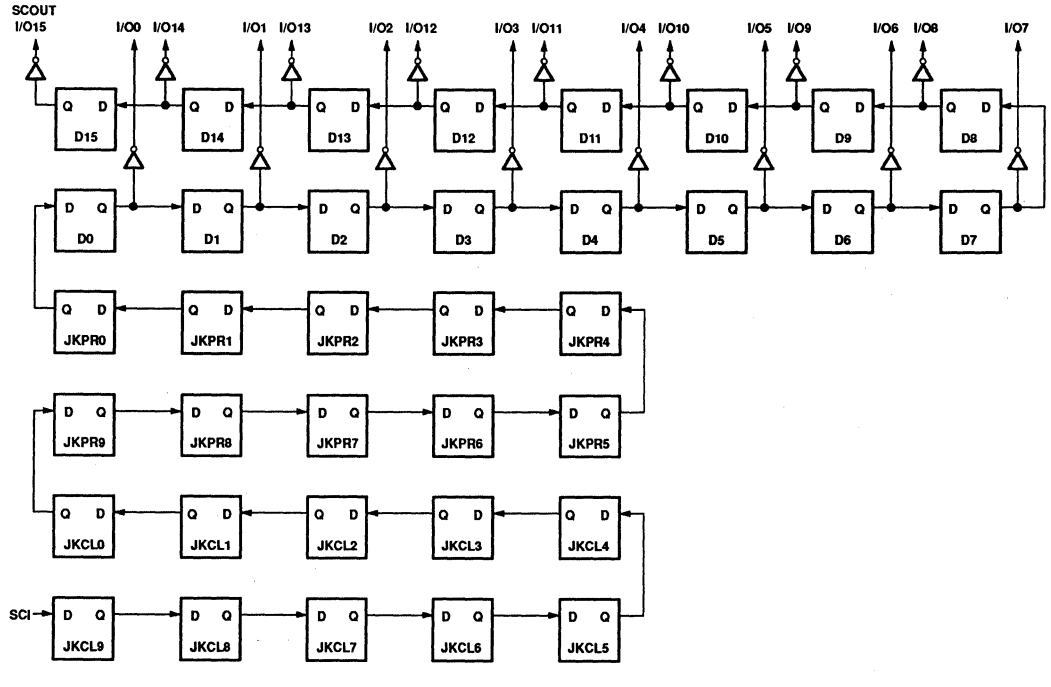


Figure 2.

SCAN TEST STRATEGY

The scan test pattern is design dependent and the user must make considerations for Design For Testability (DFT) during the initial stages of the design. A typical test sequence is to pre-load (i.e., enter a state); revert to normal operation (i.e., activate the next state transition); go back to scan mode to check the result. Note that the scan test feature available in the PML2552 is a variable length scan chain. The DATA entered at SCI (JKCL9) can be accessed anywhere between 21 clock cycles (at I/O0) to 36 clock cycles (at I/O15). For the strategy discussed here, DATA is read out after 36 clocks at I/O15 (i.e., D15). The following operation sequence suggests a possible scan test method.

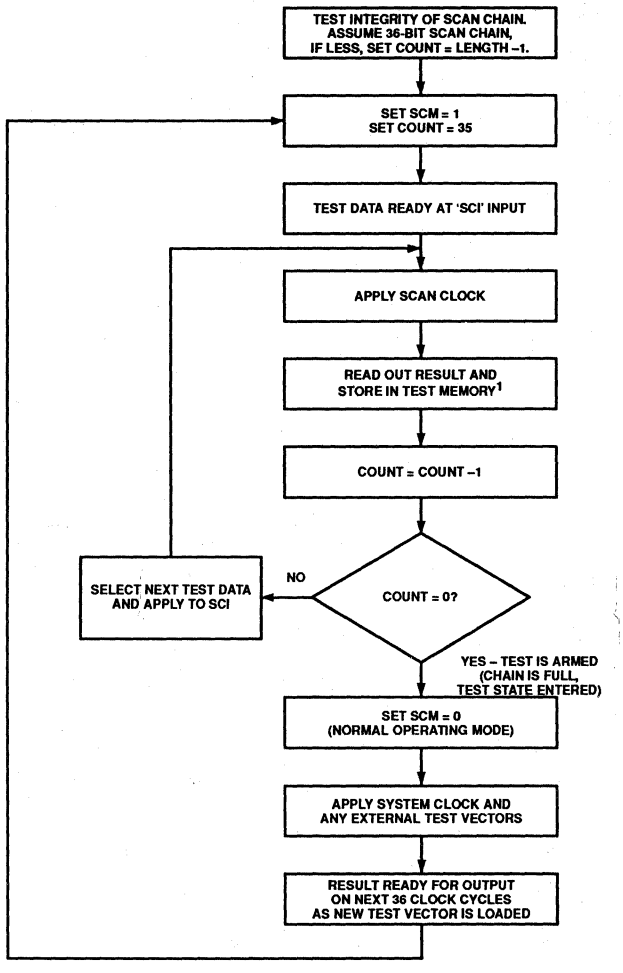
A conservative test policy demands proof that the test facility is working. Thus, to prove Scan Chain holds and maintains correct data:

a. Fill chain with several patterns (for example, all ones and all zeros).

b. Retrieve same patterns.

The user is responsible for managing an external test memory buffer for applied vectors and results, as part of the test equipment.

1. Parallel readout of I/O0 - I/O15 is possible, but assume only I/O15 is used for this strategy.
2. The first DATA entered at SCI (or JKCL9) will be the content of D15 after 36 clocks. This DATA will be inverted at the output pin I/O15 (i.e., SCOUT). The last DATA entering the scan chain will be the content of JKCL9. Thus, the scan chain resembles a first-in-first-out shift register with inverted outputs (I/O0 - I/O15).
3. 'Test Data' is read in at the SCI input and read out of the SCOUT output pin (I/O15). To enter 'Test Data':
   a. Put device in Scan Mode by applying the scan control signals (SCM=1).
   b. Clock device with scan clock (CKE1).
   c. Apply consecutive serial test vectors.
   d. Read back results as new 'Test Data' (States) are applied. The first 36 outputs read at SCOUT (I/O15) are random ('old') data (e.g., remnant of Step 1).
   e. Apply 36 'Test Data' until the chain is full.
4. To apply 'Test Data' (States), exit Scan Mode and apply on system clock together with any other possible test vectors.
5. To read result of the state transition, re-enter scan and apply the scan clock (CKE1). The result of the state transition in JKCL9 will be available at SCOUT (I/O15) after 36 clocks. The results can be stored in a user defined test memory buffer in inverted logic representation.
6. As the results are being read and stored, new 'Test Data' can be entered via SCI.
7. Repeat for all test patterns of interest.
8. Figure 3 (FLOW_CHART) depicts a flow chart version of the test sequence.



NOTE:
1. The first 36 outputs are random ('OLD') data.

Figure 3. FLOW_CHART

A Simple Example

Assume the last three cells of the scan chain (JKCL9, JKCL8, JKCL7 in Figure 4 contain a 3-bit up counter. Our test vector will be a single clock applied to the counter. Suppose we wish to first check the *State 5* (i.e., 101) to *State 6* (i.e., 110) transition, then the *State 3* (i.e., 011) to *State 4* (i.e., 100) transition. Assume the scan chain has been pre-verified and we may begin.

Enter scan mode (set SCM=1)l apply 36 bits in sequence so that the value *101* (i.e., State 5) resides in the last three cells. Exit scan mode (set SCM=0) and apply a single clock to the counter. Now the value *110* (i.e., State 6) resides in the last three cells. Re-enter scan mode (set SCM=1) and read back 36 bits from position I/O15. Note that the outputs are complemented and are also read back in the reverse order. Therefore the value for *STATE 6* read at I/O15 will be 100 which is the complement of *STATE 6* (110) read in the reverse order.

As this is being read back, apply a new state, serially equal to the value *011* (i.e., State 3). This state should be loaded on the last three clock cycles during which *STATE 6* is being read back at I/O15. After *STATE 3* has been loaded (and *STATE 6* read back), exit scan mode and apply a single clock which will invoke the *STATE 3* (i.e., 011) to *STATE 4* (i.e., 100) transition. Re-enter scan mode and read back 36 bits at I/O15. The last three bits should contain *110* which is the complement of *State 4* read in the reverse order. Figure 4 (SCAN_EXAMPLE) shows a flow diagram of this example. Note that the States will always be complemented and read back in the reverse at I/O15. Other sequences may be applied in the same manner.

A possible alternative to this example is to read back the output states at I/O0 (D0) instead of I/O15 (JKCL9). This will allow the outputs to be read back after 21 clock cycles rather than the 36 used in the above example.

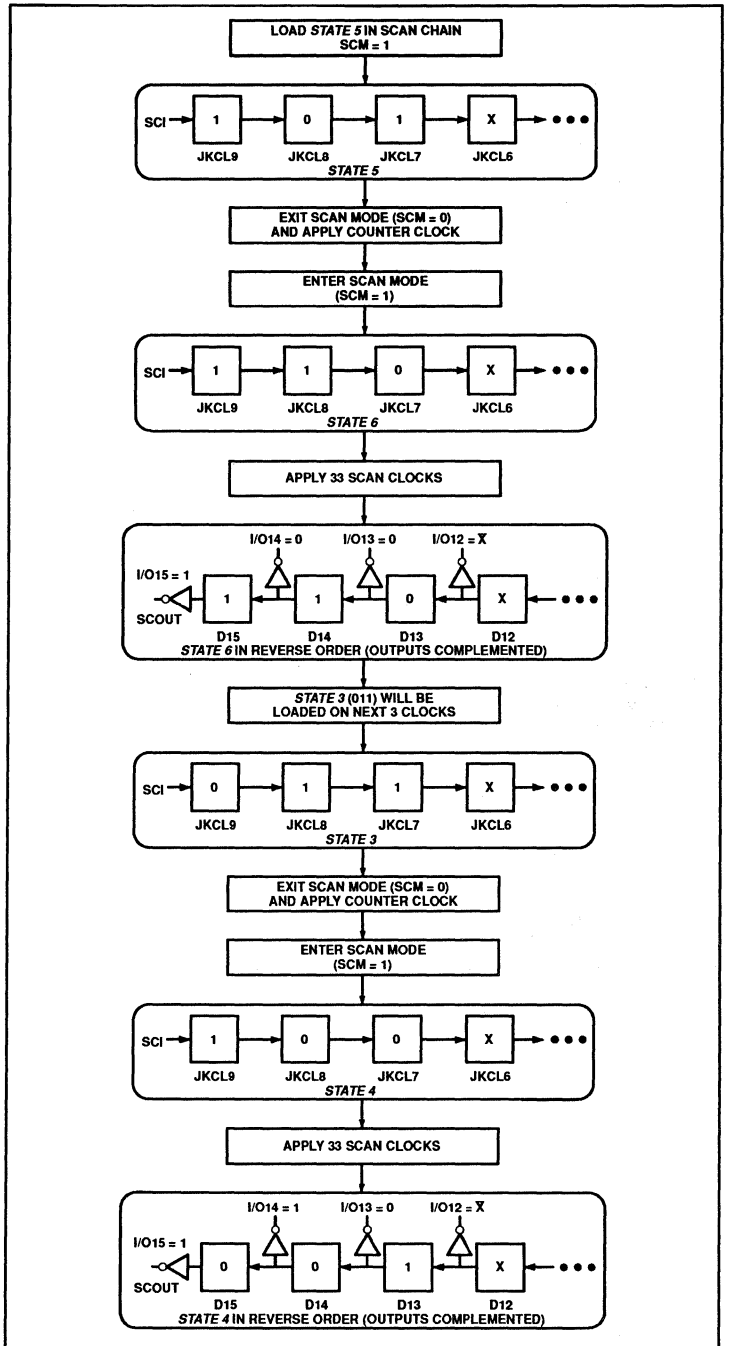


Figure 4. SCAN_EXAMPLE

POWER DOWN

The PML2552 offers the user controlled capability of putting the device to "sleep" where power dissipation is reduced to very low levels. When brought to a logic "1", the PD pin freezes the circuit while reducing the power. All data is retained. This not only includes that of the registers, but also the state of each foldback gate. For those cases where it is desirable to 3-State the outputs, that can be accomplished by raising the SCI pin to a logic "1".

There is one point that should be noted while the circuit is in its power-down mode. The switching of any external clock pin will cause a disruption of the data. All clocks must be frozen before the circuit goes into power-down and stay that way until it powered back up. Clocks that are internally generated and feed the clock array are automatically stopped by the power-down circuitry. Any other input can toggle without any loss of data.

NOTE:

1. During power down, external clocks (CKA, CKB/CKC, CKE1, CKE2) should not change.
2. SCM must be "0" as in normal operation mode.
3. External clock recovery time (low-to-high) is 60ns (high-speed) and 70ns (standard) after the device is powered up.
4. Power Down Timing Diagrams on pages 17 and 18 are for combinatorial operation only.

DEVELOPMENT TOOLS

The PM2552 is supported by the Philips Semiconductors SNAP software development package and a multitude of hardware and software development tools. These include industry standard PLD programmers and CAD software.

SNAP

Features

- Schematic entry using DASH™ 4.0 or above or OrCAD™ SDT III
- State Equation Entry
- Boolean Equation Entry
- Allows design entry in any combination of above formats
- Simulator
  - Logic and fault simulation
  - Timing model generation for device timing simulation
  - Synthetic logic analyzer format
- Macro library for standard TTL and user defined functions
- Device independent netlist generation
- JEDEC fuse map generated from netlist

SNAP (Synthesis, Netlist, Analysis and Program) is a versatile development tool that speeds the design and testing of PML. SNAP combines a user-friendly environment and powerful modules that make designing with PML simple. The SNAP environment gives the user the freedom to design independent of the device architecture.

The flexibility in the variations of design entry methodologies allows design entry in the most appropriate terms. SNAP merges the inputs, regardless of the type, into a high-level netlist for simulation or compilation into a JEDEC fuse map. The JEDEC fuse map can then be transferred from the host computer to the device programer.

SNAP's simulator uses a synthetic logic analyzer format to display and set the nodes of the design. The SNAP simulator provides complete timing information, setup and hold-time checking, plus toggle and fault grading analysis.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. A minimum of 640K bytes of RAM is required together with a hard disk.

DESIGN SECURITY

The PML2552 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary design implemented in the device cannot be copied or retrieved.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DASH is a trademark of Data I/O Corporation.
OrCAD is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

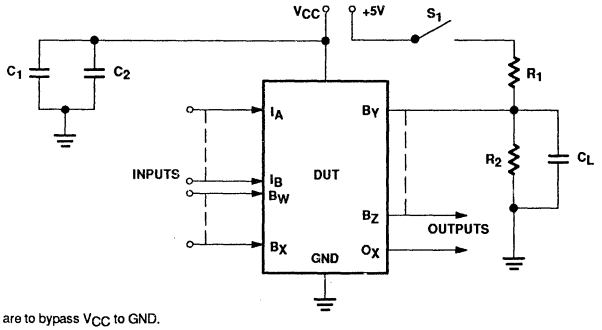
DC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS		LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage							
V_{IL}	Low	V_{CC} = MIN		−0.3		0.8	V
V_{IH}	High	V_{CC} = MAX		2.0		V_{CC} + 0.3	V
Output voltage							
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 5mA				0.45	V
V_{OH}	High	V_{CC} = MIN, I_{OH} = −2mA		2.4			V
Input current							
I_{IL}	Low	V_{IN} = GND				−10	µA
I_{IH}	High	V_{IN} = V_{CC}				10	µA
Output current							
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC}				10	µA
		V_{OUT} = GND				−10	µA
I_{OH}	Output High	V_{CC} = MIN, V_{OUT} = 2.4V				−2	mA
I_{OL}	Output Low	V_{CC} = MIN, V_{OUT} = 0.45V				5	mA
I_{OS}	Short-circuit[5]	V_{OUT} = GND				−100	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX, No load	CMOS input[2]		60	100[6]	mA
		f = 1MHz	TTL input[3]		65	120[6]	mA
I_{SB}	Standby V_{CC} supply current	V_{CC} = MAX, No load	CMOS input		1.0	10	mA
		PD = V_{IH}	TTL input		1.5	10	mA
Capacitance							
C_{IN}	Input	V_{CC} = 5V, T_{amb} = +25°C, V_{IN} = 2.0V			8		pF
C_B	I/O	V_{CC} = 5V, T_{amb} = +25°C, V_{IO} = 2.0V			16		pF

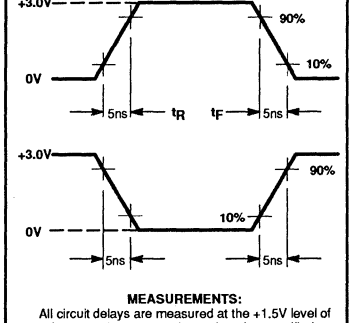
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. CMOS inputs: V_{IL} = GND, V_{IH} = V_{CC}.
3. TTL inputs: V_{IL} = 0.45V, V_{IH} = 2.4V.
4. All voltage values are with respect to network ground terminal.
5. Duration of short-circuit should not exceed one second. Test one at a time.
6. ΔI_{CC} vs. Frequency = 4mA/MHz max.

TEST LOAD CIRCUITS



NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
Test Load R_1 = 750Ω, R_2 = 442Ω, C_L = 30pF (C_L = 5pF for Output Disable)
$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

VOLTAGE WAVEFORMS



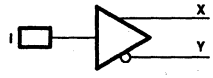
MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

MACRO CELL AC SPECIFICATIONS

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V (SNAP Resource Summary Designations in Parentheses)

Input Buffer
(DIN552, NIN552, BDIN55, BNIN552 CDIN552, CNIN552, CKDIN552, CKNIN552, IDFF552*)



SYMBOL	PARAMETER		LIMITS						UNIT
	To	From	PML2552-35			PML2552-50			
	(Output)	(Input)	MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	X	I	5	7	10	7	10	15	ns
t_{PLH}	X	I	5	7	10	7	10	15	ns
t_{PHL}	Y	I	5	7	10	7	10	15	ns
t_{PLH}	Y	I	5	7	10	7	10	15	ns

* When D flip-flop is bypassed.
Input Pins: 8–14, 16, 17, 20, 22–24.
Bidirectional Pins: 1–3, 5–7, 46–48, 50–54, 57–64, 67, 68.
Bypassed D flip-flop at pins 26, 28–31, 33–35, 37, 39–45.

Internal NAND of Main Array
(FBNAND, NAND)

SYMBOL	PARAMETER		LIMITS			LIMITS			UNIT
	To	From	PML2552-35			PML2552-50			
	(Output)	(Input)	MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Y	X	10	15	20	12	18	25	ns
t_{PLH}	Y	X	10	15	20	12	18	25	ns

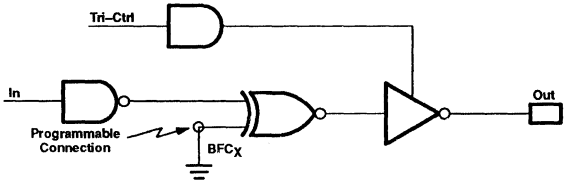
Internal NAND of Clock Array
(NAND)

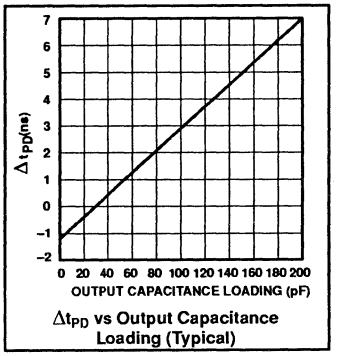
SYMBOL	PARAMETER		LIMITS			LIMITS			UNIT
	To	From	PML2552-35			PML2552-50			
	(Output)	(Input)	MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Y	X	5	7	10	7	10	15	ns
t_{PLH}	Y	X	5	7	10	7	10	15	ns

MACRO CELL AC SPECIFICATIONS (Continued)

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V (SNAP Resource Summary Designations in Parentheses)

3-State Output with Programmable Polarity (TOUT552 + EXOR552)



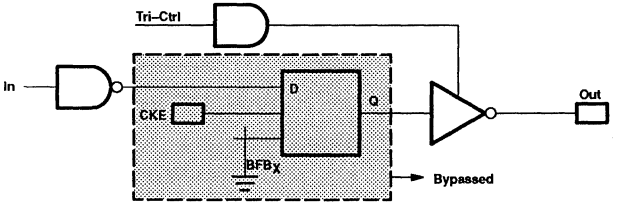


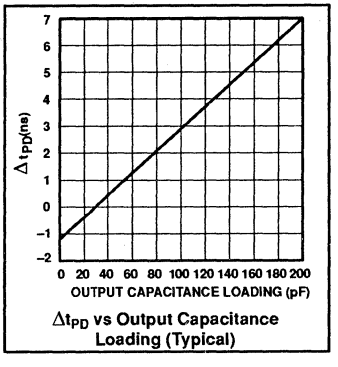
Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2552-35			PML2552-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Out	In	12	18	25	17	25	35	ns
t_{PLH}	Out	In	12	18	25	17	25	35	ns
t_{OE}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns
t_{OD}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns

Bidirectional Pins: 46–48, 50–54.

I/O Output Buffer with 3-State Control, DFF Bypassed (TOUT552 + NAND)





Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2552-35			PML2552-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Out	In	12	18	25	17	25	35	ns
t_{PLH}	Out	In	12	18	25	17	25	35	ns
t_{OE}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns
t_{OD}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns

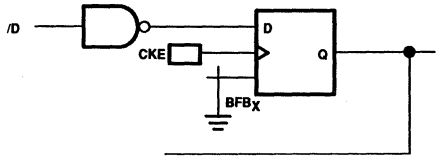
I/O Pins: 1–3, 5–7, 57–64, 67, 68.

Notes on page 481.

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

D FLIP-FLOP

Output DFF Used Internally (ODFF552)



SYMBOL	PARAMETER	LIMITS						UNIT
		PML2552-35			PML2552-50			
		MIN	TYP	MAX	MIN	TYP	MAX	
f_{CKE}	Flip-flop toggle rate			50			35	MHz
$t_{W\,CKE}$ High	Clock HIGH	10			14			ns
$t_{W\,CKE}$ Low	Clock LOW	10			14			ns
t_{SETUP} /D	/D setup time to CKE	15			20			ns
t_{HOLD} /D	/D hold time to CKE	4			6			ns

SYMBOL	PARAMETER		LIMITS						UNIT
	From	To	PML2552-35			PML2552-50			
	(Input)	(Output)	MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CKE ↑	Q	10	15	20	14	20	25	ns
t_{PHL}	CKE ↑	Q	10	15	20	14	20	25	ns

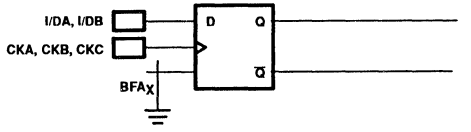
MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

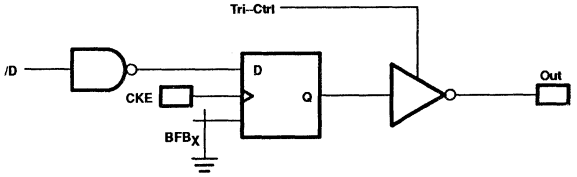
D FLIP-FLOP (Continued)

Input and Output (IDFF552 & ODFF552)

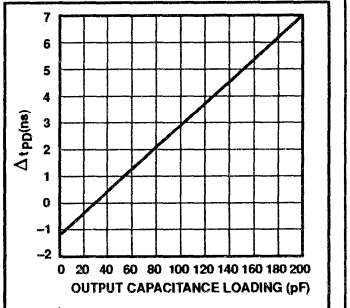
INPUTS		OUTPUTS	
CK	D	Q	$\overline{Q}$
L	X	Q_0	$\overline{Q}_0$
↑	H	H	L
↑	L	L	H

NOTE:
Q_0, $\overline{Q}_0$ represent previous stable condition of Q, $\overline{Q}$.





SYMBOL	LIMITS						UNIT
	PML2552-35			PML2552-50			
	MIN	TYP	MAX	MIN	TYP	MAX	
$f_{CKA, CKB, CKC}$			50			35	MHz
$t_{W\ CKA, CKB, CKC}$ High	10			14			ns
$t_{W\ CKA, CKB, CKC}$ Low	10			14			ns
t_{SETUP} I/DA, I/DB	5			7			ns
t_{HOLD} I/DA, I/DB	5			7			ns
f_{CKE}			50			35	MHz
$t_{W\ CKE}$ High	10			14			ns
$t_{W\ CKE}$ Low	10			14			ns
t_{SETUP} /D	15			20			ns
t_{HOLD} /D	4			6			ns



Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	From (Input)	To (Output)	PML2552-35			PML2552-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	5	7	10	7	10	15	ns
t_{PHL}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	5	7	10	7	10	15	ns
t_{PLH}	CKE ↑	Out	12	18	25	17	25	35	ns
t_{PHL}	CKE ↑	Out	12	18	25	17	25	35	ns

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

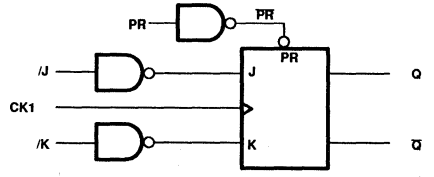
JK FLIP-FLOPS

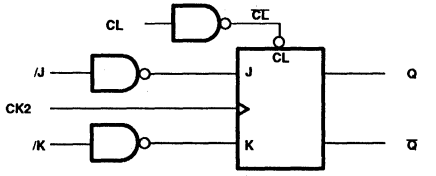
(JKPR552)

INPUTS				OUTPUTS	
$\overline{PR}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	H	L
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$

(JKCL552)

INPUTS				OUTPUTS	
$\overline{CL}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	L	H
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$





SYMBOL	PARAMETER	LIMITS						UNIT
		PML2552-35			PML2552-50			
		MIN	TYP	MAX	MIN	TYP	MAX	
f_{CK1}	CK1 toggle frequency			50			35	MHz
f_{CK2}	CK2 toggle frequency			50			35	MHz
$t_{W\,CK1}$ High	CK1 clock HIGH	10			14			ns
$t_{W\,CK1}$ Low	CK1 clock LOW	10			14			ns
$t_{W\,CK2}$ High	CK2 clock HIGH	10			14			ns
$t_{W\,CK2}$ Low	CK2 clock LOW	10			14			ns
t_{SETUP} /J, /K	/J, /K setup time to CK1, CK2	27			35			ns
t_{HOLD} /J, /K	/J, /K hold time to CK1, CK2	0			0			ns
t_W PR Low	Preset Low period	10			14			ns
t_W CL Low	Clear Low period	10			14			ns

SYMBOL	PARAMETER		LIMITS						UNIT
	From (Input)	To (Output)	PML2552-35			PML2552-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CK1,2	Q, $\overline{Q}$	2	3.5	5	3	5	7	ns
t_{PHL}	CK1,2	Q, $\overline{Q}$	2	3.5	5	3	5	7	ns
t_{PLH}	PR	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PHL}	PR	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PLH}	CL	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PHL}	CL	Q, $\overline{Q}$	12	18	25	17	24	30	ns

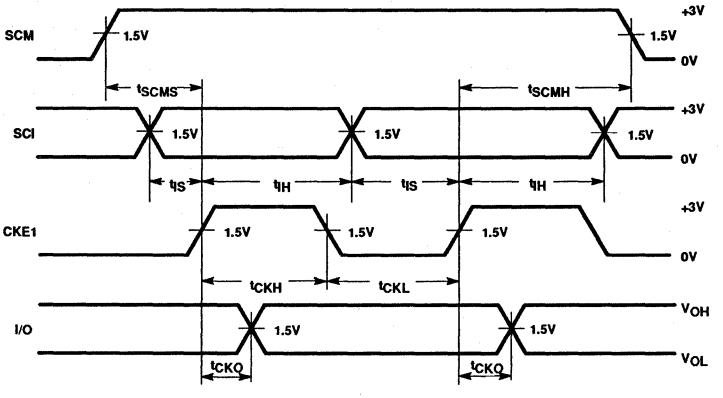
AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$, $V_{PP} = V_{CC}$,
$R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$ for Output Disable) (See Test Load Circuit Diagram)

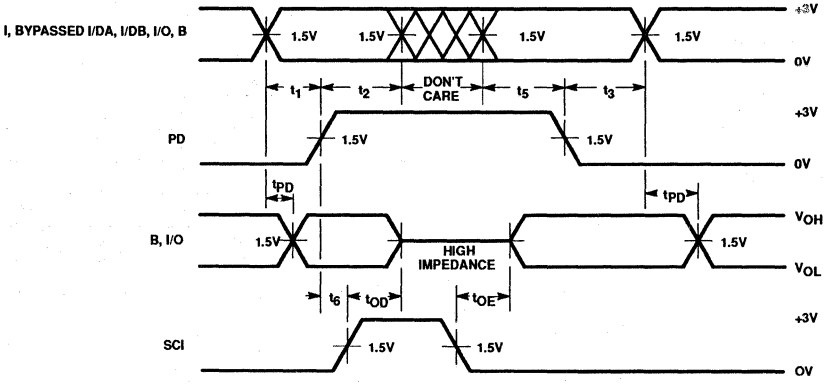
SYMBOL	PARAMETER	LIMITS				UNIT
		PML2552-35		PML2552-50		
		MIN	MAX	MIN	MAX	
Scan mode operation[1]						
t_{SCMS}	Scan Mode (SCM) Setup time	15		15		ns
t_{SCMH}	Scan Mode (SCM) Hold time	25		30		ns
t_{IS}	Data Input (SCI) Setup time	5		5		ns
t_{IH}	Data Input (SCI) Hold time	5		5		ns
t_{CKO}	Clock to Output (I/O) delay		30		40	ns
t_{CKH}	Clock High	10		15		ns
t_{CKL}	Clock Low	10		15		ns
Power down, power up[2]						
t_1	Input (I, bypassed I/DA, I/DB, I/O, B) setup time before power down	40		50		ns
t_2	Input hold time	30		35		ns
t_3	Power Up recovery time		60		70	ns
t_4	Output hold time	0		0		ns
t_5	Input setup time before Power Up	20		25		ns
t_{OE}	SCI to Output Enable time[3]		40		50	ns
t_{OD}	SCI to Output Disable time[3]		40		50	ns
t_6	Power Down setup time	10		15		ns
t_7	Power Up to Output valid		70		80	ns
Power-on reset						
t_{PPR1}	Power-on reset output register (Q = 0) to output (I/O) delay		10		15	ns
t_{PPR2}	Power-on reset input register (Q = 0), buried JK Flip-Flop (Q = 0) to output (B, bypassed I/O) delay		40		50	ns

NOTES:
1. SCM recovery time is 50ns after SCM operation. 50ns after SCM operation, normal operations can be resumed.
2. Timings are measured without foldbacks.
3. Transition is measured at steady state High level (–500mV) or steady state Low level (+500mV) on the output from 1.5V level on the input with specified test load ($R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$). This parameter is sampled and not 100% tested.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS

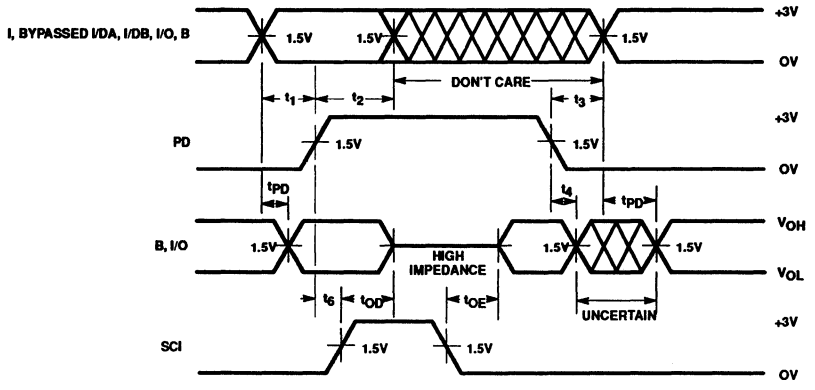


Scan Mode Operation

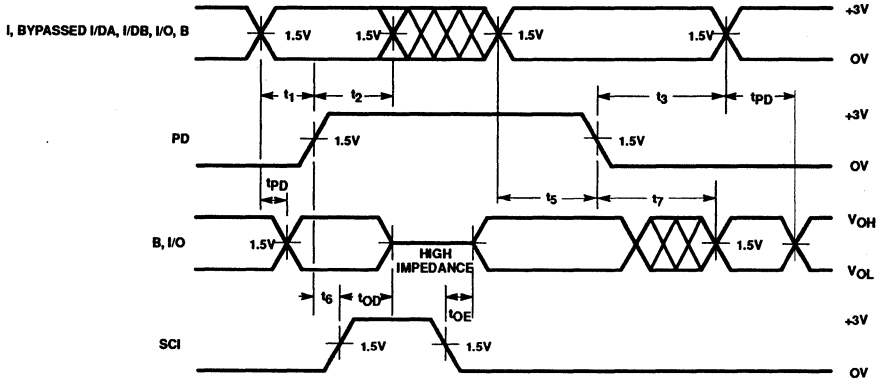


Power Down, Power Up
Input (old) Ready Before Power Up

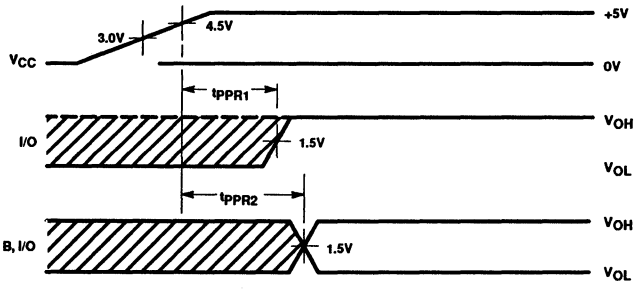
TIMING DIAGRAMS (Continued)



Power Down, Power Up
Input (new) Ready After Power Up

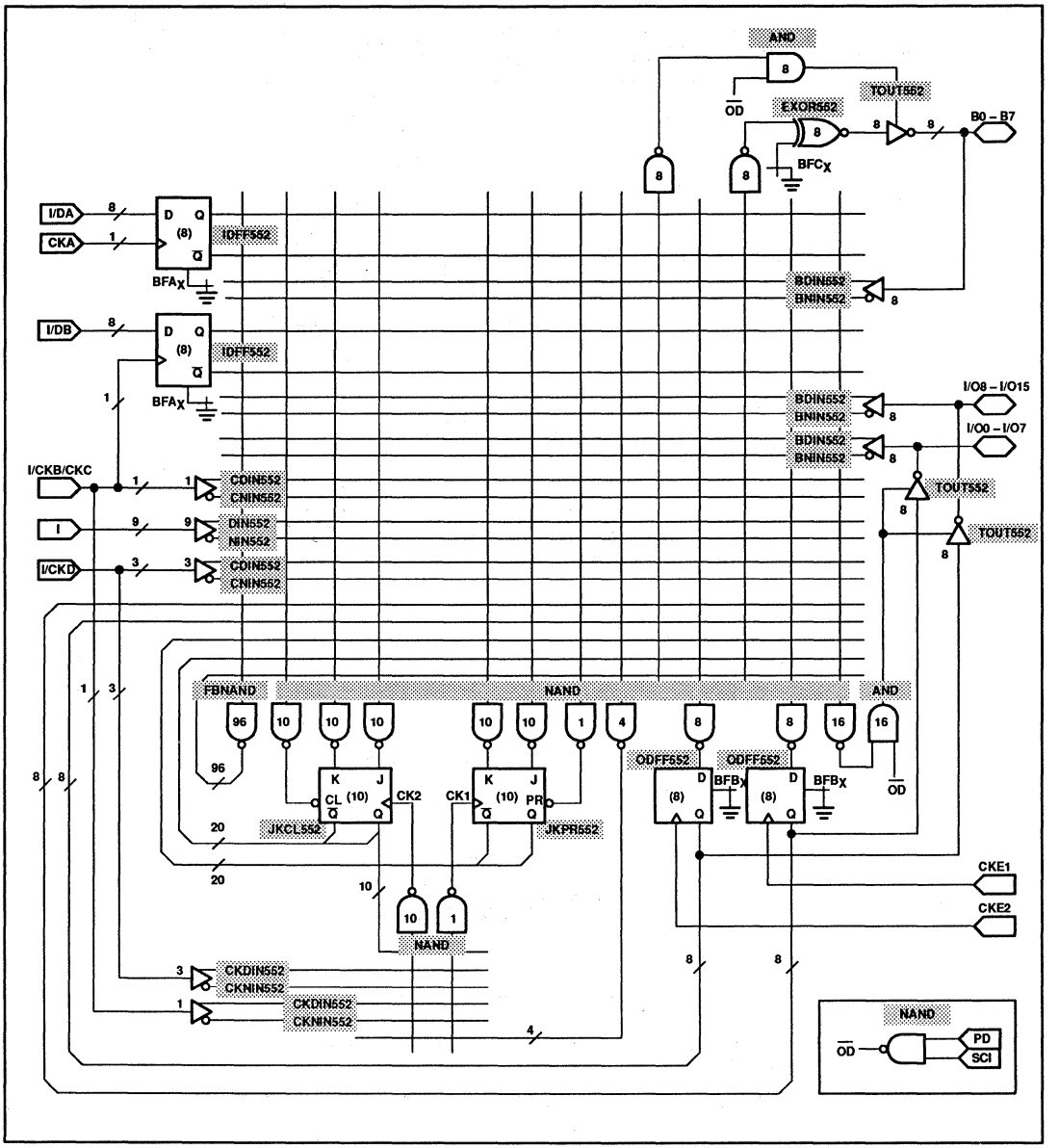


Power Down, Power Up
Input (new) Ready Before Power Up



Power-On Reset

SNAP RESOURCE SUMMARY DESIGNATIONS



ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PML2552 device is such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PML2552 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PML2552 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PML2552 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12,000µW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING/SOFTWARE SUPPORT

Refer to Section 9 *(Development Software)* and Section 10 *(Third-party Programmer/ Software Support)* of this data handbook for additional information.

CMOS high density programmable macro logic — PML2852

FEATURES

- Wide gates for efficient product term use
- Multiple I/O pins for 16–32 bit buses or up to 32-bit data flow
- Multiple I/O pins for multiple-port data handling
- Multiple clocks for independent state machines and storage banks
- 100% connectible, no place and route restrictions
- Erasable and one time programmable versions available
- Scan test
- Low CMOS power dissipation = 525mW max.
- Power down mode (52mW max.)
- Power on reset
- Security fuse for copy protection
- Supported by advanced SNAP and SLICE development systems

PERFORMANCE

- 35ns max. pin-to-pin for 32-bit decoders
- 40ns max. internal, 55ns max. pin-to-pin for 16-bit multiplexers
- 33MHz max. throughput for 16-bit latches
- 18–50MHz max. for 10-bit counters
- 31MHz max. for 10-bit shift registers
- 15ns (typ.) delay for internal NANDs
- 50MHz max. flip-flop toggle rate

APPLICATIONS

- Bus interface and control (microchannel, VME, NuBus, etc.)
- Microcomputer peripheral interface and control (printers, SCSI, hard disk drives, etc.)
- Multiport memory control and arbitration (cache, DRAM, VRAM, etc.)
- Intelligent instrumentation (data acquisition, testers, medical equipment, etc.)
- Industrial control (process control, motor control, engine control, etc.)
- Communication network control (LAN, Ethernet, T1, TDMA, etc.)
- General purpose logic integration
- Laptops, pocket computers, and handheld instruments
- Low-end gate array replacement for quick prototyping

SNAP DEVELOPMENT SYSTEM

- Supports third-party schematic entry formats
- Versatile EDIF-compatible netlist format for design portability
- TTL macro library for automatic mapping
- Logic, timing, and fault simulation
- Automatic test vector generator
- Espresso logic minimizer
- Boolean equation extractor from JEDEC fusemap

DESCRIPTION

The Philips Semiconductors family of Programmable Macro Logic is optimized for handling wide buses, wide datapaths, and multiple-port applications with the highest throughputs among high density PLDs and FPGAs. The PML2852 now expands Philips Semiconductors CMOS PML product offering into the 32-bit arena. Fabricated with a high-performance EPROM process, the PML2852 is ideal in today's bus interface control, microprocessor peripheral control, memory interface, communications, instrumentation, and industrial control. It is capable of replacing large amounts of TTL SSI and MSI logic, and literally integrates a complete custom microcontroller.

The PML2852 incorporates the folded NAND array architecture, which provides 100% connectivity to eliminate the routing restrictions associated with other high density PLD/FPGA architectures. The array of wide-input NAND gates enables the designer to implement any wide-gate logic function, from decoders to multiplexers, with no more than two gate-level delays. It also allows implementation of multiple levels of logic within the chip, without wasting I/O pins. Its flexible and potent flip-flop building blocks provide for high throughput data storage, high speed state machines, and fast counters.

The PML2852 also incorporates two unique features: scan test and power down. With user-controlled scan test, the PML2852 significantly reduces system functional test time by providing access to all of its internal registers. In the user-controlled power down mode, the PML2852 power dissipation is reduced to a mere 52mW, making it ideal for laptop or pocket computers and handheld instruments.

Thanks to its high density and its flexible architecture, the PML2852 provides **instant gate array** capabilities for all general purpose logic integration. As such, the PML2852 eliminates the NRE costs, risks, inventory problems, and hard to use design tools associated with semicustom and full custom approaches. It allows the designer to quickly bring concepts to silicon for faster learning cycles and a much shorter time to market. Functional prototypes are available within minutes.

The SNAP development software is designed to fully exploit the flexibility and density of the PML2852. It accepts a variety of design entry formats, including schematic, logic equations, and state equations in any combination for maximum flexibility. Its powerful features, but ease of use, allows literally push-button operation.

Together, the PML2852 and SNAP constitute the designer's personal **desktop silicon foundry**.

ORDERING INFORMATION

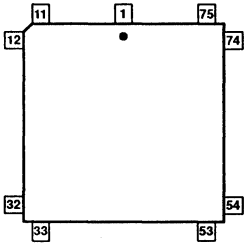
DESCRIPTION	t_{PD} (MAX)	ORDER CODE	DRAWING NUMBER
84-pin Plastic Leaded Chip Carrier	35ns	PML2852-35A	0399F
84-pin "J" Leaded Ceramic Cerquad Package	35ns	PML2852-35KA	1551
84-pin Plastic Leaded Chip Carrier	50ns	PML2852-50A	0399F
84-pin "J" Leaded Ceramic Cerquad Package	50ns	PML2852-50KA	1551

NuBus is a trademark of Texas Instruments, Inc.

CMOS high density programmable macro logic | PML2852

PIN CONFIGURATION

A and KA Packages



Pin	Function	Pin	Function	Pin	Function
1	I/O5	29	PD	57	O8
2	I/O4	30	I/DB7	58	B7
3	I/O3	31	I/DB6	59	B6
4	VCC1	32	I/DB5	60	B5
5	I/O2	33	VCC5	61	VSS3
6	I/O1	34	I/DB4	62	B4
7	I/O0	35	I/DB3	63	B3
8	O3	36	I/DB2	64	B2
9	O2	37	I/DB1	65	B1
10	O1	38	I/DB0	66	B0
11	O0	39	I/DA7	67	VCC4
12	I8	40	VSS2	68	CKE2
13	I7	41	I/DA6	69	I/O15
14	I6	42	I/DA5	70	I/O14
15	I5	43	I/DA4	71	O7
16	I4	44	CKA	72	O6
17	I3	45	I/DA3	73	O5
18	I2	46	VCC3	74	O4
19	VSS1	47	I/DA2	75	I/O13
20	I1	48	I/DA1	76	I/O12
21	I0	49	I/DA0	77	I/O11
22	SCM	50	O15	78	I/O10
23	SCI	51	O14	79	I/O9
24	I/CKD3	52	O13	80	I/O8
25	VCC2	53	O12	81	CKE1
26	I/CKD2	54	O11	82	VSS4
27	I/CKD1	55	O10	83	I/O7
28	I/CKB/CKC	56	O9	84	I/O6

FUNCTIONAL BLOCK DIAGRAM

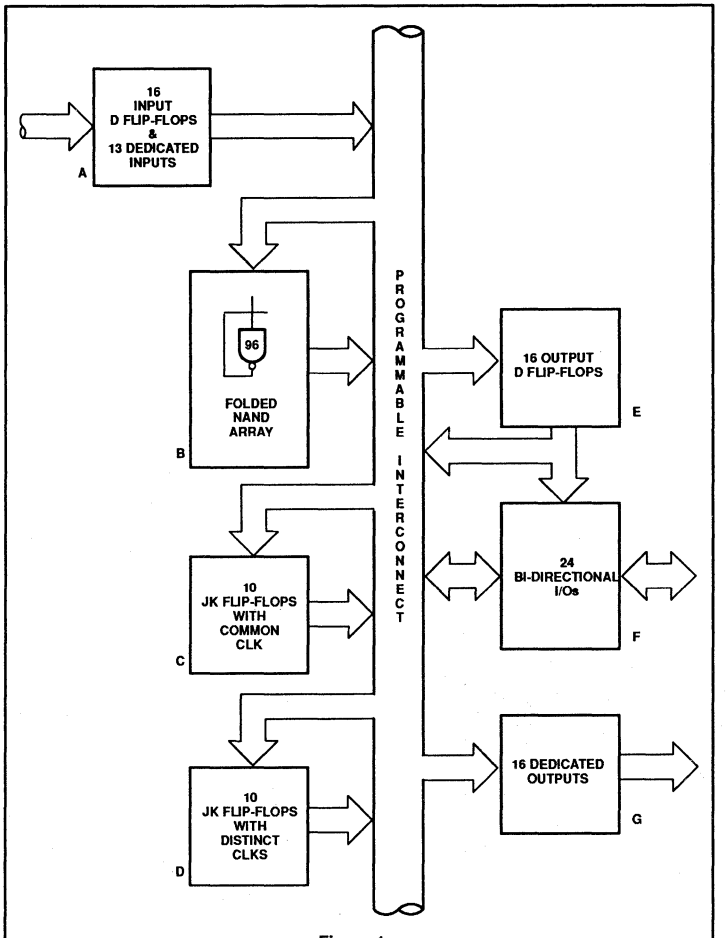
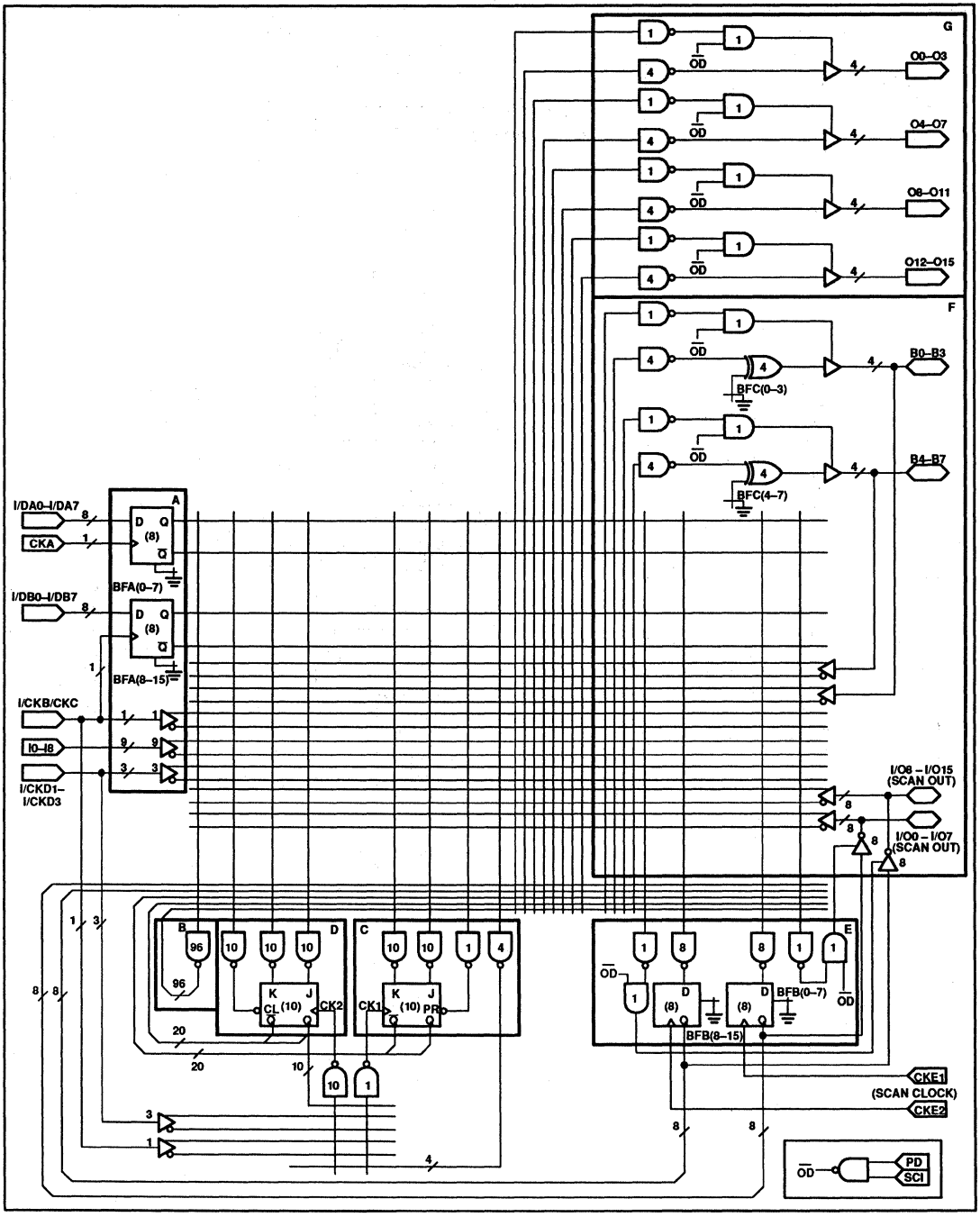


Figure 1.

LOGIC DIAGRAM



CMOS high density programmable macro logic — PML2852

STRUCTURE

- 112 possible foldback NAND gates:
  - 96 internal NAND
  - 16 from the I/O macros
- 114 additional logic terms
- 53 possible inputs (with programmable polarity)
  - 29 dedicated inputs
  - 24 bidirectional I/Os
- 24 bidirectional pins
- 16 dedicated output pins
- 52 flip-flops
- 40 possible outputs with Output Enable control (8 with programmable polarity)
- Multiple independent clocks
- 20 Buried JK-type flip-flops with foldback (JKFFs):
  - 10 JKFFs with one shared preset signal and one shared clocked signal originating from the clock array.
  - 10 JKFFs with 10 independent clock signals originating from the clock array and 10 independent clear signals
- 258 inputs per NAND gate
- Bypassable Input D-type flip-flop (DFFs)/Combinatorial Inputs:
  - 16 DFFs/combinatorial inputs
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
- Inputs/bypassable D-type flip-flop outputs/foldback NAND gates:
  - 16 output DFFs/combinatorial inputs/outputs with individual Output Enable control
  - DFFs clocked in two groups of eight
  - DFFs not bypassed in unprogrammed state
  - Independent bypass fuse on each DFF
  - The DFF can be used as an internal DFF or an internal foldback NAND gate.
- Combinatorial inputs:
  - 9 dedicated inputs to the NAND array
  - 3 inputs optional to NAND array and/or clock array
  - 1 input optional to NAND array and/or clock array, and/or clock of Input D Flip-Flops (Group B)
- Separate clock array:
  - Separate clock array for JKFFs clock inputs
  - 4 inputs to clock array originated from NAND array
  - 4 inputs (with programmable polarity) directly from input pins
  - 10 inputs from Q outputs of JKFFs with clear
- Dedicated clocks:
  - One dedicated clock for input DFFs (Group A)
  - Two dedicated clocks for output DFFs (Group E)
- Scan test feature:
  - Scan chain is implemented through the 20 buried JKFFs and 16 output DFFs
  - Pins SCI, SCM, and CKE1 are used to operate the scan test
- Power down mode
  - Dedicated pin (PD) freezes the circuit when brought to logic "1". The circuit remains in the same state prior to the logic "0" to logic "1" transition of the "PD" pin.
  - When in the power down mode, the SCI pin acts as the 3-State pin for the 40 outputs.
- Power on reset:
  - All flip-flops (16 input DFFs, 20 buried JKFFs, and 16 output DFFs) are reset to logic "0" after V_{CC} power on.

ARCHITECTURE

The core of the PML2852 is a programmable NAND array of 96 NAND gates and 20 buried JKFFs. The output of each NAND gate folds back upon itself and all other NAND gates and flip-flops. The 'Q' and '$\overline{Q}$' output of each flip-flop also folds back in the same manner. Thus, total connectivity of all logic functions is achieved in the PML2852. Any logic function can be created within the core without wasting valuable I/O pins. Furthermore, a speed advantage is acquired by implementing multi-level logic within a fast internal core without incurring any delays from the I/O buffers. Figure 1 shows the functional block diagram of the PML2852.

Macro Cells

There are 16 bypassable DFFs on the input to the NAND array. These flip-flops are split in two banks of 8 (Bank A and Bank B). Each bank of flip-flops has a common clock. In the unprogrammed state of the device the flip-flops are active. In order to bypass any DFF, its respective bypass fuse (BFA_X) must be programmed.

The 16 I/O pins (IO_0 - IO_{15}) and their respective D flip-flop macros can be used in any one of the following configurations:

1. As combinatorial input(s):
   Each of the 16 3-State outputs can be individually disabled by the associated NAND term and the pin is used as an inverting or non-inverting input.
2. As registered DFF outputs:
   These DFFs are split into two banks of 8, and each bank is clocked separately. The bypass fuse BFB_X (see PML2552 Logic Diagram) is used to bypass any one of these DFFs. The flip-flops are all active in an unprogrammed device.
3. As combinatorial outputs:
   By programming the bypass (BFB_X) fuse of any one of the DFFs, the flip-flop(s) is bypassed. The I/O pin can then be used as a combinatorial output.
4. As Internal foldback DFFs or foldback NAND gates:
   When the I/O pin is used as an input, the output macro can be used as a buried DFF or a foldback NAND term. If the bypass fuse is programmed, the macro will act as a foldback NAND term. Otherwise it will act as a buried DFF.

The 8 bidirectional pins (B0-B7) can be used as either combinatorial inputs or outputs with programmable polarity. The Exclusive-OR polarity gates are non-inverting in the unprogrammed state.

The NAND signal labeled '$\overline{OD}$' (Output Disable) shown on the PML2852 logic diagram is used for the Power Down mode operation. This signal disables the outputs when the device enters the Power Down mode and SCI is high.

Clock Array

The 20 buried JKFFs are clocked through the 'Clock Array'. The Clock Array consists of 11 NAND terms. Ten of these terms are connected to the clock inputs of the Bank A flip-flops that can be clocked individually. One NAND gate is connected to Bank B flip-flops that have a common clock. There are 18 inputs to the clock array. Four come directly from the input pins (with programmable polarity), 4 inputs are from 4 NAND gates connected directly to the folded NAND array. 10 inputs are from the Q outputs of the JKFFs with clear.

SCAN TEST FEATURE

With the rise in the ratio of devices on a chip to the number of I/O pins, Design For Testability is becoming an essential factor in logic design methodology. The PML2852 incorporates a variable length scan test feature which permits access to the internal flip-flop nodes without requiring a separate external I/O pin for each node accessed. Figure 2 (Scan Mode Operation) shows how a scan chain is implemented through the 20 buried JKFFs and 16 output DFFs. Two dedicated pins, SCI (Scan In) and SCM (Scan Mode), are used to operate the scan test. The SCM pin is used to put the circuit in scan mode. When this pin is brought to a logic "1", the circuit enters the scan mode.

In this mode it is possible to shift an arbitrary test pattern into the flip-flops. The SCI pin is used to input the pattern. The inverted outputs of flip-flops D0 - D15 are observable on pins I/O0 - I/O15.

The following are features and characteristics of the device when in Scan Mode:

1. CKE1 is the common scan-clock for all the flip-flops when in scan mode. CKE1 overrides all clock resources of normal operational mode.
2. The Preset (PR) and Clear (CL) functions of the flip-flops are disabled.
3. Scan overrides the bypass fuse of the flip-flops. This means that all the bypassable DFFs remain intact during scan operation even though they may have been bypassed during normal operation.
4. To observe the SCAN data, the output buffers must be enabled by the Output Enable (tri-ctrl) terms.
5. The outputs of the flip-flops are complemented on pins I/O0 - I/O15.
6. All external inputs to flip-flops in the scan chain are disabled when the device enters the scan mode.
7. Blowing the security fuse does not disable the Scan Test feature.

SCAN MODE OPERATION

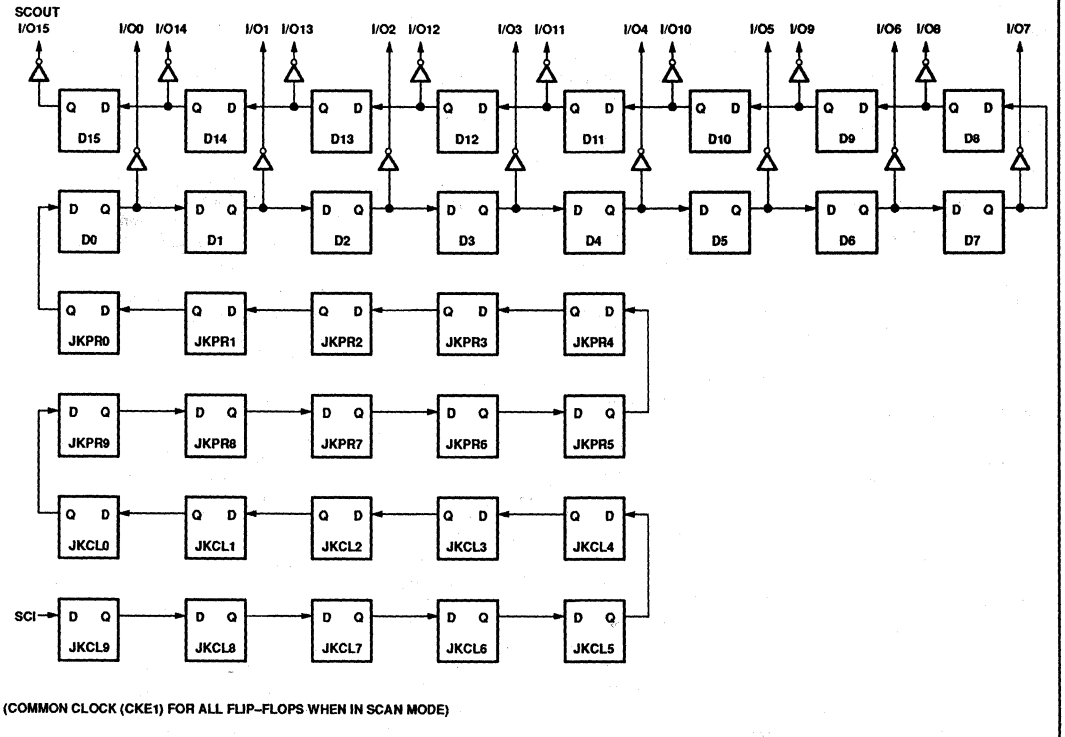


Figure 2.

SCAN TEST STRATEGY

The scan test pattern is design dependent and the user must make considerations for Design For Testability (DFT) during the initial stages of the design. A typical test sequence is to pre-load (i.e., enter a state); revert to normal operation (i.e., activate the next state transition); go back to scan mode to check the result. Note that the scan test feature available in the PML2852 is a variable length scan chain. The DATA entered at SCI (JKCL9) can be accessed anywhere between 21 clock cycles (at I/O0) to 36 clock cycles (at I/O15). For the strategy discussed here, DATA is read out after 36 clocks at I/O15 (i.e., D15). The following operation sequence suggests a possible scan test method.

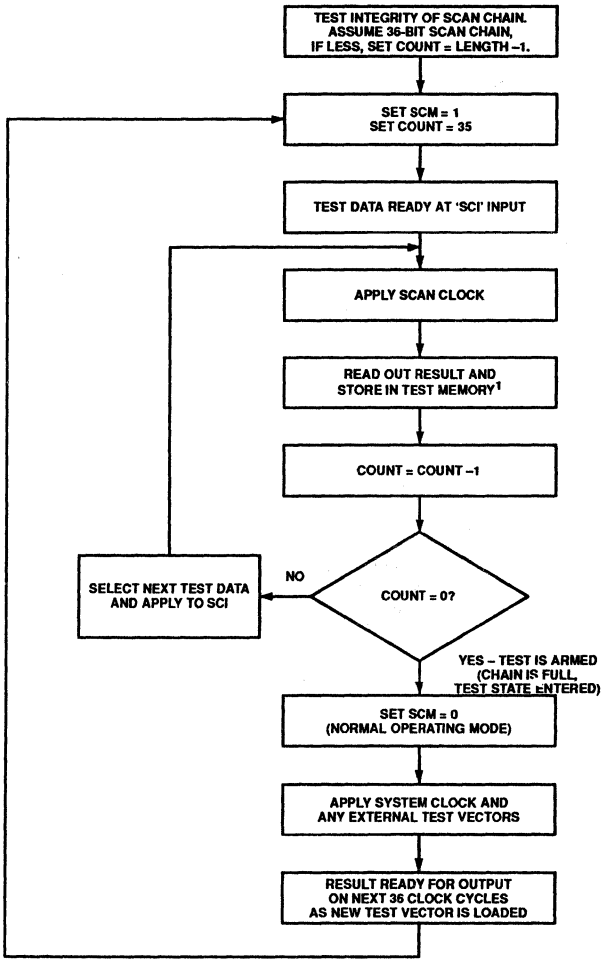
A conservative test policy demands proof that the test facility is working. Thus, to prove Scan Chain holds and maintains correct data:

a. Fill chain with several patterns (for example, all ones and all zeros).

b. Retrieve same patterns.

The user is responsible for managing an external test memory buffer for applied vectors and results, as part of the test equipment.

1. Parallel readout of I/O0 - I/O15 is possible, but assume only I/O15 is used for this strategy.
2. The first DATA entered at SCI (or JKCL9) will be the content of D15 after 36 clocks. This DATA will be inverted at the output pin I/O15 (i.e., SCOUT). The last DATA entering the scan chain will be the content of JKCL9. Thus, the scan chain resembles a first-in-first-out shift register with inverted outputs (I/O0 - I/O15).
3. 'Test Data' is read in at the SCI input and read out of the SCOUT output pin (I/O15). To enter 'Test Data':
   a. Put device in Scan Mode by applying the scan control signals (SCM=1).
   b. Clock device with scan clock (CKE1).
   c. Apply consecutive serial test vectors.
   d. Read back results as new 'Test Data' (States) are applied. The first 36 outputs read at SCOUT (I/O15) are random ('old') data (e.g., remnant of Step 1).
   e. Apply 36 'Test Data' until the chain is full.
4. To apply 'Test Data' (States), exit Scan Mode and apply on system clock together with any other possible test vectors.
5. To read result of the state transition, re-enter scan and apply the scan clock (CKE1). The result of the state transition in JKCL9 will be available at SCOUT (I/O15) after 36 clocks. The results can be stored in a user defined test memory buffer in inverted logic representation.
6. As the results are being read and stored, new 'Test Data' can be entered via SCI.
7. Repeat for all test patterns of interest.
8. Figure 3 (FLOW_CHART) depicts a flow chart version of the test sequence.



NOTE:
1. The first 36 outputs are random ('OLD') data.

Figure 3. FLOW_CHART

A Simple Example

Assume the last three cells of the scan chain (JKCL9, JKCL8, JKCL7 in Figure 4 contain a 3-bit up counter. Our test vector will be a single clock applied to the counter. Suppose we wish to first check the *State 5* (i.e., 101) to *State 6* (i.e., 110) transition, then the *State 3* (i.e., 011) to *State 4* (i.e., 100) transition. Assume the scan chain has been pre-verified and we may begin.

Enter scan mode (set SCM=1)I apply 36 bits in sequence so that the value *101* (i.e., State 5) resides in the last three cells. Exit scan mode (set SCM=0) and apply a single clock to the counter. Now the value *110* (i.e., State 6) resides in the last three cells. Re-enter scan mode (set SCM=1) and read back 36 bits from position I/O15. Note that the outputs are complemented and are also read back in the reverse order. Therefore the value for *STATE 6* read at I/O15 will be 100 which is the complement of *STATE 6* (110) read in the reverse order.

As this is being read back, apply a new state, serially equal to the value *011* (i.e., State 3). This state should be loaded on the last three clock cycles during which *STATE 6* is being read back at I/O15. After *STATE 3* has been loaded (and *STATE 6* read back), exit scan mode and apply a single clock which will invoke the *STATE 3* (i.e., 011) to *STATE 4* (i.e., 100) transition. Re-enter scan mode and read back 36 bits at I/O15. The last three bits should contain *110* which is the complement of *State 4* read in the reverse order. 4 (SCAN_EXAMPLE) shows a flow diagram of this example. Note that the States will always be complemented and read back in the reverse at I/O15. Other sequences may be applied in the same manner.

A possible alternative to this example is to read back the output states at I/O0 (D0) instead of I/O15 (JKCL9). This will allow the outputs to be read back after 21 clock cycles rather than the 36 used in the above example.

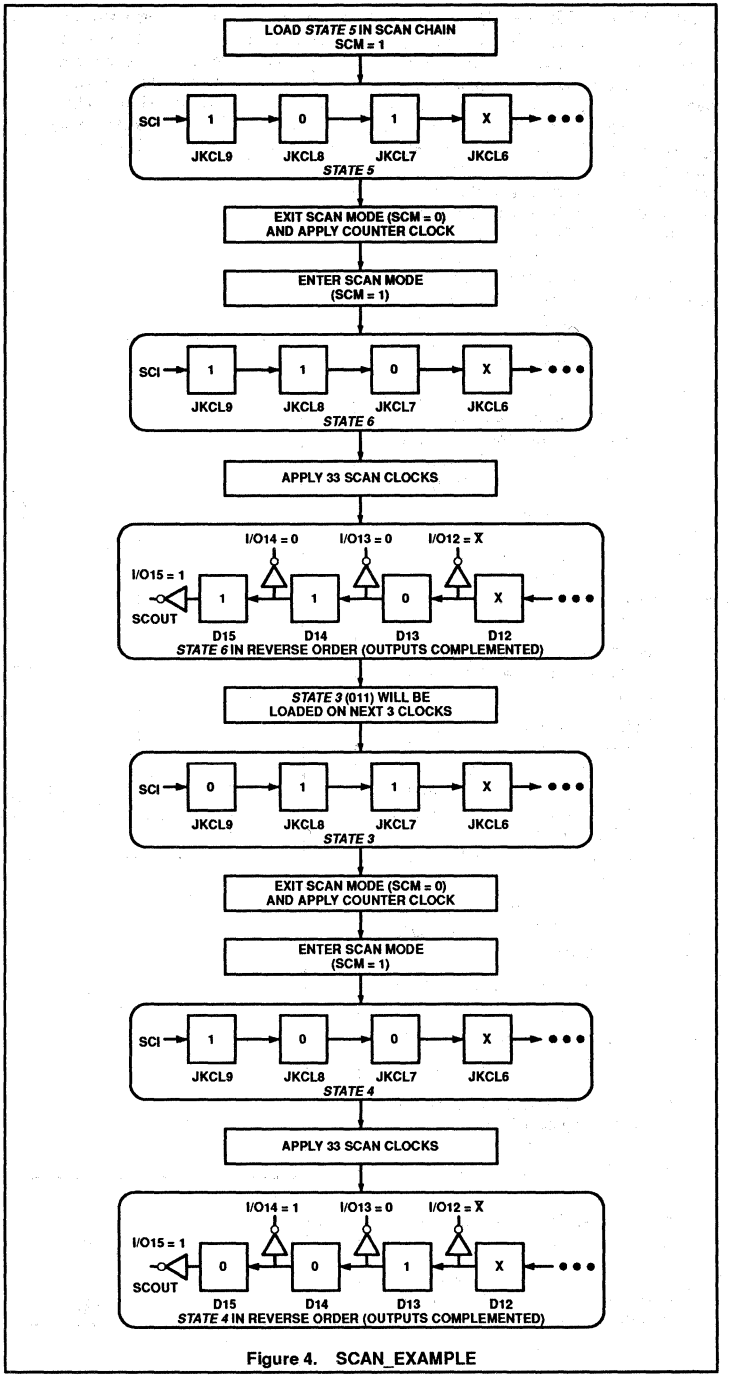


Figure 4. SCAN_EXAMPLE

POWER DOWN

The PML2852 offers the user controlled capability of putting the device to "sleep" where power dissipation is reduced to very low levels. When brought to a logic "1", the PD pin freezes the circuit while reducing the power. All data is retained. This not only includes that of the registers, but also the state of each foldback gate. For those cases where it is desirable to 3-State the outputs, that can be accomplished by raising the SCI pin to a logic "1".

There is one point that should be noted while the circuit is in its power-down mode. The switching of any external clock pin will cause a disruption of the data. All clocks must be frozen before the circuit goes into power-down and stay that way until it powered back up. Clocks that are internally generated and feed the clock array are automatically stopped by the power-down circuitry. Any other input can toggle without any loss of data.

NOTE:

1. During power down, external clocks (CKA, CKB/CKC, CKE1, CKE2) should not change.
2. SCM must be "0" as in normal operation mode.
3. External clock recovery time (low-to-high) is 60ns (high-speed) and 70ns (standard) after the device is powered up.
4. Power Down Timing Diagrams on pages 502 and 503 are for combinatorial operation only.

DEVELOPMENT TOOLS

The PM2852 is supported by the Philips Semiconductors SNAP software development package and a multitude of hardware and software development tools. These include industry standard PLD programmers and CAD software.

SNAP

Features

- Schematic entry using DASH™ 4.0 or above or OrCAD™ SDT III
- State Equation Entry
- Boolean Equation Entry
- Allows design entry in any combination of above formats
- Simulator
  - Logic and fault simulation
  - Timing model generation for device timing simulation
  - Synthetic logic analyzer format
- Macro library for standard TTL and user defined functions
- Device independent netlist generation
- JEDEC fuse map generated from netlist

SNAP (Synthesis, Netlist, Analysis and Program) is a versatile development tool that speeds the design and testing of PML. SNAP combines a user-friendly environment and powerful modules that make designing with PML simple. The SNAP environment gives the user the freedom to design independent of the device architecture.

The flexibility in the variations of design entry methodologies allows design entry in the most appropriate terms. SNAP merges the inputs, regardless of the type, into a high-level netlist for simulation or compilation into a JEDEC fuse map. The JEDEC fuse map can then be transferred from the host computer to the device programer.

SNAP's simulator uses a synthetic logic analyzer format to display and set the nodes of the design. The SNAP simulator provides complete timing information, setup and hold-time checking, plus toggle and fault grading analysis.

SNAP operates on an IBM® PC/XT, PC/AT, PS/2, or any compatible system with DOS 2.1 or higher. A minimum of 640K bytes of RAM is required together with a hard disk.

DESIGN SECURITY

The PML2852 has a programmable security fuse that controls the access to the data programmed in the device. By using this programmable feature, proprietary design implemented in the device cannot be copied or retrieved.

ABSOLUTE MAXIMUM RATINGS[1]

SYMBOL	PARAMETER	RATINGS	UNIT
V_{CC}	Supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_{OUT}	Output voltage	+5.5	V_{DC}
I_{IN}	Input currents	–30 to +30	mA
I_{OUT}	Output currents	+100	mA
T_{amb}	Operating temperature range	0 to +75	°C
T_{stg}	Storage temperature range	–65 to +150	°C

NOTE:

1. Stresses above those listed may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.

THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

DASH is a trademark of Data I/O Corporation.
OrCAD is a trademark of OrCAD, Inc.
IBM is a registered trademark of International Business Machines Corporation.

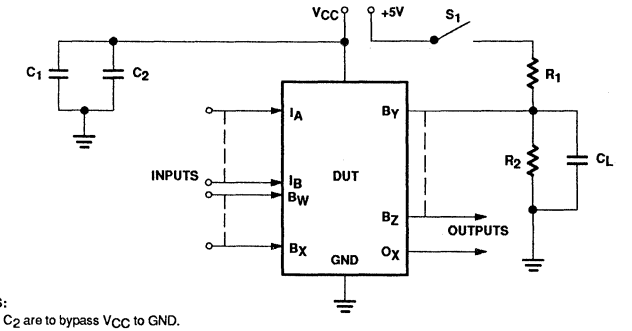
DC ELECTRICAL CHARACTERISTICS

$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

SYMBOL	PARAMETER	TEST CONDITIONS		LIMITS MIN	LIMITS TYP[1]	LIMITS MAX	UNIT
Input voltage							
V_{IL}	Low	V_{CC} = MIN		–0.3		0.8	V
V_{IH}	High	V_{CC} = MAX		2.0		V_{CC} + 0.3	V
Output voltage							
V_{OL}	Low	V_{CC} = MIN, I_{OL} = 5mA				0.45	V
V_{OH}	High	V_{CC} = MIN, I_{OH} = –2mA		2.4			V
Input current							
I_{IL}	Low	V_{IN} = GND				–10	µA
I_{IH}	High	V_{IN} = V_{CC}				10	µA
Output current							
$I_{O(OFF)}$	Hi-Z state	V_{OUT} = V_{CC}				10	µA
		V_{OUT} = GND				–10	µA
I_{OH}	Output High	V_{CC} = MIN, V_{OUT} = 2.4V				–2	mA
I_{OL}	Output Low	V_{CC} = MIN, V_{OUT} = 0.45V				5	mA
I_{OS}	Short-circuit[5]	V_{OUT} = GND				–100	mA
I_{CC}	V_{CC} supply current	V_{CC} = MAX, No load	CMOS input[2]		60	100[6]	mA
		f = 1MHz	TTL input[3]		65	120[6]	mA
I_{SB}	Standby V_{CC} supply current	V_{CC} = MAX, No load	CMOS input		1.0	10	mA
		PD = V_{IH}	TTL input		1.5	10	mA
Capacitance							
C_{IN}	Input	V_{CC} = 5V, T_{amb} = +25°C, V_{IN} = 2.0V			8		pF
C_B	I/O	V_{CC} = 5V, T_{amb} = +25°C, V_{IO} = 2.0V			16		pF

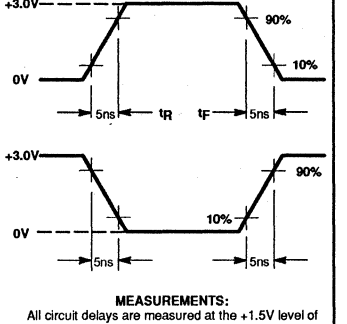
NOTES:
1. All typical values are at V_{CC} = 5V, T_{amb} = +25°C.
2. CMOS inputs: V_{IL} = GND, V_{IH} = V_{CC}.
3. TTL inputs: V_{IL} = 0.45V, V_{IH} = 2.4V.
4. All voltage values are with respect to network ground terminal.
5. Duration of short-circuit should not exceed one second. Test one at a time.
6. ΔI_{CC} vs. Frequency = 4mA/MHz max.

TEST LOAD CIRCUITS



NOTES:
C_1 and C_2 are to bypass V_{CC} to GND.
Test Load R_1 = 750Ω, R_2 = 442Ω, C_L = 30pF (C_L = 5pF for Output Disable)
$0°C \le T_{amb} \le +75°C$, $4.75V \le V_{CC} \le 5.25V$

VOLTAGE WAVEFORMS



MEASUREMENTS:
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.

Input Pulses

MACRO CELL AC SPECIFICATIONS

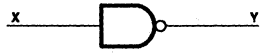
Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V (SNAP Resource Summary Designations in Parentheses)

Input Buffer
(DIN552, NIN552, BDIN552, BNIN552 CDIN552, CNIN552, CKDIN552, CKNIN552, IDFF552*)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	X	I	5	7	10	7	10	15	ns
t_{PLH}	X	I	5	7	10	7	10	15	ns
t_{PHL}	Y	I	5	7	10	7	10	15	ns
t_{PLH}	Y	I	5	7	10	7	10	15	ns

* When input D flip-flop is bypassed.
Input Pins: 12–18, 20, 21, 24, 26–28.
I/O and Bidirectional Pins: 1–3, 5–7, 58–60, 62–66, 69, 70, 75–80, 83, 84.
Bypassed DFF at Pins: 30–32, 34–39, 41–43, 45, 47–49.

Internal NAND of Main Array
(FBNAND, NAND)



SYMBOL	PARAMETER		LIMITS			LIMITS			UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Y	X	10	15	20	12	18	25	ns
t_{PLH}	Y	X	10	15	20	12	18	25	ns

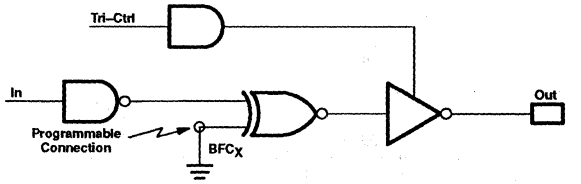
Internal NAND of Clock Array
(NAND)

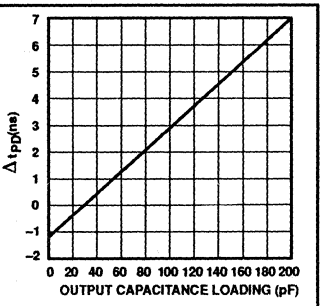
SYMBOL	PARAMETER		LIMITS			LIMITS			UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Y	X	5	7	10	7	10	15	ns
t_{PLH}	Y	X	5	7	10	7	10	15	ns

MACRO CELL AC SPECIFICATIONS (Continued)

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V (SNAP Resource Summary Designations in Parentheses)

3-State Output with Programmable Polarity (TOUT552 + EXOR552 + NAND)



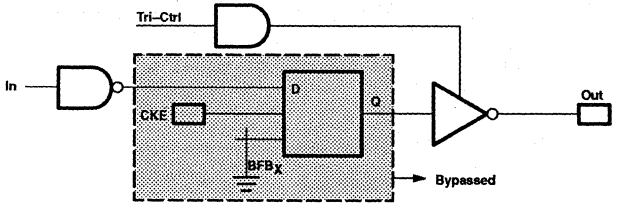


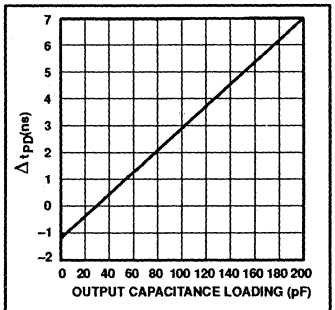
Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Out	In	12	18	25	17	25	35	ns
t_{PLH}	Out	In	12	18	25	17	25	35	ns
t_{OE}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns
t_{OD}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns

Bidirectional Pins: 58–60, 62–66.

I/O Output Buffer with 3-State Control, DFF Bypassed (TOUT552 + NAND)





Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Out	In	12	18	25	17	25	35	ns
t_{PLH}	Out	In	12	18	25	17	25	35	ns
t_{OE}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns
t_{OD}[4]	Out	Tri-Ctrl	5	7	10	7	10	15	ns

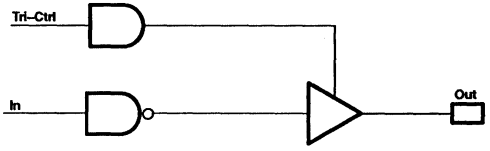
I/O Pins: 1–3, 5–7, 69, 70, 75–80, 83, 84.

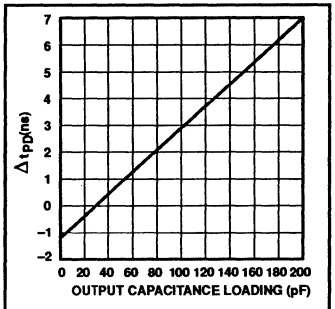
Notes on page 501.

MACRO CELL AC SPECIFICATIONS (Continued)

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V (SNAP Resource Summary Designations in Parentheses)

Active-Low 3-State Output (TOUT852 + NAND)





Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	To (Output)	From (Input)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PHL}	Out	In	12	18	25	17	25	35	ns
t_{PLH}	Out	In	12	18	25	17	25	35	ns

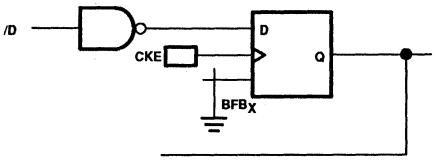
Output Pins: 8–11, 50–57, 71–74.

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

D FLIP-FLOP

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V

Output DFF Used Internally (ODFF552)



SYMBOL	PARAMETER	LIMITS						UNIT
		PML2852-35			PML2852-50			
		MIN	TYP	MAX	MIN	TYP	MAX	
f_{CKE}	Flip-flop toggle rate			50			35	MHz
$t_{W\,CKE}$ High	Clock HIGH	10			14			ns
$t_{W\,CKE}$ Low	Clock LOW	10			14			ns
t_{SETUP} /D	/D setup time to CKE	15			20			ns
t_{HOLD} /D	/D hold time to CKE	4			6			ns

SYMBOL	PARAMETER		LIMITS						UNIT
	From (Input)	To (Output)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CKE ↑	Q	10	15	20	14	20	25	ns
t_{PHL}	CKE ↑	Q	10	15	20	14	20	25	ns

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

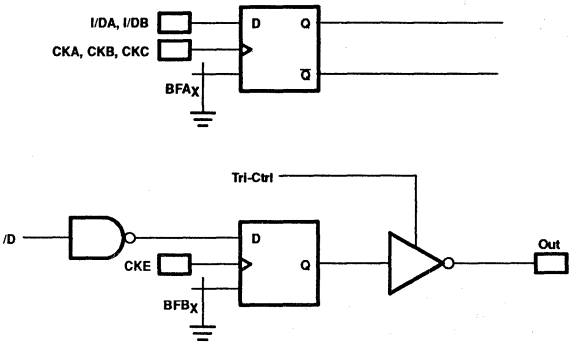
D FLIP-FLOP (Continued)

Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V

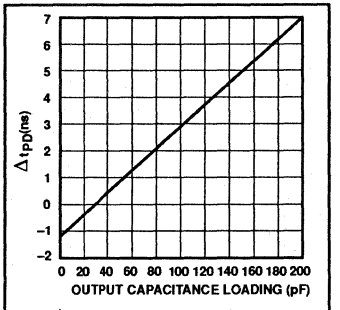
Input and Output (IDFF552 & ODFF552)

INPUTS		OUTPUTS	
CK	D	Q	$\overline{Q}$
L	X	Q_0	$\overline{Q}_0$
↑	H	H	L
↑	L	L	H

NOTE:
Q_0, $\overline{Q}_0$ represent previous stable condition of Q, $\overline{Q}$.



SYMBOL	LIMITS						UNIT
	PML2852-35			PML2852-50			
	MIN	TYP	MAX	MIN	TYP	MAX	
$f_{CKA, CKB, CKC}$			50			35	MHz
$t_{W\ CKA, CKB, CKC}$ High	10			14			ns
$t_{W\ CKA, CKB, CKC}$ Low	10			14			ns
t_{SETUP} I/DA, I/DB	5			7			ns
t_{HOLD} I/DA, I/DB	5			7			ns
f_{CKE}			50			35	MHz
$t_{W\ CKE}$ High	10			14			ns
$t_{W\ CKE}$ Low	10			14			ns
t_{SETUP} /D	15			20			ns
t_{HOLD} /D	4			6			ns



Δt_{PD} vs Output Capacitance Loading (Typical)

SYMBOL	PARAMETER		LIMITS						UNIT
	From (Input)	To (Output)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	5	7	10	7	10	15	ns
t_{PHL}	CKA, CKB/CKC ↑	Q, $\overline{Q}$	5	7	10	7	10	15	ns
t_{PLH}	CKE ↑	Out	12	18	25	17	25	35	ns
t_{PHL}	CKE ↑	Out	12	18	25	17	25	35	ns

MACRO CELL AC SPECIFICATIONS (Continued) (SNAP Resource Summary Designations in Parentheses)

JK FLIP-FLOPS

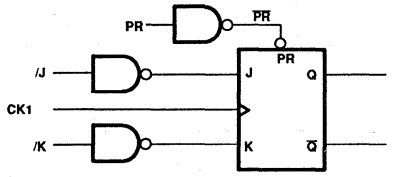
Min: 0°C, 5.25V; Typ: 25°C, 5.0V; Max: 75°C, 4.75V

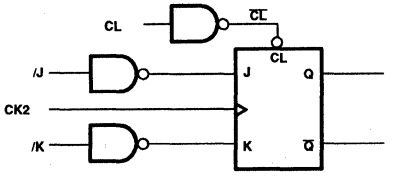
(JKPR552)

INPUTS				OUTPUTS	
$\overline{PR}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	H	L
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$

(JKCL552)

INPUTS				OUTPUTS	
$\overline{CL}$	CK	J	K	Q	$\overline{Q}$
L	X	X	X	L	H
H	↑	L	L	Q_0	$\overline{Q}_0$
H	↑	H	L	H	L
H	↑	L	H	L	H
H	↑	H	H	TOGGLE	
H	L	X	X	Q_0	$\overline{Q}_0$





SYMBOL	PARAMETER	LIMITS						UNIT
		PML2852-35			PML2852-50			
		MIN	TYP	MAX	MIN	TYP	MAX	
f_{CK1}	CK1 toggle frequency			50			35	MHz
f_{CK2}	CK2 toggle frequency			50			35	MHz
$t_{W\,CK1}$ High	CK1 clock HIGH	10			14			ns
$t_{W\,CK1}$ Low	CK1 clock LOW	10			14			ns
$t_{W\,CK2}$ High	CK2 clock HIGH	10			14			ns
$t_{W\,CK2}$ Low	CK2 clock LOW	10			14			ns
t_{SETUP} /J, /K	/J, /K setup time to CK1, CK2	27			35			ns
t_{HOLD} /J, /K	/J, /K hold time to CK1, CK2	0			0			ns
t_W PR Low	Preset Low period	10			14			ns
t_W CL Low	Clear Low period	10			14			ns

SYMBOL	PARAMETER		LIMITS						UNIT
	From (Input)	To (Output)	PML2852-35			PML2852-50			
			MIN	TYP	MAX	MIN	TYP	MAX	
t_{PLH}	CK1,2	Q, $\overline{Q}$	2	3.5	5	3	5	7	ns
t_{PHL}	CK1,2	Q, $\overline{Q}$	2	3.5	5	3	5	7	ns
t_{PLH}	PR	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PHL}	PR	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PLH}	CL	Q, $\overline{Q}$	12	18	25	17	24	30	ns
t_{PHL}	CL	Q, $\overline{Q}$	12	18	25	17	24	30	ns

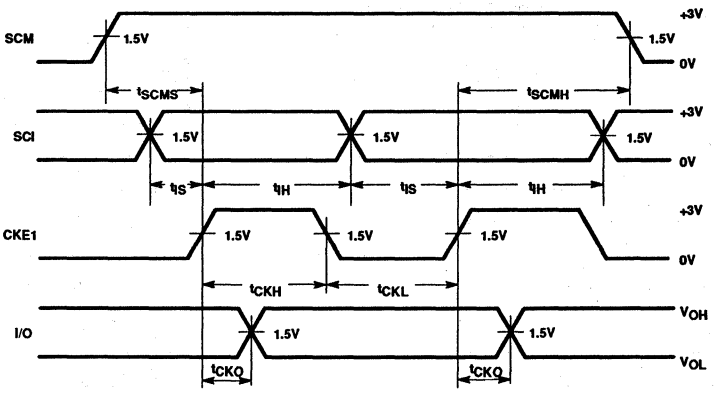
AC ELECTRICAL CHARACTERISTICS

$0°C \leq T_{amb} \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$, $V_{PP} = V_{CC}$,
$R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$ for Output Disable) (See Test Load Circuit Diagram)

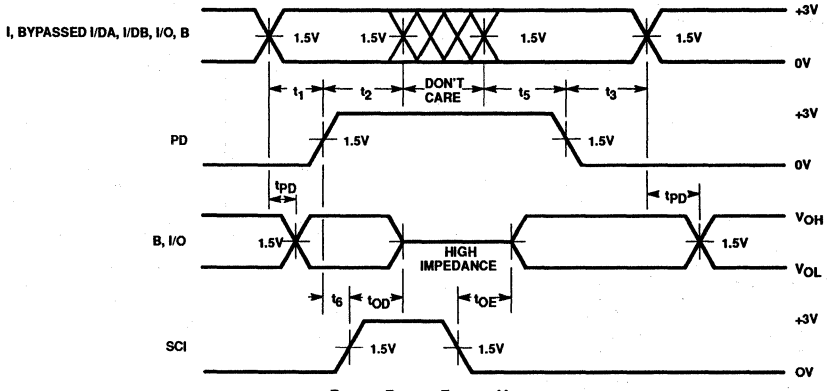
SYMBOL	PARAMETER	LIMITS				UNIT
		PML2852-35		PML2852-50		
		MIN	MAX	MIN	MAX	
Scan mode operation[1]						
t_{SCMS}	Scan Mode (SCM) Setup time	15		15		ns
t_{SCMH}	Scan Mode (SCM) Hold time	25		30		ns
t_{IS}	Data Input (SCI) Setup time	5		5		ns
t_{IH}	Data Input (SCI) Hold time	5		5		ns
t_{CKO}	Clock to Output (I/O) delay		30		40	ns
t_{CKH}	Clock High	10		15		ns
t_{CKL}	Clock Low	10		15		ns
Power down, power up[2]						
t_1	Input (I, bypassed I/DA, I/DB, I/O, B) setup time before power down	40		50		ns
t_2	Input hold time	30		35		ns
t_3	Power Up recovery time		60		70	ns
t_4	Output hold time	0		0		ns
t_5	Input setup time before Power Up	20		25		ns
t_{OE}	SCI to Output Enable time[3]		40		50	ns
t_{OD}	SCI to Output Disable time[3]		40		50	ns
t_6	Power Down setup time	10		15		ns
t_7	Power Up to Output valid		70		80	ns
Power-on reset						
t_{PPR1}	Power-on reset output register (Q = 0) to output (I/O) delay		10		15	ns
t_{PPR2}	Power-on reset input register (Q = 0), buried JK Flip-Flop (Q = 0) to output (B, bypassed I/O) delay		40		50	ns

NOTES:
1. SCM recovery time is 50ns after SCM operation. 50ns after SCM operation, normal operations can be resumed.
2. Timings are measured without foldbacks.
3. Transition is measured at steady state High level (–500mV) or steady state Low level (+500mV) on the output from 1.5V level on the input with specified test load ($R_1 = 750\Omega$, $R_2 = 442\Omega$, $C_L = 5pF$). This parameter is sampled and not 100% tested.
4. For 3-State output; output enable times are tested with $C_L = 30pF$ to the 1.5V level, and S_1 is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with $C_L = 5pF$. High-to-High impedance tests are made to an output voltage of $V_T = (V_{OH} - 0.5V)$ with S_1 open, and Low-to-High impedance tests are made to the $V_T = (V_{OL} + 0.5V)$ level with S_1 closed.

TIMING DIAGRAMS

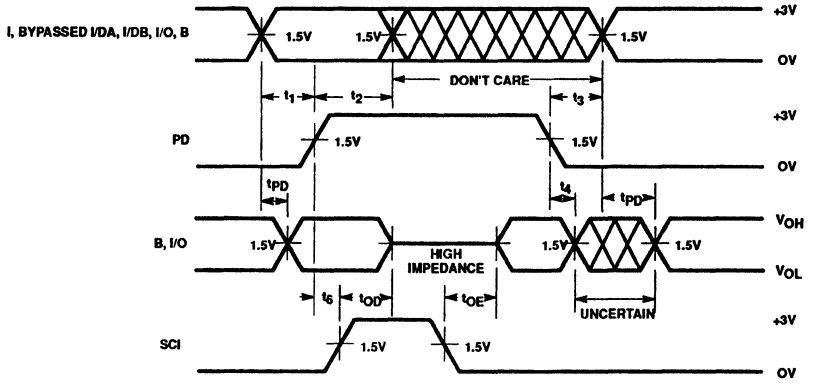


Scan Mode Operation

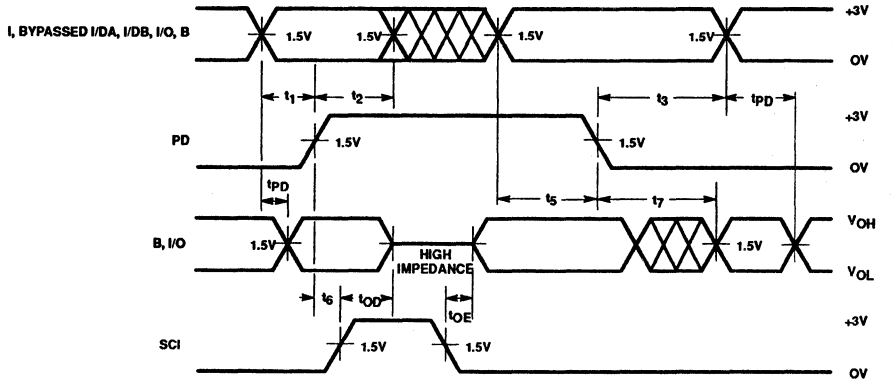


Power Down, Power Up
Input (old) Ready Before Power Up

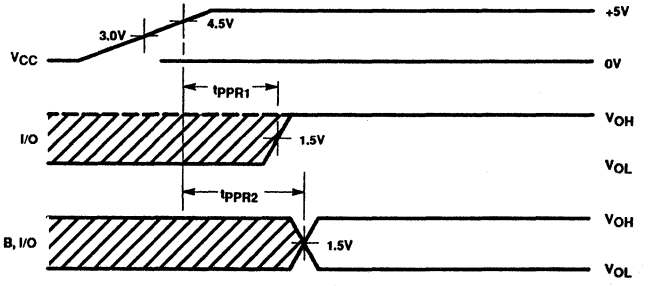
TIMING DIAGRAMS (Continued)



Power Down, Power Up
Input (new) Ready After Power Up

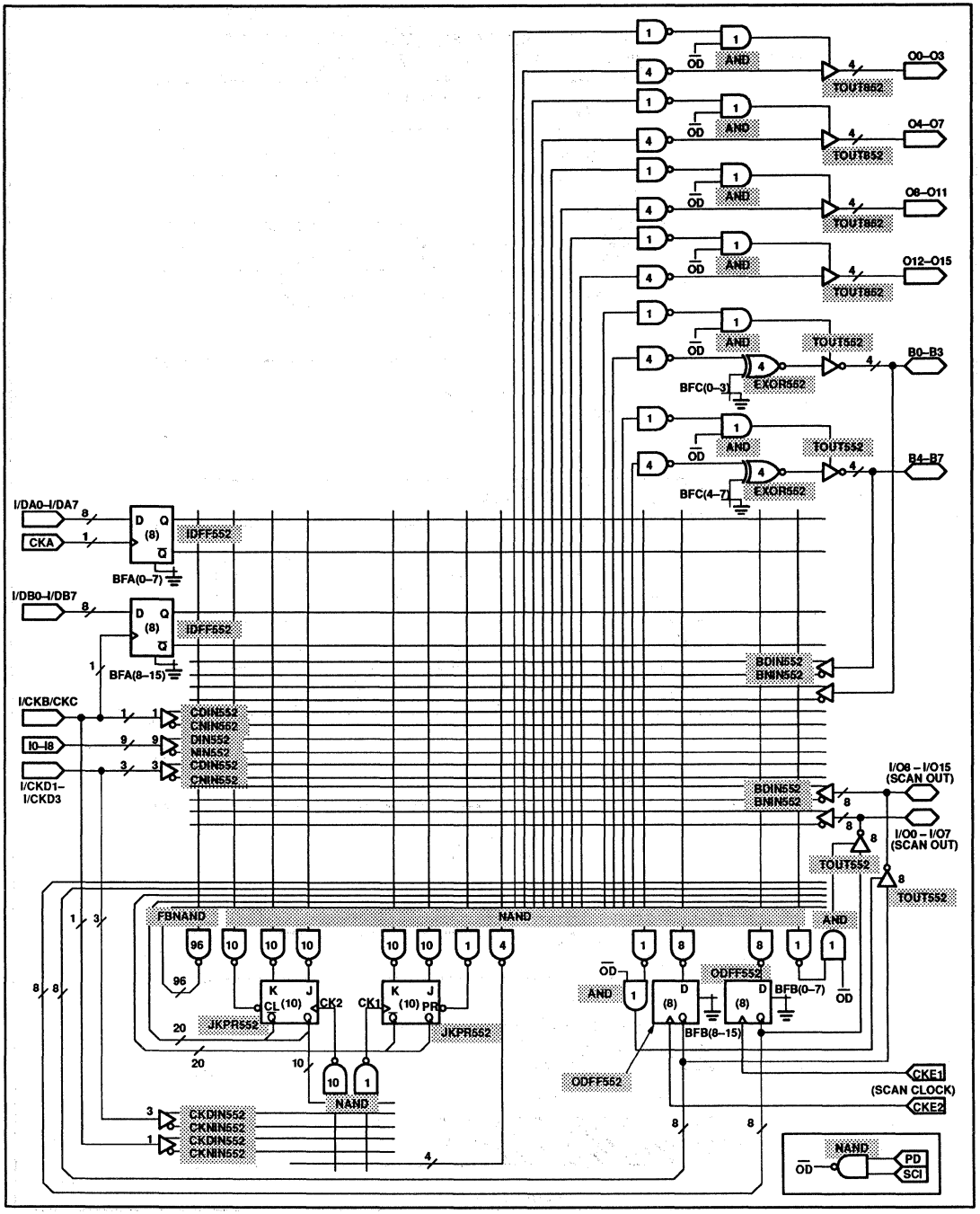


Power Down, Power Up
Input (new) Ready Before Power Up



Power-On Reset

SNAP RESOURCE SUMMARY DESIGNATIONS



ERASURE CHARACTERISTICS (For Quartz Window Packages Only)

The erasure characteristics of the PML2852 device is such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000 – 4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase a typical PML2852 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the PML2852 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the PML2852 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15Wsec/cm^2. The erasure time with this dosage is approximately 30 to 35 minutes using an ultraviolet lamp with a 12,000µW/cm^2 power rating. The device should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose a CMOS EPLD can be exposed to without damage is 7258Wsec/cm^2 (1 week @ 12,000µW/cm^2). Exposure of these CMOS EPLDs to high intensity UV light for longer periods may cause permanent damage.

The maximum number of guaranteed erase/write cycles is 50. Data retention exceeds 20 years.

PROGRAMMING

Refer to the following charts for qualified manufacturers of programmers and software tools:

PROGRAMMER MANUFACTURER	PROGRAMMER MODEL	FAMILY/PINOUT CODES
DATA I/O CORPORATION 10525 WILLOWS ROAD, N.E. P.O. BOX 97046 REDMOND, WASHINGTON 98073–9746 (800)247–5700	UNISITE 40/48 Ver. 3.5 PINSITE Ver. 3.5	15918C* (with adaptor) 15918D
STREBOR DATA COMMUNICATIONS 1008 N. NOB HILL AMERICAN FORK, UT 84003	PLP–S1A Programmer MP68CC adapter	
BASIC COMPUTER SYSTEMS AG WOLFGANG–PAULI–GASSE A–1140 WIEN–AUHOF, AUSTRIA	UP2000 Rev. 2.28	
SMS – W. STEUDEL IM MORGENTAL 13 D–8994 HERGATZ, GERMANY	SPRINT PLUS/EXPERT Rev. TBD	
SYSTEM GENERAL 244 SOUTH PARK VICTORIA DRIVE MILPITAS, CALIFORNIA 95035	TURPRO-1 Rev. 1.42	

* Needs a 40-pin DIP to 84-pin PLCC adaptor that is available from Emulation Technology.
Part Number: AS–84–40–01P–6YAM

EMULATION TECHNOLOGY, INC.
2368B Walsh Avenue, Building D
Santa Clara, California 95051
Telephone No. (408) 982–0660
Fax. No. (408) 982–0664

SOFTWARE MANUFACTURER	DEVELOPMENT SYSTEM
PHILIPS SEMICONDUCTORS 811 EAST ARQUES AVENUE P.O. BOX 3409 SUNNYVALE, CALIFORNIA 94088–3409 (408)991–2000	SNAP SOFTWARE

Section 8
Military Selection Guide

CONTENTS

Military selection guide

Part Number	Device Description	Package Description	Standard MIL-Drawing
PLC18V8Z/BRA	GAL	20DIP3	PLANNED
PLC18V8Z/B2A	GAL	20LLCC	PLANNED
PLC22V10-15/BLA	GAL	24DIP3	5962-8984105MLA
PLC22V10-20/BLA##	GAL	24DIP3	5962-8984102MLA
PLC22V10-25/BLA##	GAL	24DIP3	5962-8984104MLA
PLC22V10-30/BLA##	GAL	24DIP3	5962-8984101MLA
PLS167/BLA	PLS	24DIP3	
PLS168/BLA	PLS	24DIP30	5962-9201201MLA
PLS173	PLA	24DIP3	5962-8850402MLA
82S100/BXA	PLA	28DIP6	M38510/50202XA*
82S100/BYA	PLA	28FLAT	
82S101/BXA	PLA	28DIP6	M38510/50201XA*
82S101/BYA	PLA	28FLAT	
82S101/B3A	PLA	28LLCC	
82S105/BXA	PLS	28DIP6	5962-8670901XA
82S105/BYA	PLS	28FLAT	5962-8670901YA
82S105/B3A	PLS	28LLCC	5962-86709013A
82S153A/BRA	PLA	20DIP3	5962-8768201RA
82S153A/B2A	PLA	20LLCC	5962-87682012A

* Available in QPL part IV specifications
Available as an SMD part number only

Section 9
Development Software

CONTENTS

Synthesis Netlist Analysis Program — SNAP 1.9

FEATURES

- Schematic entry available using Data I/O DASH™ OrCAD SDT IV™
- State equation entry
- Boolean equation entry
- Truth table entry
- Netlist entry
- Edif 2.xx entry
- Graphical simulation waveform entry
- Capability to design in one or any combination of formats
- Device independent, netlist based design platform
- Boolean equation extractor
- Fuse table editor
- Philips LESIM 5-State gate array simulator:
  - Logic and fault simulation
  - Model extraction and timing simulation
  - Synthetic logic analyzer format
  - Stimuli entry in waveform format
  - Simulate multiple-PLD design
- Freezing of selected Critical paths
- Capability to create user defined macros or to use TTL elements
- Automatic test vector generation for combinatorial circuits
- JEDEC fusemap compiler and device programmer interface

GENERAL DESCRIPTION

SNAP PLD development software.
Simple-to-use tools for demanding designs.

Get ready for greater design productivity. SNAP, the complete logic synthesis, simulation and layout package for Philips Semiconductors full line of PLDs, saves one commodity in preciously short supply: design time. **Fully equipped with every tool you need to turn out PLD designs quickly,** SNAP eliminates the "learning curve" that can keep you from being immediately productive. Regardless of whether you're a PLD novice or seasoned pro, SNAP allows you to produce optimized designs within a matter of hours.

For rapid design you need flexibility and SNAP provides lots of it. Enter your design in the most convenient way possible — using **any combination of schematics, truth tables, Boolean equations, state equations or netlists.** SNAP merges the inputs and generates a dense, high-speed design that can be simulated in SNAP's powerful simulator and then downloaded to a PLD programmer.

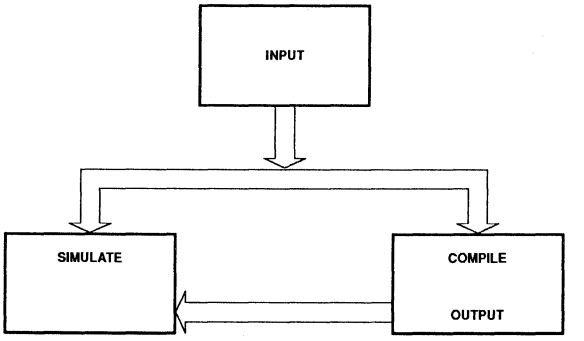
With SNAP, you produce your design in a netlist-based, device-independent environment. No need to commit to a particular part from the start of the design process: With SNAP, you can change the target PLD at will. If you find that your design needs a larger device or can fit into a smaller, less expensive one, simply select a new part and resimulate. SNAP allows you to take advantage of the most appropriate PLD for the job without wasting time.

SNAP'S UNRIVALED SIMULATION FACILITY

Simulation is a key part of the SNAP design process. **SNAP incorporates Philips 5-State ASIC simulator,** a simulator so unsurpassed in its accuracy and diagnostic ability that it is a standard tool used by the company's own chip designers. You can examine any of your design's internal nodes and apply SNAP's virtual logic analyzer to display the precise timing at that node. Then change the stimulus and put the design through its paces with SNAP's built-in waveform editor. Compile into a specific PLD and resimulate. When you finally program a PLD, chances are that it will run perfectly the first time.

Since testability represents an ever-important measure of the success of a PLD design, **SNAP includes a powerful fault simulator** that simplifies the task of analyzing fault coverage. The tool rapidly generates a report detailing undetected and potentially undetectable faults, coverage efficiency, and other useful data. With it, you get the most thorough fault coverage possible in a limited test period.

SNAP Design Flow



DASH is a trademark of Data I/O Corp.
OrCAD STD IV is a trademark of OrCAD, Inc.

Synthesis Netlist Analysis Program SNAP 1.9

Would you like to know how many potential faults your test vectors can detect?
Just look at the output of the SNAP FAULT SIMULATOR...

```
FAULT LIST:
   TOTAL NUMBER OF SIGNALS                =     6
   NUMBER OF NAMED SIGNALS                =     6
   NUMBER OF CIRCUIT FAULTS               =    12
   NUMBER OF INSERTED FAULTS              =    10
   NUMBER OF COLLAPSED FAULTS             =     2

FAULT DETECTION:
   NUMBER OF HARD DETECTED FAULTS         =    12
   NUMBER OF POTENTIALLY DETECTED FAULTS  =     0
   NUMBER OF UNDETECTED FAULTS            =     0

FAULT COVERAGE:
   HARD DETECTION FAULT COVERAGE          =  100.0%
   POTENTIAL DETECTION FAULT COVERAGE     =  0.0  %
   TOTAL DETECTION FAULT COVERAGE         =  100.0%

HARD DETECTION FAULT COVERAGE VERSUS PATTERN# :
PATTERN#   %      0         20        40        60        80        100
-----------------+---------+---------+---------+---------+---------+
       1  58.3    .*****************************
       2  75.0    .*************************************
       3  83.3    .*******************************************
       4  100.0   .**************************************************
       5  100.0   .**************************************************
-----------------+---------+---------+---------+---------+---------+
```

Designers who need to consolidate the designs of existing logic devices will draw considerable benefit from SNAP's unique **Boolean equation extractor.** It take the design data from existing PLDs and converts it to the actual, corresponding Boolean equations, which can then be used as an input to SNAP. It eliminates the need to find and re-enter design data, often a time-consuming process.

And for added convenience, **SNAP features the powerful logic optimizer, Espresso Minimizer.** Espresso automatically removes all unnecessary gates from your design, assuring that it will be the fastest and densest possible. Espresso allows you to pack more in — or fit it into a smaller PLD. The result can be substantial cost and power savings.

FULL SUPPORT NOW — AND INTO THE FUTURE

SNAP supports Philips broad line of PLDs, which includes high-speed PAL®-type devices, programmable logic arrays, programmable logic sequencers, and sophisticated programmable macro logic. It is fully compatible with SLICE, Philips entry-level design package. And as Philips introduces new PLDs in the future, SNAP will support those too, in a timely manner. You can standardize on SNAP for your future development, with confidence.

Menu-driven and supported by clear, concise documentation, SNAP is a pleasure to use. But if problems do arise, Philips network of field applications engineers stand ready to help. Specially trained and backed by a comps of factory experts, Philips FAEs are stationed in all major cities in the U.S. and overseas. Wherever you are, chances are that support is nearby.

PAL is a registered trademark of AMD/MMI, Inc.

Synthesis Netlist Analysis Program SNAP 1.9

SNAP Boolean Equation Extractor

FROM THIS:

```
L0000
11111111111111011101111011111111111111
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
11111111111110111111111111111111011111
```

TO THIS:

```
@LOGIC EQUATION
S.D     =/(ASN*RWN*/DREQN1+S*/DREQN1);
```

AUTOMATICALLY • IMMEDIATELY • EFFORTLESSLY

PRODUCT SUPPORT

SLICE supports the Philips line of PLDs, which ranges from high-speed PAL devices to complex Programmable Macro Logic circuits. It will also support new Philips PLDs as they are introduced. The devices currently supported are:

Programmable Logic Arrays:

PLUS153 PLS100
PLUS173

Programmable Macro Logic:

PLHS501 PML2852
PML2552

Programmable Logic Sequencers:

PLS155 PLC42VA12
PLS157 PLC415
PLS159 PLS105
PLS167 PLUS105
PLS168 PLUS405
PLS179

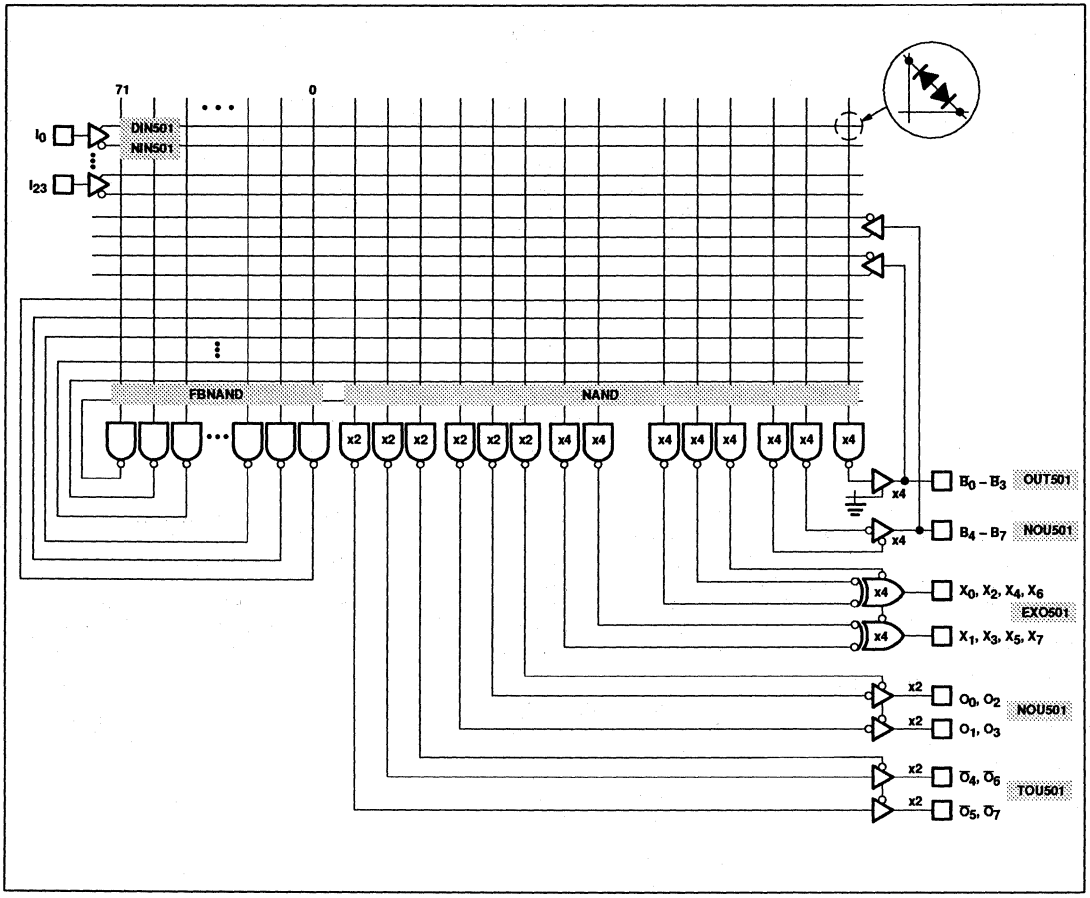
Programmable Array Logic

PLUS20L8 PLUS16L8
PLUS20R8 PLUS16R8
PLUS20R6 PLUS16R6
PLUS20R4 PLUS16R4
10H20EV8 PLQ22V10
10020EV8 PLC18V8Z
PHD48N22 PL22V10
PHD16N8

TRY IT — YOU'LL LIKE IT

Pop the enclosed SNAP demo disk into your computer and see how easy PLD design can be. The demo, like SNAP itself, runs on almost any IBM® PC or compatible having DOS 3.0 or higher, 640K RAM and a hard disk.

IBM is a registered trademark of International Business Machines Corp.




```
SNAP Resources Summary

   Cell name   used/total     %
=============================================
      DIN501    18 /  32    56%
      NIN501     7 /  32    21%
      FBNAND    72 /  72   100%
        NAND    34 /  44    77%
      OUT501     2 /   4    50%
      NOU501     4 /   8    50%
      EXO501     8 /   8   100%
      TOU501     4 /   4   100%
Please hit any key to continue...
```

PLHS501 Resources

Synthesis Netlist Analysis Program SNAP 1.9

SNAP OVERVIEW

Philips SNAP (Synthesis Netlist Analysis and Program) is a software program used in implementing logic designs with Philips Programmable Logic Devices. The software runs on any IBM PS/2, AT, XT, or compatible computer. SNAP accepts the logic design specified in the form of schematics, EDIF netlists, Boolean logic equations, and/or state equations; combines the different forms and different parts of the design into a single netlist; prompts the user to select a target PLD; and generates the JEDEC fuse map used for programming the target PLD device.

Schematics can be created with either OrCAD SDT III, OrCAD SDT IV or DASH, three schematic capture packages offered as options to SNAP. Logic and state equations can be created using any ASCII text editor. After you specify the design, SNAP converts the schematic, logic equations, and state equations into a single netlist. You can then use SNAP to perform the following functions:

- Create, display, and edit the stimulus waveforms for simulation
- Simulate the logic functions and timing
- Display and print the simulation results
- Determine the fault coverage for a given set of inputs
- Generate the test vectors
- Generate the fuse map for the target PLD device
- Generate a netlist of the PLD implementation for simulation
- Download the fuse map and test vectors to the PLD programmer

Specification of the logic design is independent of the type of PLD device. You can specify the design first and choose the PLD device later, after simulating and debugging the logic design. If the chosen device is unable to accommodate the design, it is a simple matter to select another device and generate the fuse map for that device. After this has been done, SNAP can generate a netlist and a set of logic equations directly from the final fuse map, allowing analysis and simulation of the final design as implemented in the target device. Also, a design using several PLD devices can be accurately simulated with the use of real delays.

Supported PLD Devices

The PLD devices supported by SNAP 1.8 are listed below, showing the part number, architecture (Inputs x Terms x Outputs), and number of pins for each device type.

Programmable Macro Logic (PML) Devices

PLHS501	104 x 116 x 24	52 pins
PML2552	185 x 226 x 24	68 pins
PML2852	185 x 226 x 40	84 pins

Programmable Logic Sequencer (PLS) Devices

PLS155	16 x 45 x 12	20 pins
PLS157	16 x 45 x 12	20 pins
PLS159	16 x 45 x 12	20 pins
PLS167	14 x 48 x 6	24 pins
PLS168	16 x 45 x 12	20 pins
PLS179	12 x 48 x 8	24 pins
PLC42VA12	42 x 105 x 12	24 pins
PLC415	17 x 68 x 8	28 pins
PLS105	16 x 48 x 8	28 pins
PLUS105	22 x 48 x 8	28 pins
PLUS405	24 x 64 x 8	28 pins

Programmable Logic Array (PLA) Devices

PLS100	16 x 48 x 8	28 pins
PLUS153	18 x 42 x 10	20 pins
PLUS173	22 x 42 x 10	24 pins

PAL Devices

PLUS16L8	16 x 64 x 8	20 pins
PLUS16R8	16 x 64 x 8	20 pins
PLUS16R6	16 x 64 x 8	20 pins
PLUS16R4	16 x 64 x 8	20 pins
PHD16N8	16 x 16 x 8	20 pins
PLC18V8Z	18 x 74 x 8	20 pins
PLUS20L8	20 x 64 x 8	24 pins
PLUS20R8	20 x 64 x 8	24 pins
PLUS20R6	20 x 64 x 8	24 pins
PLUS20R4	20 x 64 x 8	24 pins
10X20EV8	20 x 90 x 8	24 pins
PHD48N22	48 x 73 x 22	68 pins
PL22V10	22 x 132 x 10	24 pins
PLQ22V10	22 x 132 x 10	24 pins

Before you can begin using SNAP, you must first install the software and learn the function keys and top-level menu. As part of the setup procedure, you specify the text editor and schematic capture software you are using with SNAP so that SNAP can invoke these programs as needed.

Overview of SNAP Process

The OrCAD SDT IV and DASH schematic capture systems are available as options to the SNAP software package.

OrCAD SDT IV is a complete schematic capture package, one of several design tools offered by OrCad Systems Corporation. OrCAD SDT IV lets you create, edit, save, and print logic schematics. Schematic data files are accepted directly by the SNAP software. Instructions on installing and using OrCAD SDT IV are provided with the OrCAD SDT IV software package. Supplemental information is provided in Appendix A of the User's Manual on configuring OrCAD SDT IV for compatibilty with SNAP.

DASH is Data I/O schematic capture package. Schematic data files are accepted directly by the SNAP software. Instructions on installing and using DASH are provided in Appendix B of the User's Manual, serving as an addendum to the DASH User's Manual.

Any of these schematic capture systems may be used for logic design purposes with SNAP.

SNAP is an interactive, menu-driven software package. At the top level of the program is a graphical menu that allows selection of the desired SNAP operation. See Figure 5.

The boxes show the SNAP program operations that you can select. Operations may be performed at any time and in any order, provided that the input files for that operation are available.

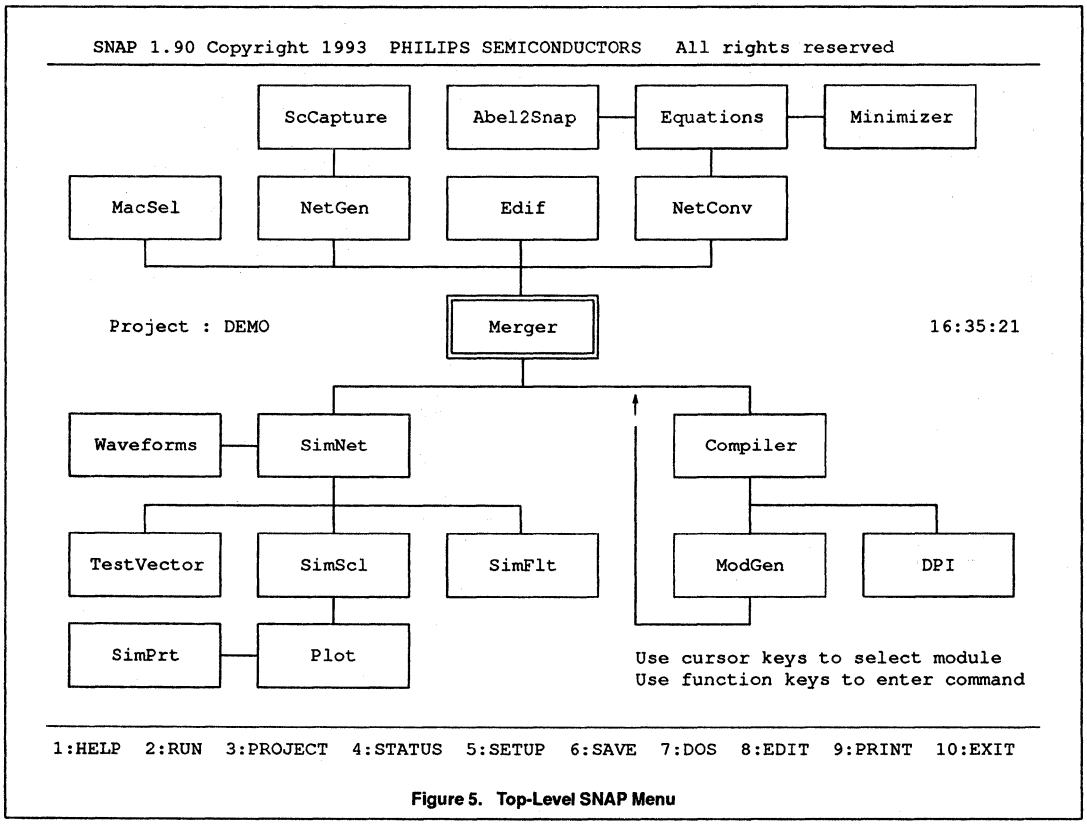


Figure 5. Top-Level SNAP Menu

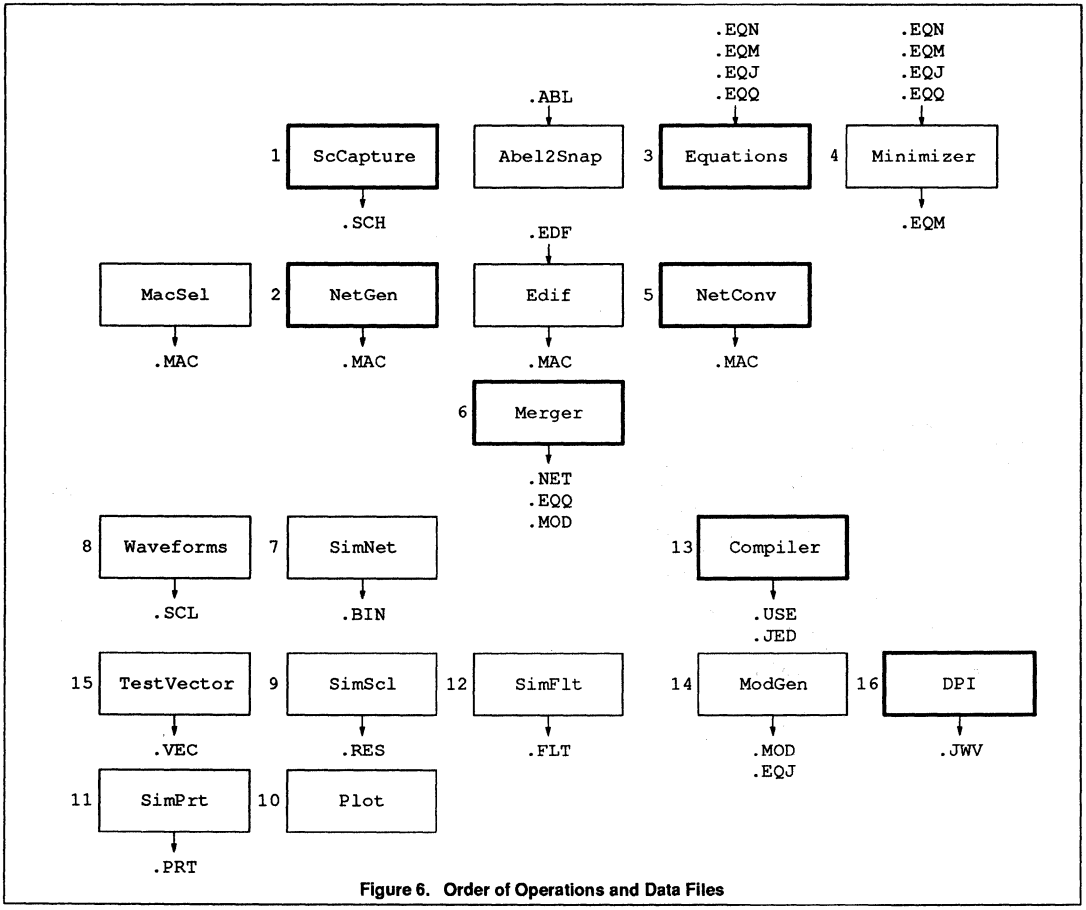


Figure 6. Order of Operations and Data Files

Figure 6 shows the typical order in which the SNAP program operations are executed. The figure also shows the file name extensions for the files produced by (and used by) the individual program operations. The menu options shown in bold boxes are the minimum required to specify a design and generate the fuse map. The remaining menu options may be used as needed for simulation and testing purposes.

In Step 1 (ScCapture), you specify part or all of the logic design with the schematic capture package (OrCAD SDT IV or DASH), using a library of logic elements recognized by SNAP. The design may be drawn hierarchically. In Step 2 (NetGen), SNAP converts the schematic into an intermediate netlist (.MAC file).

In Step 3 (Equations), you specify part or all of the logic design with Boolean logic equations and/or state equations. In Step 4 (Minimizer), an optional step, SNAP changes the form of the equations to minimize the number of gates necessary to implement the design. In Step 5 (NetConv), SNAP converts the logic and state equations into an intermediate netlist (.MAC file).

The complete design may be specified with any combination of schematic capture, logic equations, and state equations. Different parts of the design may be specified separately. Each lower-level part of the design is a "macro" that can be used multiple times at a higher level of the hierarchy.

In Step 6 (Merger), SNAP combines the separate netlists (.MAC files) into a single master netlist (.NET file).

In Step 7 (SimNet), SNAP converts the master netlist into a binary-format file (.BIN file) that is accepted by the simulator.

In Step 8 (Waveforms), you use a graphical waveform editor to create the input signals for the simulation. SNAP converts the waveforms into the "Simulation Control Language" format (.SCL file).

In Step 9 (SimScl), SNAP simulates the logic operation and timing of the design using the input signals created previously. The resulting output signals are stored in a "results" file (.RES file).

In Step 10 (Plot), SNAP displays the results graphically on the screen. You can analyze the simulation results in detail by adjusting the time range and time scale of the display.

In Step 11 (SimPrt), SNAP prints out the simulation results on the printer, monitor screen, or a disk file. You select the type of display (alphanumeric or graphic), the time range, and the time scale for the printout.

In Step 12 (SimFlt), SNAP simulates the design with circuit faults, and reports the percentage of potential faults that can be detected with the given set of input test signals. Test signals may be specified as waveforms or by an ASCII file. A detailed fault coverage report is generated (.FLT file).

In Step 13 (Compiler), SNAP generates the fuse map for implementing the logic design. You select the PLD device type and then specify the input/output signal name associated with each device pin. SNAP optimizes the design for the selected device, generates the fuse map, and writes out the results in JEDEC format (.JED file). The percentage utilization of the on-chip PLD resources is reported on the screen and stored in a separate file (.USE).

In Step 14 (ModGen), SNAP takes the PLD device structure and fuse map, and generates a new netlist (.MOD file) based on the actual implementation of the logic design in the PLD device. This new netlist can be simulated in the same manner as the original design, allowing verification of the PLD implementation.

In Step 15 (TestVector), the test vectors (input signals and expected output signals) are converted to JEDEC format. This information can be downloaded to the device programmer machine along with the fuse map (.VEC file).

In Step 16 (DPI), the Device Programmer Interface, SNAP downloads the fuse map and test vectors to the PLD programmer machine through a serial port. The programmer machine uses the fuse map to program the PLD device, and the test vectors to test the device after programming.

The programmed device operates as specified by the schematics, logic equations, and state equations created in Steps 1 and 3.

Many of the steps described above are optional. The minimum steps necessary for a project are either ScCapture and NetGen, or Equations and NetConv, to specify the logic design; Merger to make the netlist; Compiler to generate the fuse map; and DPI to download the fuse map to the device programmer. The other steps allow you to analyze and simulate the design, and to generate the test vectors.

Hardware and Software Requirements

SNAP requires the following computer resources:

- IBM PS/2, AT, XT or compatible computer
- MS-DOS operating system, version 3.0 or higher
- 640 Kbytes RAM
- Hard disk drive: 10 Mbytes (20 Mbytes or more preferred)
- Floppy disk drive
- Monitor: Hercules, EGA, or VGA recommended for schematic capture
- Text editor software

Installation

The SNAP software is provided on a set of floppy diskettes. Two functionally identical sets are provided: a $5^1/_4$ inch set and a $3^1/_2$ inch set.

The files are stored on the diskettes in compressed-data format, so you cannot simply copy the files to the hard disk. Instead, use the installation program provided on the diskettes. Running the installation program is straightforward. The program takes care of creating a SNAP subdirectory (if it doesn't already exist), and automatically "decompresses" the SNAP files and transfers them to the hard disk.If you have an earlier version of SNAP installed on your system, first make a backup of all data files (if any) in your SNAP subdirectory. To ensure that you don't lose any valuable files, make a separate, complete backup of the SNAP subdirectory using the BACKUP command or a backup utility program. Then delete all the files from the SNAP subdirectory.

If you are upgrading from SLICE, you can install SNAP without removing SLICE. Once you are familiar with SNAP, you can delete SLICE from the hard disk.

Section 10
Programmer/Software Support

CONTENTS

Philips Semiconductors PLD programming guide

Philips Semiconductors	ADVANTEST	ADVIN	PIE	BP MICROSYSTEMS		DATA I/O*		
	R4971	PILOT-U84	UP2000	BP-1200	CP-1128	UNISITE	MODEL 2900	MODEL 3900
PRODUCT NAME	Revision	Revision	Revision	Software Revision	Software Revision	Revision	System Revision	System Revision
PAL® DEVICES								
10H20EV8-4	–	–	2.25	2.22	1.78	4.2	1.5	–
10020EV8-4	–	–	2.25	2.22	1.78	4.2	1.5	–
PHD16N8-5	–	10.16	2.25	2.22	1.78	2.8	1.5	1.0
PHD48N22-7	–	–	2.25	2.22	–	3.4	–	1.1
PL22V10-10/-12/-15	–	10.32	3.00	2.22	1.81	3.5 ***	1.5	1.1
PLC18V8Z-25/-35	–	10.16	2.25	2.22	1.78	2.6 ***	1.5	1.0
PLUS16L8-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.2
PLUS16R4-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.2
PLUS16R6-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.2
PLUS16R8-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.2
PLUS20L8-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.0
PLUS20R4-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.0
PLUS20R6-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.0
PLUS20R8-7/D **	C50	10.16	2.25	2.22	1.78	4.1	1.5	1.0
PLA DEVICES								
PLS100	C50	10.35	–	2.22	1.78	2.2	1.5	1.0
PLS101	C50	10.35	–	2.22	1.78	2.2	1.5	1.0
PLS153/153A	C50	10.35	2.25	2.22	1.78	2.8	1.5	1.0
PLS173	C50	10.35	2.25	2.22	1.78	1.7	1.5	1.0
PLUS153-10/D/B	C50	10.35	2.25	2.22	1.78	3.6	1.5	1.1
PLUS173-10/D/B	C50	10.35	2.25	2.22	1.78	2.3	1.5	1.1
PLS DEVICES								
PLC415-16	–	–	2.25	2.21C	1.78	2.6	1.5	1.0
PLC42VA12	–	10.32	2.25	2.21C	1.78	3.5	1.6	1.0
PLS105/105A	C50	10.35	2.25	2.21C	1.78	1.5	1.0	1.0
PLS155	C50	10.35	2.25	2.21C	1.78	1.5	4.3	1.0
PLS157	C50	10.35	2.25	2.21C	1.78	1.5	4.3	1.0
PLS159A	C50	10.35	2.25	2.21C	1.78	3.0	1.5	1.0
PLS167/167A	C50	10.35	2.25	2.21C	1.78	1.5	1.5	1.0
PLS168/168A	C50	10.35	2.25	2.21C	1.78	1.5	1.5	1.0
PLS179	C50	10.35	2.25	2.21C	1.78	3.3	1.5	1.0
PLUS105-55/-45	C50	–	2.25	2.21C	1.78	4.2	1.6	1.0
PLUS405-55/-45/-37	C50	10.35	2.25	2.21C	1.78	4.2	1.6	1.0
PML DEVICES								
PLHS501/501I	–	–	2.25	–	–	1.7	1.7	1.1
PML2552-35/-50	–	–	2.25	–	–	2.8	1.7	1.1
PML2852-35/-50	–	–	3.0	–	–	3.8	–	2.0

* See individual programmer reference guide for more details.

** New revision listed is required if you wish to program the security fuse on the following products:
PLUS20L8/R8/R6/R4 Rev. G or later
PLUS16R8/R6/R4 Rev. I or later
PLUS16L8 Rev. K or later

*** Need version 4.0 for SO Package Support

®PAL is a registered trademark of Advanced Micro Devices, Inc.

Philips Semiconductors PLD programming guide

DATA I/O* (continued)		LOGICAL DEVICES		SMS	STAG*	STREBOR	SYSTEM GENERAL	
MODEL 29B	**MODEL 60**	**ALLPRO40**	**ALLPRO88**	**SPRINT PLUS**	**ZL30A**	**PLP-S1A**	**SGUP-85A**	**TURPRO-1**
303A-011A Revision	**System Revision**	**Software Revision**	**Software Revision**	**System Revision**	**System Revision**	**Software Revision**	**System Revision**	**System Revision**
PAL® DEVICES								
–	–	1.50C	2.10	3.5	–	–	2.4	1.68
–	–	1.50C	2.10	3.5	–	–	2.4	1.68
V12	V15	1.50C	2.10	3.5	30A36	–	2.4	1.68
–	–	–	2.10	–	–	FD	–	1.68
V14	V18	–	–	–	30B01	–	–	1.68
V09	V15	1.50C	2.10	3.5	30A31	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
V08	V14.2/19	1.50C	2.10	3.5	30A31/B07	–	2.4	1.68
PLA DEVICES								
V05*	V01	1.50C	2.10	3.5	30A01	–	–	–
V05*	V01	1.50C	2.10	3.5	30A01	–	–	–
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V07	V15	1.50C	2.10	3.5	30A40	–	2.4	1.68
V07	V15	1.50C	2.10	3.5	30A40	–	2.4	1.68
PLS DEVICES								
V10	V17.1	1.50C	2.10	3.5	30A34	–	–	1.68
V12	V15	1.50C	2.10	–	30A34	–	2.4	–
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	V13	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	V12	1.50C	2.10	3.5	30A25	–	2.4	1.68
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	V12	1.50C	2.10	3.5	30A01	–	2.4	1.68
V02	–	1.50C	2.10	3.5	30A27	–	2.4	1.68
V09	–	1.50C	2.10	3.5	30A37	–	–	–
V07	–	1.50C	2.10	3.5	30A31	–	2.4	1.68
PML DEVICES								
–	–	1.50C	2.10	–	30A22	FA	2.4	1.68
–	–	–	–	–	–	FC	–	1.68
–	–	–	–	–	–	FD	–	1.68

PLD programmer reference guide — Data I/O Corporation

Data I/O Corporation
10524 Willows Road, N.E.
Redmond, Washington 98073-9746
Telephone Number: (800) 247-5700

Philips Semiconductors Part Number	Device Code	MODEL 29B Adapter Revision		UNISITE		MODEL 2900	MODEL 3900	MODEL 60 Adapter Revision		
		DIP	PLCC	Site 40/48	Chip/ Pin Site			System Revision	DIP	PLCC
PHD										
PHD16N8	1B8F	303A-011A;V12	303A-011B;V05	V2.8	V3.4	1.6	1.0	V15	360A001	360A006
PHD48N22	0960B2	—	—	V3.4****	V3.4**	—	1.1	—	—	—
ECL										
10H20EV8	14013B	—	—	V2.7	V2.7	1.5	—	—	—	—
10020EV8	14013B	—	—	V3.0	TBD	1.5	—	—	—	—
PAL®										
PLC18V8Z-35/-25	864F	303A-011A;V09	303A-011B;V04	V2.6	V2.8	1.5	1.0	V15	360A001	360A006
PL22V10	A628	303A-011A;V14	303A-011B;V04	V3.5	V3.5	1.5	1.1	V18	360A001	360A006
PLUS16L8	1B17	303A-011A;V08	303A-011B;V04	V3.8*	V3.8*	1.5	1.2	V14.2	360A001	360A006
PLUS16R8/R6/R4	1B24	303A-011A;V08	303A-011B;V04	V3.8*	V3.8*	1.5	1.2	V14.2	360A001	360A006
PLUS20L8	1B26	303A-011A;V08	303A-011B;V04	V3.9*	V3.9*	1.5	1.0	V14.2	360A001	360A006
PLUS20R8/R6/R4	1B27	303A-011A;V08	303A-011B;V04	V3.9*	V3.9*	1.5	1.0	V14.2	360A001	360A006
PLA										
PLS100/101	9601	303A-001;V01	—	—	—	1.0	1.0	—	—	—
PLS100/101	9661	303A-001;V05	—	V2.2	V2.2	1.0	1.0	V01	360A003	—
PLS153/A	9665	303A-011A;V02	303A-011B;V02	V2.8	—	1.0	1.0	V01	360A002	A ONLY
PLS153/A	9665	303A-001;V05	303A-011B;V02	V2.8	V2.8	1.0	1.0	V12	360A009	A ONLY
PLS173	9676	303A-011A;V02	303A-011B;V02	V1.7	—	1.0	1.0	V08	360A002	—
PLS173	9676	303A-001;V06	303A-011B;V02	V1.7	V1.7	1.0	1.0	V12	—	360A009
PLUS153B/D/-10	1B65	303A-011A;V07	303A-011B;V03	V3.6	V3.6	1.0	1.1	V15	360A001	360A009
PLUS173B/D/-10	1B76	303A-011A;V07	303A-011B;V03	V2.3	V2.3	1.0	1.1	V15	360A002	360A009
PLS										
PLC415-16	86AA	303A-011A;V10	303A-011B;V04	V2.6	V2.7	1.5	1.0	V17.1	360A003	TBD
PLC42VA12	868A	303A-011A;V12	303A-011B;V05	V3.5	V3.5	1.6	1.0	V15	360A002	TBD
PLS105/A	9603	303A-011A;V02	—	V1.5	—	1.0	1.0	V01	360A003	A ONLY
PLS105/A	9603	303A-001;V01	—	—	—	1.0	1.0	V12	—	—
PLS105/A	9663	303A-001;V05	303A-011B;V02	V1.5	—	1.0	1.0	—	360A003	—
PLS105/A	9663	303A-011A;V02	303A-011B;V02	V1.5	V1.5	1.0	1.0	V01	—	360A008
PLUS105-45/-55	1B63	303A-011A;V09	303A-011B;V04	V3.6	V3.6	2.3	1.0	—	—	—
PLS155	9667	303A-011A;V02	303A-011B;V02	V1.5	—	1.0	1.0	V01	360A002	—
PLS155	9667	303A-001;V05	303A-011B;V02	V1.5	V1.5	1.0	1.0	V12	—	360A009
PLS157	9668	303A-001;V05	303A-011B;V02	V1.5	—	1.5	1.0	V13	360A002	—
PLS157	9668	303A-011A;V02	303A-011B;V02	V1.5	V1.5	1.5	1.0	V13	—	360A009
PLS159A	6466	303A-011A;V02	303A-011B;V02	V3.0	V2.8	1.5	1.0	V12	360A002	360A009
PLS159A	6466	—	—	V3.0	—	1.5	1.0	—	—	—
PLS167/A	9660	303A-011A;V02	303A-011B;V02	V1.5	V1.5	1.5	1.0	V05	360A002	—
PLS167/A	9660	—	—	—	—	1.5	1.0	V12	—	360A009
PLS168/A	9674	303A-011A;V02	303A-011B;V02	V1.5	V1.5	1.5	1.0	V05	360A002	—
PLS168/A	9674	303A-001;V06	—	—	—	1.5	1.0	V12	—	360A009
PLS179	9677	303A-011A;V02	303A-011B;V02	V3.3	V3.3	1.5	1.0	—	—	—
PLUS405-37/-45/-55	1B79	303A-011A;V07	303A-011B;V04	V3.6	V3.6	2.3	1.7	—	—	—

PLD programmer reference guide — Data I/O Corporation

Philips Semiconductors Part Number	Device Code	MODEL 29B Adapter Revision		UNISITE		MODEL 2900	MODEL 3900	MODEL 60 Adapter Revision		
		DIP	PLCC	Site 40/48	Chip/ Pin Site			System Revision	DIP	PLCC
PML										
PLHS501	1002	—	—	—	V1.7	—	1.1	—	—	—
PLHS502	01C05E	—	—	V2.4***	V3.2**	—	—	—	—	—
PML2552-35/-50	15908C	—	—	V2.8****	V3.1**	—	1.1	—	—	—
PML2852-35/-50	15918C	—	—	V3.5*****	V3.5	—	—	—	—	—

NOTES:

1. The software and hardware revisions listed are the first revisions released. All following revisions maintain support.
2. FOR UNISITE USERS: PLCC packages can be programmed on either the Chipsite or Pinsite adaptors.
3. FOR UNISITE USERS ONLY: Family codes listed above (the first two digits) must be preceeded with a "0" for PLCC packages. Pin codes listed above (the last two digits) must be preceeded with a "7" or "6" for PLCC packages. Also, product name might be preceeded by "-FN".
4. * This version required to program security fuse on newer product.
   Older parts can use Version 2.3 or later.
   ** Pinsite adaptor required to program and functionally test these products without a DIP to PLCC adaptor.
   *** Needs a 40-Pin DIP to 68-Pin PLCC adaptor available from Emulation Technology. Part Number: AS-68-40-O1P-6
   Pinsite is also available for programming and functional testing without an adaptor.
   **** Needs a 40-Pin DIP to 68-Pin PLCC adaptor that is available from Emulation Technology. Part Number: AS-68-40-04P-6
   Pinsite is also available for programming and functional testing without an adaptor.
   ***** Needs a 40-Pin DIP to 84-Pin PLCC adaptor available from Emulation Technology Part Number: AS-84-40-O1P-6YAM
   EMULATION TECHNOLOGY, INC.
   2368B Walsh Avenue, Building D
   Santa Clara, California 95051
   Telephone No. (408) 982-0660
   Fax. No. (408) 982-0664
5. DEVICE CODE: XXYY
   XX = FAMILY CODE
   YY = PIN CODE

®PAL is a registered trademark of Advanced Micro Devices, Inc.

PLD programmer reference guide — Stag Micro Systems, Inc.

STAG MICRO SYSTEMS, INC.

Western Area:
1600 Waytt Drive, Suite 3
Santa Clara, CA 95054
(408) 988-1118

Eastern Area:
3 Northern Blvd., Suite B4
Amherst, NH 03031
(603) 673-4380

PHILIPS SEMICONDUCTORS PART NUMBER	DEVICE CODES		MODEL ZL30 (DIP ONLY)	MODEL ZL30A	
	FAMILY CODES	PIN CODES		SYSTEM REVISION	PLCC ADAPTER
PHD DEVICES					
PHD16N8-5	10	167	30A36	30A36	30A001
ECL DEVICES					
10H/10020EV8	--	--	--	--	--
PAL® DEVICES					
PL22V10-10/-12/-15	12	070	--	30B01	TBA
PLC18V8Z	12	205	30A34	30A34	30A001
PLUS20L8D/-7	11	56	30A31	30A31/B07	30A001
PLUS20R8D/-7	11	57	30A31	30A31/B07	30A001
PLUS20R6D/-7	11	58	30A31	30A31/B07	30A001
PLUS20R4D/-7	11	59	30A31	30A31/B07	30A001
PLUS16L8D/-7	11	29	30A31	30A31/B07	30A001
PLUS16R8D/-7	11	30	30A31	30A31/B07	30A001
PLUS16R6D/-7	11	31	30A31	30A31/B07	30A001
PLUS16R4D/-7	11	32	30A31	30A31/B07	30A001
PLA DEVICES					
PLS100/101	13	00	30A01	30A01	30A001
PLS153/153A	14	05	30A01	30A01	30A001
PLUS153B/D/-10	11	05	30A39S	30A39S	30A001
PLS173	15	96	30A01	30A01	TBA
PLUS173B/D/-10	11	96	30A39S	30A39S	TBA
PLS DEVICES					
PLS105/105A	13	02	30A01	30A01	30A001
PLUS105-45/-55	11	02	30A39	30A37	30A001
PLC415	12	177	30A34	30A34	30A001
PLC42VA12	12	197	30A45	30A45	30A001
PLS155	14	06	30A01	30A01	30A001
PLS157	14	07	30A01	30A01	30A001
PLS159A	13	08	30A25	30A25	30A001
PLS167/167A	15	91	30A01	30A01	30A001
PLS168/168A	15	97	30A01	30A01	30A001
PLS179	15	130	30A27	30A27	30A001
PLUS405-37/-45/-55	11	138	30A31	30A31	30A001
PML DEVICES					
PLHS501	10	133	N/A	30A22♦	30A101

NOTES:
The software and hardware revisions listed are the earliest revisions that support these products. Later revisions can also be assumed to support these products.

♦ Requires 30A101 adaptor; includes PLCC support.

1. The second revision listed is required if you wish to program the security fuse on the following products:
   PLUS20L8/R8/R6/R4 Rev. G or later
   PLUS16R8/R6/R4 Rev. I or later
   PLUS16L8 Rev. K or later

PLD programmer vendors contact guide

COMPANY	LOCATION	PERSON TO CONTACT	CERTIFICATION
Advin Systems	1050-L E. Duane Avenue Sunnyvale, CA 94086	Wing F. Hui (408) 243-7000	Pending update 75% done
American Reliance	9952 Eash Bladwin Place El Monte, CA 91731	John Goosseff Tel: (800) 654-9838 (818) 575-5110	Vendor to provide equipment
Aval Data	Daisan-Maruzen Building 6-16-6 Nishishinjuku Shinjuku-ku, Tokyo Japan 160	Toshiko Ishii 03-3344-2001	Vendor to provide equipment
B&C Microsystems	750 N. Pastoria Avenue Sunnyvale, CA 94086	(408) 730-5511	Pending new update
Basic Computer Systems AG	Wolfgang–Pauli–Gasse A–1140 Klagenfurt–Auhof, Austria	Tel: +43–222–9736360 Fax: +43–222–975915	Certified 4/89 UP2000
BP Microsystems	1000 North Post Oak Rd. Suite 225 Houston, TX 77055	Bill Cates (800) 225-2102 Fax # (713) 461-7413	Certified PLD1100, CP1128
Data I/O	10525 Willow Road, N.E. Redmond, WA 98073-9746	(800) 247-5700	Certified Model 29/60 UNISITE, S1000, 2900, 3900
Eden Engineering	12505 Loma Rica Drive Grass Valley, CA 95945	Dan Mower (916) 272-2770	Vendor to provide equipment
Elan Digital Systems	538 Valley Way Milpitas, CA 95035	(800) 541-ELAN (408) 946-8495	Pending new update
HiLo/Tribal Microsystems	44388 S. Grimmer Blvd. Fremont, CA 94538	Robert Kruger (510) 623-8860	Pending new update
Logical Devices	692 South Military Trail Deerfield Beach, FL 33442	Joleen Rasmussen (800) 331-7766 (305) 428-6868 (FL only)	ALLPRO 40 Certified 7/91
Minato	3628 Madison Avenue, Suite 5 North Highlands, CA 95660	Tel: (916) 348-6066 Fax: (916)348–0926	Certified System 1891 & 1910
Red Square Co.	2098 South Grand Avenue Suite H Santa Ana, CA 92705	Stanley Fiala (714) 751-1373	Vendor to provide equipment
SMS Sprint Plus/Expert	P.O. Box 3159 Redmond, WA 98073—3159	Bob Young (206) 883-8447	Certified Sprint Plus
	SMS – G. Steudel Im Morgental 13 D–8994 Hergatz, Germany	Tel: +49–7522–4460 Fax: +49–7522–8929	
Stag Micro Systems, Inc.	1600 Wyatt Drive, Suite 3 Santa Clara, CA 95054	Terry Hepner (408) 988-1118	Certified ZL30A
Strebor PML Support Only	1008 North Nob Hill Drive America Fork, UT 84003	Larry Roberts (801) 756-3605	Certified PLP-S1/S1A
ByTek	543 N.W. 77th Street Boca Raton, FL. 33487	Buddy Farmer 800-523-1565	Vendor to provide EQ.
Sunrise Electronics	524 South Vermont Glendora, CA 81740	Anh Le (818) 914-1926	Vendor to provide equipment
System General	510 South Park Victoria Dr. P.O. Box 361898 Milpitas, CA 95036-1898	Tim Morse (408) 263-6667	Certified - SGUP-85/85A TURPRO-1
Xeltek	757 No Pastoria Avenue Sunnyvale, CA 94086	Young Oh (408) 745-7974 (800) 541-1975	Pending new update

Approved software support

PHILIPS PRODUCT NAME	PHILIPS		ACUGEN	DATA I/O	ISDATA	LOGIC DEV	MINC
	SLICE	SNAP	ATGEN	ABEL *	LOG/IC	CUPL*	PL * Designer
	Rev	Rev	Rev	Rev	Rev	Rev	Rev
PAL® DEVICES							
10H20EV8-4	1.0	1.6	2.47	3.1	3.4	4.2A	2.1
10020EV8-4	1.0	1.6	2.47	3.1	3.4	4.2A	2.1
PHD16N8-5	1.0	1.6	2.47	4.0	3.3	2.50A	2.1
PHD48N22-7	1.0	1.6	–	4.0	3.6	4.2A	–
PL22V10-10/-12/-15	1.05	1.8	2.47	3.1	3.4	2.11A	3.0
PLC18V8Z-25/-35	1.05	1.8	–	4.1	3.4	4.2A	2.1
PLUS16L8-7/D	1.0	1.6	2.47	3.1	3.3	1.01A	2.1
PLUS16R4-7/D	1.0	1.6	2.47	3.1	3.3	1.01A	2.1
PLUS16R6-7/D	1.0	1.6	2.47	3.1	3.3	1.01A	2.1
PLUS16R8-7/D	1.0	1.6	2.47	3.1	3.3	1.01A	2.1
PLUS20L8-7/D	1.0	1.6	2.47	3.1	3.3	2.0A	2.1
PLUS20R4-7/D	1.0	1.6	2.47	3.1	3.3	2.0A	2.1
PLUS20R6-7/D	1.0	1.6	2.47	3.1	3.3	2.0A	2.1
PLUS20R8-7/D	1.0	1.6	2.47	3.1	3.3	2.0A	2.1
PLA DEVICES							
PLS100	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS101	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS153/153A	1.0	1.6	2.47	3.1	3.3	2.15A	2.1
PLS173	1.0	1.6	2.47	3.1	3.3	2.15A	2.1
PLUS153-10/D/B	1.0	1.6	2.47	3.1	3.3	2.15A	2.1
PLUS173-10/D/B	1.0	1.6	2.47	3.1	3.3	2.1A	2.1
PLS DEVICES							
PLC415-16	1.0	1.6	2.47	4.0	–	4.0A	2.1
PLC42VA12	1.05	1.8	2.47	4.1	–	4.2A	–
PLS105/105A	1.0	1.6	2.47	3.1	3.3	2.0A	2.1
PLS155	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS157	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS159A	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS167/167A	1.05	1.8	2.47	3.1	3.3	2.0A	2.1
PLS168/168A	1.05	1.8	2.47	3.1	3.3	2.1A	2.1
PLS179	1.05	1.8	2.47	3.1	3.3	3.0A	2.1
PLUS105-55/-45	1.0	1.6	2.47	3.1	3.3	3.0A	3.0
PLUS405-55/-45/-37	1.0	1.6	2.47	3.1	3.3	3.0A	2.1
PML DEVICES							
PLHS501/501I	1.0	1.6	2.47	3.1	–	3.2A	–
PML2552-35/-50	1.0	1.6	–	–	–	–	–
PML2852-35/-50	1.05	1.8	–	–	–	–	–

®PAL is a registered trademark of Advanced Micro Devices, Inc.
* See individual software guide.

Third-party software support

ABEL

Data I/O Corporation
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073–9746
Telephone: (206) 881-6444

PART NUMBER	TYPE	PACKAGE	# PINS	DEVICE FILE	ABEL REV.
10020EV8	PAL®	DIP	24	EC20EV8A	3.1
10020EV8	PAL	PLCC	28	EC20EV8AC	4.2
10H20EV8	PAL	DIP	24	EC20EV8A	3.1
10H20EV8	PAL	PLCC	28	EC20EV8AC	4.2
PHD16N8	PAL	DIP	20	P16N8	4.0
PHD16N8	PAL	PLCC	20	P16N8	4.0
PHD48N22	PAL	PLCC	68	P48N22	4.0
PL22V10	PAL	DIP	24	P22V10	3.4
PL22V10	PAL	PLCC	28	P22V10C	3.4
PLC18V8Z	EPLD	DIP	20	P18V8Z	4.1
PLC18V8Z	EPLD	PLCC	20	P18V8Z	4.1
PLC415	FPLS	DIP	28	F415	4.0
PLC415	FPLS	PLCC	28	F415	4.0
PLC42VA12	FPLS	DIP	24	F42VA12	4.1
PLC42VA12	FPLS	PLCC	28	F42VA12	4.2
PLHS501	PML	PLCC	52	PML501	3.1
PLS100	FPLA	DIP	28	F100	3.1
PLS100	FPLA	PLCC	28	F100	3.1
PLS101	FPLA	DIP	28	F100	3.1
PLS101	FPLA	PLCC	28	F100	3.1
PLS105/105A	FPLS	DIP	28	F105	3.1
PLS105/105A	FPLS	PLCC	28	F105	3.1
PLS153/153A	FPLA	DIP	20	F153	3.1
PLS153/153A	FPLA	PLCC	20	F153	3.1
PLS155	FPLS	DIP	20	F155	3.1
PLS155	FPLS	PLCC	20	F155	3.1
PLS157	FPLS	DIP	20	F157	3.1
PLS157	FPLS	PLCC	20	F157	3.1
PLS159A	FPLS	DIP	20	F159	3.1
PLS159A	FPLS	PLCC	20	F159	3.1
PLS167/167A	FPLS	DIP	24	F167	3.1
PLS167/167A	FPLS	PLCC	28	F167C	4.2
PLS168/168A	FPLS	DIP	24	F168	3.1
PLS168/168A	FPLS	PLCC	28	F168C	4.2
PLS173	FPLA	DIP	24	F173	3.1
PLS173	FPLA	PLCC	28	F173C	4.2
PLS179	FPLS	DIP	24	F179	3.1
PLS179	FPLS	PLCC	28	F179C	4.2
PLUS105	FPLS	DIP	28	F105	3.1
PLUS105	FPLS	PLCC	28	F105	3.1
PLUS153	FPLA	DIP	20	F153	3.1
PLUS153	FPLA	PLCC	20	F153	3.1
PLUS16L8	PAL	DIP	20	P16L8	3.1
PLUS16L8	PAL	PLCC	20	P16L8	3.1
PLUS16R4	PAL	DIP	20	P16R4	3.1
PLUS16R4	PAL	PLCC	20	P16R4	3.1
PLUS16R6	PAL	DIP	20	P16R6	3.1
PLUS16R6	PAL	PLCC	20	P16R6	3.1
PLUS16R8	PAL	DIP	20	P16R8	3.1
PLUS16R8	PAL	PLCC	20	P16R8	3.1
PLUS173	FPLA	DIP	24	F173	3.1
PLUS173	FPLA	PLCC	28	F173C	4.2

®PAL is a registered trademark of Advanced Micro Devices, Inc.

Third-party software support

ABEL – Data I/O Corporation (CONTINUED)

PART NUMBER	TYPE	PACKAGE	# PINS	DEVICE FILE	ABEL REV.
PLUS20L8	PAL	DIP	24	P20L8	3.1
PLUS20L8	PAL	PLCC	28	P20L8C	4.1
PLUS20R4	PAL	DIP	24	P20R4	3.1
PLUS20R4	PAL	PLCC	28	P20R4C	4.1
PLUS20R6	PAL	DIP	24	P20R6	3.1
PLUS20R6	PAL	PLCC	28	P20R6C	4.1
PLUS20R8	PAL	DIP	24	P20R8	3.1
PLUS20R8	PAL	PLCC	28	P20R8C	4.1
PLUS405	FPLS	DIP	28	F405	3.1
PLUS405	FPLS	PLCC	28	F405	3.1

Third-party software support

CUPL

Logical Devices, Inc.
1201 N.W. 65th Place
Ft. Laudedale, FL 33309
Telephone: (305) 974-0967

PART NUMBER	DEVICE MNEMONIC	# PINS	# FUSES	# OF P-TERMS	CUPL REV.
10020EV8	P1020EV8	24	3616	80	4.2a
10H20EV8	P1020EV8	24	3616	80	4.2a
82S100	F100	28	1928	48	2.00a
82S101	F100	28	1928	48	2.00a
82S105/105A	F105	28	3553	48	2.00a
82S153/153A	F153	20	1842	42	2.15a
PHD16N8	P16N8	20	512	16	2.50a
PHD48N22	F48N22	68	7008	73	4.2a
PLC18V8Z	F18V8Z	20	2689	72	4.2a
PLC415	F415	28	5751	68	4.0a
PLC42VA12	F42VA12	24	8994	10	4.2a
PLHS501	F501	52	15780	112	3.2a
PLS100	F100	28	1928	48	2.00a
PLS101	F100	28	1928	48	2.00a
PLS105/105A	F105	28	3553	48	2.00a
PLS153/153A	F153	20	1842	42	2.15a
PLS155	F155	20	2108	43	2.00a
PLS157	F157	20	2108	43	2.00a
PLS159A	F159	20	2108	43	2.00a
PLS167/167A	F167	24	3361	48	2.00a
PLS168/168A	F168	24	3553	48	2.10a
PLS173	F173	24	2178	42	2.15a
PLS179	F179	24	2452	43	3.0a
PLUS105-45/-55	F105	28	3553	48	3.0a
PLUS153B/D/-10	F153	20	1842	42	2.15a
PLUS16L8D/-7	F16L8	20	2048	64	1.01a
PLUS16R4D/-7	P16R4	20	2048	64	1.01a
PLUS16R6D/-7	P16R6	20	2048	64	1.01a
PLUS16R8D/-7	P16R8	20	2048	64	1.01a
PLUS173B/D/-10	P173	24	2178	42	2.10a
PLUS20L8D/-7	P20L8	24	2560	64	2.00a
PLUS20R4D/-7	P20R4	24	2560	64	2.00a
PLUS20R6D/-7	P20R6	24	2560	64	2.00a
PLUS20R8D/-7	P20R8	24	2560	64	2.00a
PLUS405	F405	28	5410	64	3.0a
PL22V10	P22V10	24	5828	130	2.11a

Third-party software support

PLDesigner

Minc, Incorporated
6755 Earl Drive
Colorado Springs, CO 80918
Telephone: (719) 590-1155

PART NUMBER	TEMPLATE NAME	TECHNOLOGY	PACKAGES	REVISION
PLS100	A100	TTL	DIP/PLCC	2.1
PLS101	A100	TTL	DIP/PLCC	2.1
PLS153/153A	A153	TTL	DIP/PLCC	2.1
PLUS153	A153	TTL	DIP/PLCC	2.1
PLS173	A173	TTL	DIP/PLCC	2.1
PLUS173	A173	TTL	DIP/PLCC	2.1
10020EV8-4	P20EV8	ECL	CDIP/PLCC	2.1
10H20EV8-4	P20EV8	ECL	CDIP/PLCC	2.1
PLUS16L8	P16L8	TTL	DIP/PLCC	2.1
PHD16N8-5	P16N8	TTL	DIP/PLCC	2.1
PLUS16R4	P16R4	TTL	DIP/PLCC	2.1
PLUS16R6	P16R6	TTL	DIP/PLCC	2.1
PLUS16R8	P16R8	TTL	DIP/PLCC	2.1
PLC18V8Z35	P18V8S	CMOS	DIP/CDIP/PLCC	2.1
PLUS20L8	P20L8	TTL	DIP/PLCC	2.1
PLUS20R4	P20R4	TTL	DIP/PLCC	2.1
PLUS20R6	P20R6	TTL	DIP/PLCC	2.1
PLUS20R8	P20R8	TTL	DIP/PLCC	2.1
PL22V10	P22V10	CMOS	DIP/PLCC	3.0
PLS105/105A	S105	TTL	DIP/PLCC	2.1
PLUS105	S105	TTL	DIP/PLCC	3.0
PLS155	S155	TTL	DIP/PLCC	2.1
PLS157	S157	TTL	DIP/PLCC	2.1
PLS159A	S159	TTL	DIP/PLCC	2.1
PLS167/167A	S167	TTL	DIP/PLCC	2.1
PLS168/168A	S168	TTL	DIP/PLCC	2.1
PLS179	S179	TTL	DIP/PLCC	2.1
PLUS405	S405	TTL	DIP/PLCC	2.1
PLC415	S415	CMOS	DIP/CDIP/PLCC	2.1

PLD software vendors contact guide

PRODUCT	LOCATION	CONTACT NUMBER
*Philips Semiconductors SNAP	811 E. Arques Avenue Sunnyvale, CA 94088	(800) 451-6644 Bulletin board #
ACUGEN Software, Inc. ATGEN	427-3 Amherst St., Ste. 391 Nashua, NH 03063	(603) 881-8821
Data I/O ABEL	10525 Willow Rd., N.E. Redmond, WA 98073-9746	(800) 247-5700
Logical Devices CUPL	1201 N.W. 65th Place Ft. Lauderdale, FL 33309	(800) EE1-PROM
Daisy/Cadnetix PLD Master	5775 Flatiron Parkway Boulder, CO 80301	(303) 444-8075
MINC PLD Designer	6755 Earl Drive Colorado Springs, CO 80918	Michael O. Watry (719) 590-1155
Mentor Graphics Corp. PLD Synthesis	8500 S.W. Creekside Place Beaverton, OR 97005	(503) 626-7000
OrCAD Systems ORCAD/PLD	1049 S.W. Baseline St., Suite 500 Hillsboro, OR 97123	(503) 640-9488
ISDATA LOG/ic	ISDATA GmbH Daimlerstr. 51 D–7500 Karlsruhe 21 Germany	Tel: +49–721–751087 Fax: +49–721–752634
Logic Automation	19500 N.W. Gibbs Drive P.O. Box 310 Beaverton, OR 87075	(503) 690-6900

* The SNAP phone number connects to the SPG bulletin board. Compatible with 1200/2400 baud modems, messages can be left, problem files uploaded, and solution files downloaded.

Section 11
Application Notes

PAL devices

INTRODUCTION

Philips Semiconductors provides state-of-the-art industry standard PAL® devices, both bipolar and CMOS. The range of offers spans the entire gamut of performance options; zero-standby power generic devices specified over the commercial, industrial and military temperature ranges, or the ultimate in high speed, an ECL compatible 20EV8 device. Almost every option in between is also offered.

The PAL architecture consists of a programmable AND array, followed by a fixed OR array as shown in Figure 1. The somewhat rigid architecture lends itself to less complex, narrower logic functions. There are three basic PAL-type device configurations. The XXL8 devices are strictly combinatorial. The XXRX series offers a range of registered and combinatorial outputs.

The XXV8 series is considered to be generic in nature, in that the output macros are variable (hence the "V") as combinatorial or registered. Most frequent applications include counters and shifters (the RX series), and small decoders and multiplexers (the L8 series).

The 22V10 is a popular PLD architecture. Philips Semiconductors offers both CMOS and BiCMOS versions of this device. The PL22V10 is an electrically erasable CMOS implementation. The PLQ22V10 is a low-noise, high-speed BiCMOS version with additional improved circuit characteristics.

In addition to the standard devices just described, Philips Semiconductors also offers devices tailored for specific applications. The PHD48N22 is an expanded high-speed PAL-type architecture device, with up to 48 inputs, 22 outputs and a 7.5ns propagation delay, it is optimized for very wide decoding applications. Another decoder, the PHD16N8 is optimized for decoding speed by removing the OR array.

If you are designing using ECL, Philips Semiconductors offers the 10H20EV8/10020EV8. This device provides 8 outputs which may be individually configured for registered or combinatorial operation. It also features ultra high-speed operation with a t_{PD} of 4.5ns and a f_{MAX} of 108MHz. An application note in this section demonstrates how to implement various circuits in a 20EV8 device.

Industry standard software can be used with Philips Semiconductors PAL-type devices. Full support is also provided via the Philips Semiconductors SNAP Design software.

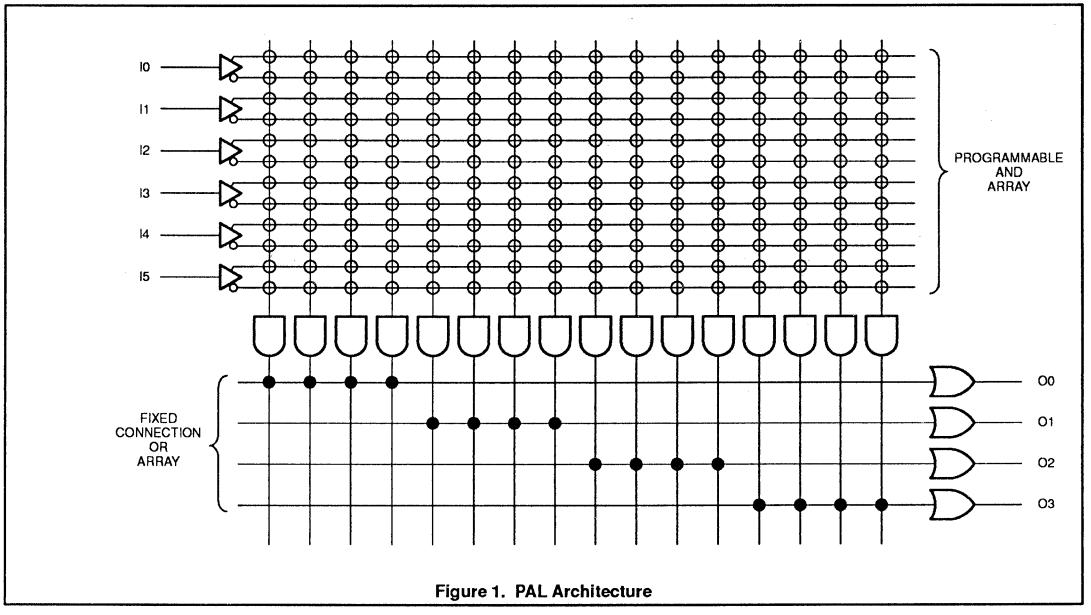


Figure 1. PAL Architecture

®PAL is a registered trademark of Advanced Micro Devices, Inc.

PAL devices

The PLUS16L8D and –7

The PLUS16L8D and –7 PAL-type devices are functionally identical to other commercially available 16L8 PAL ICs. Figure 2 shows an extremely simplified version. Less flexible than a PLA, the PLUS16L8D/–7 provides raw speed and current drive so important for driving SRAM arrays on RISC processors or the control/data lines on rapid bus structures. The PLUS16L8D has a worst-case propagation delay of 10ns. The worst-case t_{PD} of the –7 is 7.5ns. 24mA output drive is guaranteed.

The PLUS16L8D/–7 have seven product terms per OR function and one per 3-State control. Six of the eight outputs can be configured as inputs or outputs. The PLUS16L8D/–7 are available in 20-pin plastic DIL or 20-pin PLCC packages.

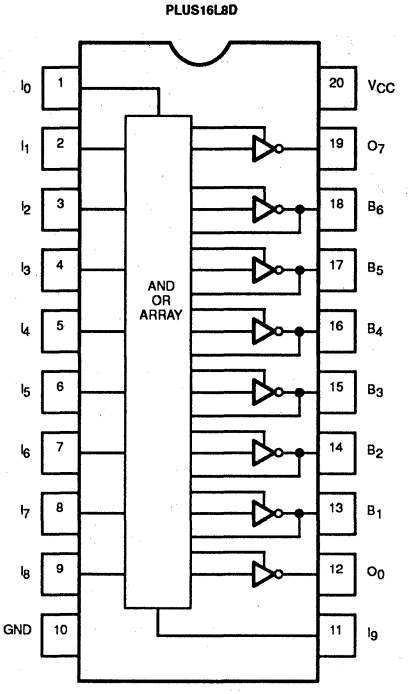


Figure 2. PLUS16L8 (D and –7)

PAL devices

The PLUS16R8D and –7

The PLUS16R8D and –7, like the PLUS16L8D and –7 is identical to other manufacturers' registered PAL devices. The parts have eight inputs, eight outputs, and eight D-flip-flops. Each flip-flop feeds an output pin through a 3-State buffer. The output of each D-flip-flop, $\overline{Q}$, is also fed back to the AND array. Each output is capable of driving 24mA I_{OL} max, with all ouputs simultaneously asserted.

The PLUS16R8D has a worst-case propagation delay of 10ns. The worst-case t_{PD} of the –7 is 7.5ns. The PLUS16R8D and –7 are available in 20-pin plastic DIP and 20-pin PLCC.

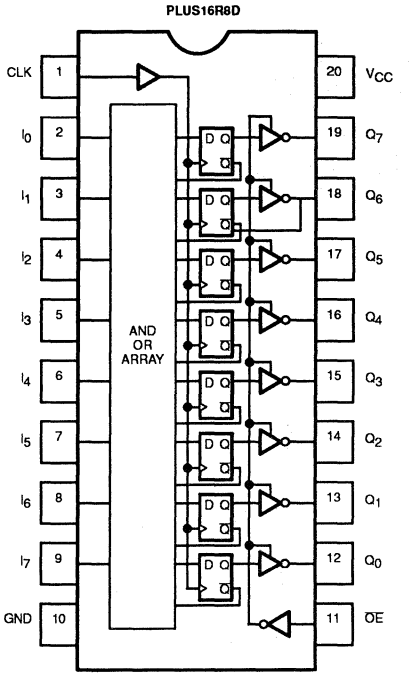


Figure 3. PLUS16R8 (D and –7)

PAL devices

The PLUS20L8D and –7

The PLUS20L8D and –7 devices have 14 inputs, two dedicated outputs and six bidirectionals. The t_{PD} are 10ns max and 7.5ns max, respectively. The 24mA of output low current of these devices can drive capacitive address line inputs and pc-board traces through long layouts. This makes the particularly suitable for driving SRAM, video DRAM, and FAST dynamic RAM arrays in 32-bit microprocessor environments.

Identical to other commercially available 20L8 PAL devices, the PLUS20L8D and –7 have 56 functional product terms which are hard-wired to eight OR gates. Each OR gate drives an Active-Low output. The 3-State control of each output is from a dedicated AND product term.

The worst-case propagation delays for the PLUS20L8D and 20L8–7 are 10ns and 7.5ns, respectively.

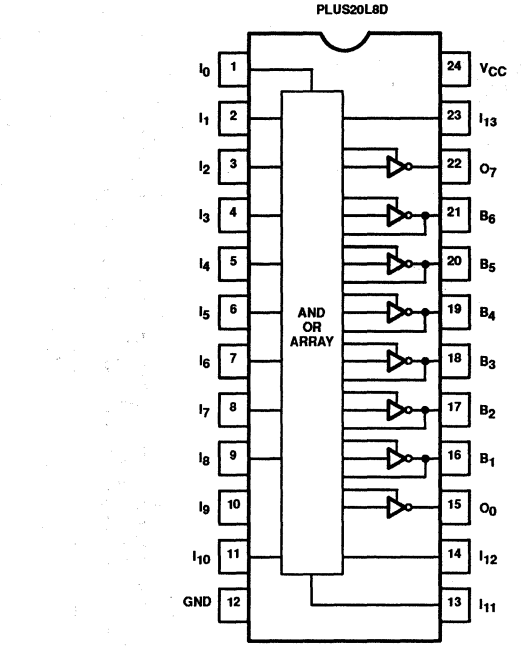


Figure 4. PLUS20L8 (D and –7)

PAL devices

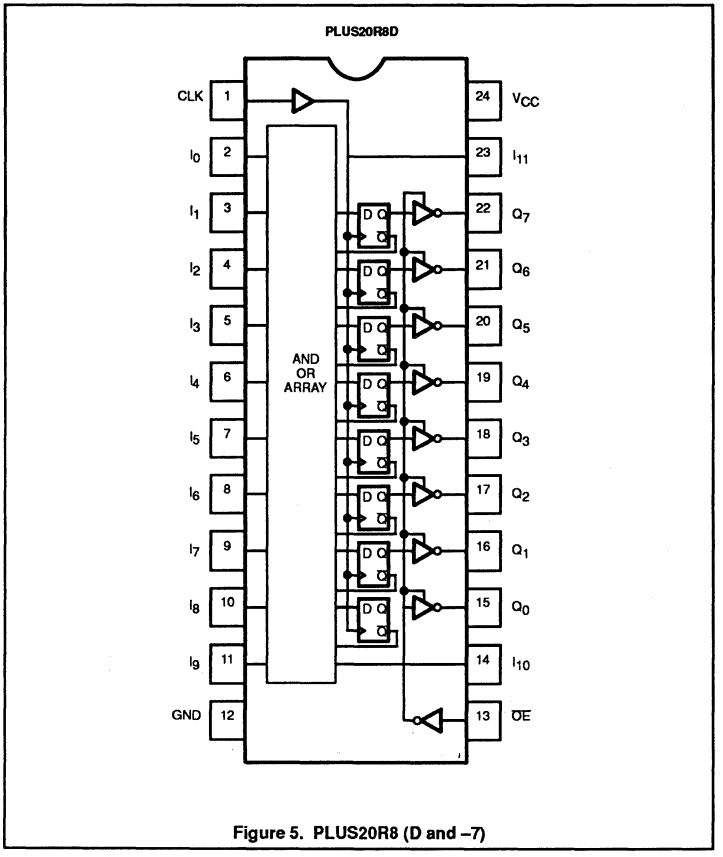


Figure 5. PLUS20R8 (D and –7)

The PLUS20R8D and –7

The PLUS20R8D and –7 are 24-pin versions of the 16R8 PAL device. With propagation delays of 10ns and 7.5ns max, the parts deliver 24mA of output low current drive. Eight D-flip-flops share a common clock and output enable line. The output of each flip-flop is dedicated to a separate output pin and is also fed back to the AND array.

The PLUS20R8D and –7 are available in 24-pin plastic DIL and 28-pin PLCC.

The PLC18V8Z

The PLC18V8Z is a multi-function, generic PAL-type device. It is pin-compatible with, and can replace 22 different 20-pin registered and combinatorial PAL devices. To accomplish this, the conventional 'single function' output pin has been replaced by a configurable Output Macro Cell. Each Macro Cell contains a D-flip-flop or a combinatorial I/O path. Output polarity and 3-State control functions are also individually configurable.

Each OMC is fed by nine AND product terms, which are hard-wired in the classic PAL fashion.

One of the key features of the part is its ability to sink 24 milliamps I_{OL}, compatible with other bipolar PAL devices—yet still comply with internal CMOS circuitry. The UV erasable version is available in 20-pin ceramic DIL with a quartz window.

10H/10020EV8 high-speed (4.4ns) ECL PLD

AN043

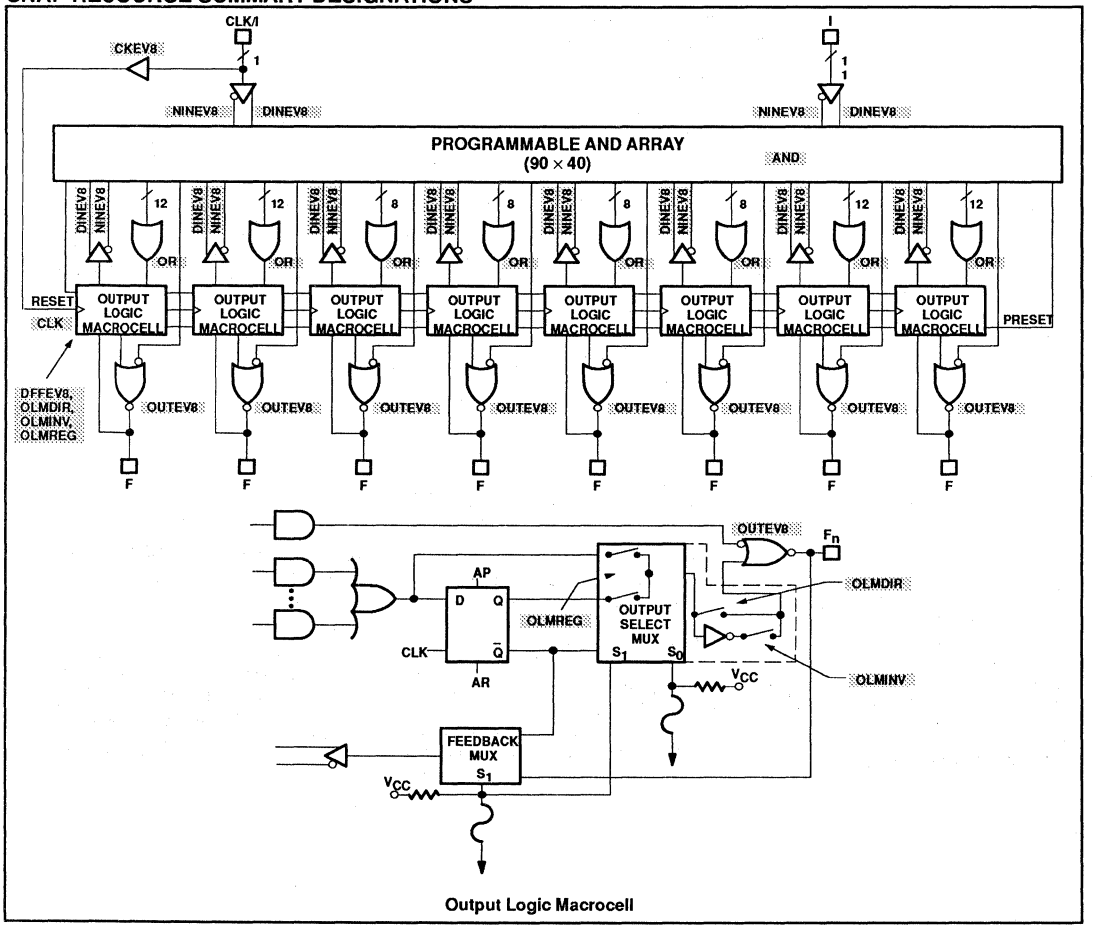
INTRODUCTION

ECL designers have never had enough chips to complete high performance designs. Period. The TTL and CMOS designers always had full MSI and LSI catalogs to rely on, and the ECL guys only had a handful of chips, to solve the hardest technological problems yet! But, that's all changed with the introduction of the Philips Semiconductors PAL™ 20EV8. Here's how.

Philips Semiconductors 10H/10020EV8 PAL devices are two 24-pin PLDs which can be configured to perform a whole catalog of parts in just two ICs. Whether you need to do simple decoding, multiplexing, counting, shifting or form complex state machines, these extraordinary parts can do it all. You can think of them as equivalents to these valuable functions, or as an empty canvas to paint your own catalog on. Don't be frustrated by the nonavailable entries in your ECL catalog, go ahead and roll your own!

Designers don't have time to read lengthy application notes, so this one was created to show you exactly how to make the basic functions you'll probably need. First, there is a single page description of the parts. Then, there is a series of four application briefs showing the exact equations, pinning, internal resource usage and simulation files for the basic mux, decoder, shifter and counter. These design files are created using Philips Semiconductors SNAP and SLICE syntax, which is simple: *=AND, +=OR and /=INVERT. These products are also supported on Data I/O ABEL™, which has a similar syntax.

SNAP RESOURCE SUMMARY DESIGNATIONS

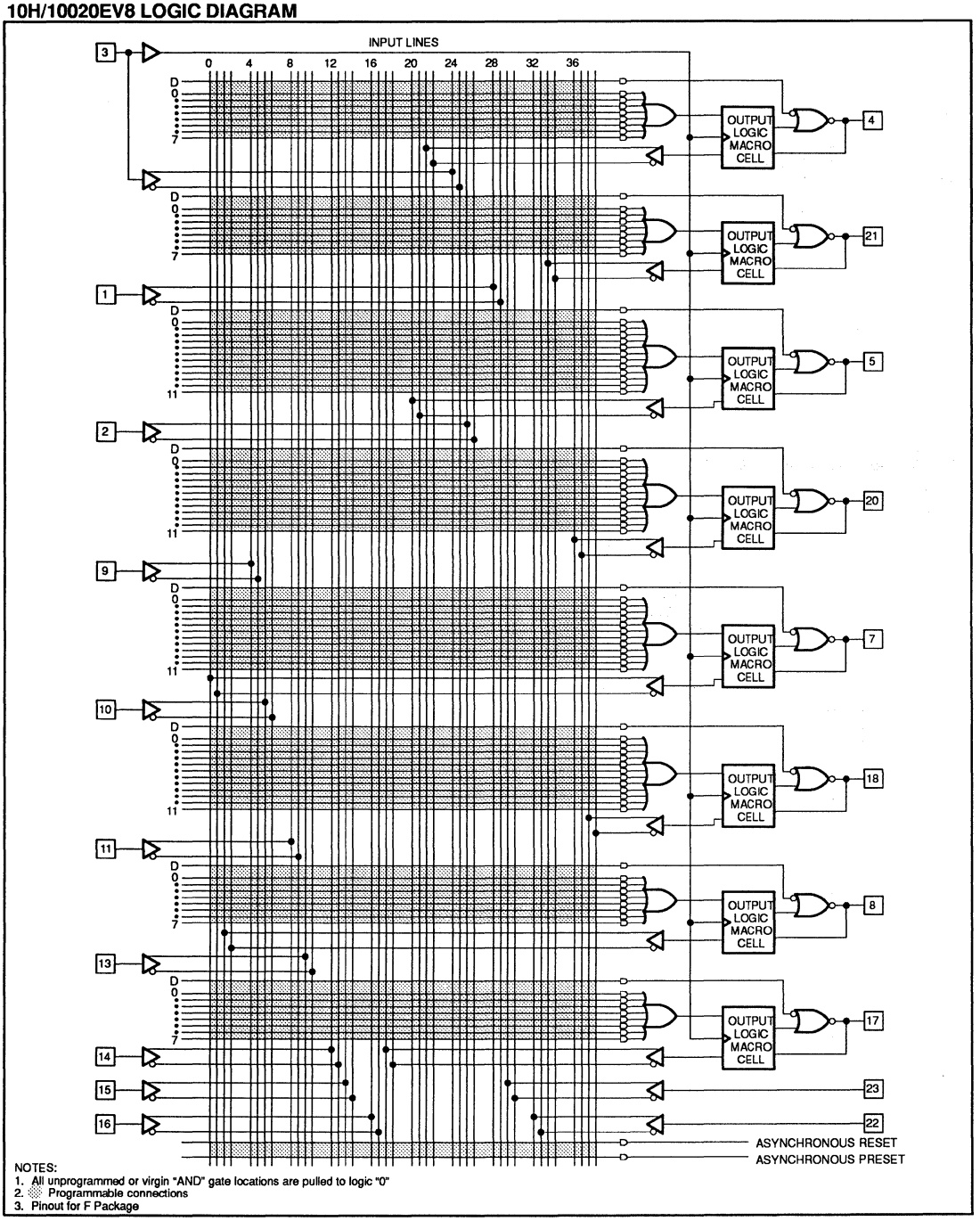


Output Logic Macrocell

PAL is a registered trademark of Advanced Micro Devices.
ABEL is a trademark of Data I/O Corporation.

10H/10020EV8 high-speed (4.4ns) ECL PLD AN043

10H/10020EV8 LOGIC DIAGRAM



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections
3. Pinout for F Package

Design Brief #1: Three-To-Eight Decoder Using SNAP/SLICE

```
@PINLIST
A0 I;A1 I;A2 I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4
 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0 =/      ( /A2* /A1* /A0);
OUT1 =/      ( /A2* /A1*  A0);
OUT2 =/      ( /A2*  A1* /A0);
OUT3 =/      ( /A2*  A1*  A0);
OUT4 =/      (  A2* /A1* /A0);
OUT5 =/      (  A2* /A1*  A0);
OUT6 =/      (  A2*  A1* /A0);
OUT7 =/      (  A2*  A1*  A0);
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device	=	0X20EV8
Pin1	=	A0
Pin2	=	A1
Pin4	=	OUT0
Pin5	=	OUT1
Pin7	=	OUT2
Pin8	=	OUT3
Pin9	=	A2
Pin17	=	OUT4
Pin18	=	OUT5
Pin20	=	OUT6
Pin21	=	OUT7

DESIGN FROM DECODE.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	0 / 1	0%
DINEV8	3 / 28	10%
NINEV8	3 / 28	10%
AND	16 / 90	17%
OR	8 / 8	100%
OLMDIR	8 / 8	100%
OLMINV	0 / 8	0%
OLMREG	0 / 8	0%
DFFEV8	0 / 8	0%
OUTEV8	8 / 8	100%

Design Brief #1: Three-To-Eight Decoder Using ABEL

```
module_decode;

decode device 'ec20ev8a';

a0,a1,a2              pin 1,2,9;
out0,out1,out2,out3   pin 4,5,7,8;
out4,out5,out6,out7   pin 17,18,20,21

inputs = [a2,a1,a0];
H,L = 1,0;

equations
out0 = ! (inputs==0);
out1 = ! (inputs==1);
out2 = ! (inputs==2);
out3 = ! (inputs==3);
out4 = ! (inputs==4);
out5 = ! (inputs==5);
out6 = ! (inputs==6);
out7 = ! (inputs==7);

test vectors
( [inputs ]  ⟶  [out0 ,out1 ,out2,  out3  ,out4 ,out5   ,out6  ,out7 ]);
  [   0   ]  ⟶  [ L   ,  H  ,H   ,  H     ,H    ,H      ,  H   ,  H   ];
  [   1   ]  ⟶  [ H   ,  L  ,H   ,  H     ,H    ,H      ,  H   ,  H   ];
  [   2   ]  ⟶  [ H   ,  H  ,L   ,  H     ,H    ,H      ,  H   ,  H   ];
  [   3   ]  ⟶  [ H   ,  H  ,H   ,  L     ,H    ,H      ,  H   ,  H   ];
  [   4   ]  ⟶  [ H   ,  H  ,H   ,  H     ,L    ,H      ,  H   ,  H   ];
  [   5   ]  ⟶  [ H   ,  H  ,H   ,  H     ,H    ,L      ,  H   ,  H   ];
  [   6   ]  ⟶  [ H   ,  H  ,H   ,  H     ,H    ,H      ,  L   ,  H   ];
  [   7   ]  ⟶  [ H   ,  H  ,H   ,  H     ,H    ,H      ,  H   ,  L   ];
  [   0   ]  ⟶  [ L   ,  H  ,H   ,  H     ,H    ,H      ,  H   ,  H   ];

end
```

Design Brief #2: Dual Four-to-One MUX with SNAP/SLICE

```
@PINLIST
A0 I;A1 I;B0 I;B1 I;C0 I;C1 I;D0 I;D1 I;SEL I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0 =    A0*SEL   +   A1*/SEL;      "MUX A0-D0 ON SEL, A1-D1 ON /SEL"
OUT1 =    B0*SEL   +   B1*/SEL;
OUT2 =    C0*SEL   +   C1*/SEL;
OUT3 =    D0*SEL   +   D1*/SEL;
OUT4 =    A1*SEL   +   A0*/SEL;      "DATA SWAP FROM ABOVE METHOD"
OUT5 =    B1*SEL   +   B0*/SEL;
OUT6 =    C1*SEL   +   C0*/SEL;
OUT7 =    D1*SEL   +   D0*/SEL;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device	=	0X20EV8
Pin1	=	A0
Pin2	=	A1
Pin4	=	OUT0
Pin5	=	OUT1
Pin7	=	OUT2
Pin8	=	OUT3
Pin9	=	B0
Pin10	=	B1
Pin11	=	C0
Pin13	=	C1
Pin14	=	D0
Pin15	=	D1
Pin16	=	SEL
Pin17	=	OUT4
Pin18	=	OUT5
Pin20	=	OUT6
Pin21	=	OUT7

DESIGN FROM MUX.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	0 / 1	0%
DINEV8	9 / 28	32%
NINEV8	1 / 28	3%
AND	24 / 90	26%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	8 / 8	100%
OLMREG	0 / 8	0%
DFFEV8	0 / 8	0%
OUTEV8	8 / 8	100%

Design Brief #2: Dual Four-to-One MUX Using ABEL

```
module_mux;

mux device 'ec20ev8a';

sel                          pin 16;
a0,b0,c0.d0                  pin 1,9,11,14;
a1,b1,c1,d1                  pin 2,10,13,15;
out0,out1,out2,out3          pin 4,5,7,8;
out4,out5,out6,out7          pin 17,18,20,21;


"mux a0-d0 to out0-out3 on sel==1
"mux a1-d1 to out4-out7 on sel==1
"mux a0-d0 to out4-out7 on sel==0
"mux a1-d1 to out0-out3 on sel==0

out1 = [out0..out3];
outh = [out4..out7];
in1  = [a0,b0,c0,d0];
inh  = [a1,b1,c1,d1];



equations

when (sel==1) then out1=in1;
when (sel==1) then outh=inh;
when (sel==0) then out1=inh;
when (sel==0) then outh=in1;

test_vectors
( [sel,in1,inh  ]   ⟶    [out1,outh]);
  [1, ^h0 ,^hf  ]   ⟶    [^h0 , ^hf];
  [1, ^ha ,^h5  ]   ⟶    [^ha ,  ^h5];
  [0, ^h0 ,^hf  ]   ⟶    [^h0 ,  ^hf];
  [0, ^ha ,^h5  ]   ⟶    [^ha ,  ^h5];
  [1, ^h1 ,^h8  ]   ⟶    [^h1 ,  ^h8];


end
```

Design Brief #3: 8-bit Counter Using SNAP

```
@PINLIST
CLOCK I;RESET I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0.D =      /OUT0;
OUT1.D =      /OUT0*OUT1
              + OUT0*/OUT1;
OUT2.D =      /OUT2*OUT1*OUT0
              + OUT2*(/OUT1 + /OUT0);
OUT3.D =      /OUT3*OUT2*OUT1*OUT0
              + OUT3*(/OUT2 + /OUT1 + /OUT0);
OUT4.D =      /OUT4*OUT3*OUT2*OUT1*OUT0
              + OUT4*(/OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT5.D =      /OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
              + OUT5*(/OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT6.D =      /OUT6*OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
              + OUT6*(/OUT5 + /OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT7.D =      /OUT7*OUT6*OUT5*OUT4*OUT3*OUT2*OUT1*OUT0
              + OUT7*(/OUT6 + /OUT5 + /OUT4 + /OUT3 + /OUT2 + /OUT1 + /OUT0);
OUT0.CLK =    CLOCK;OUT1.CLK = CLOCK;OUT2.CLK = CLOCK;OUT3.CLK = CLOCK;
OUT4.CLK =    CLOCK;OUT5.CLK = CLOCK;OUT6.CLK = CLOCK;OUT7.CLK = CLOCK;
OUT0.RST =    RESET;OUT1.RST = RESET;OUT2.RST = RESET;OUT3.RST = RESET;
OUT4.RST =    RESET;OUT5.RST = RESET;OUT6.RST = RESET;OUT7.RST = RESET;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device = 0X20EV8
Pin1 = RESET
Pin3 = CLOCK
Pin4 = OUT0
Pin5 = OUT5
Pin7 = OUT7
Pin8 = OUT3
Pin17 = OUT4
Pin18 = OUT1
Pin20 = OUT6
Pin21 = OUT2

DESIGN FROM COUNTER.N2 FOR DEVICE 10X20EV8

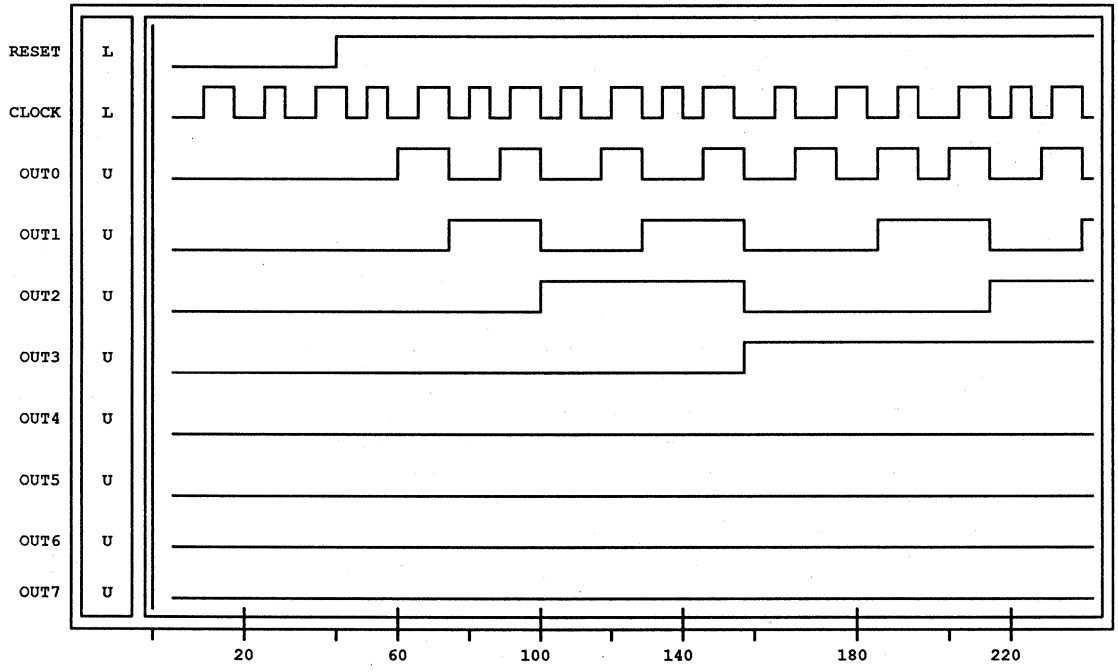
Cell Name	Used/Total	%
CKEV8	1 / 1	100%
DINEV8	8 / 28	28%
NINEV8	9 / 28	32%
AND	45 / 90	50%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	0 / 8	0%
OLMREG	8 / 8	100%
DFFEV8	8 / 8	100%
OUTEV8	8 / 8	100%

Design Brief #3: 8-bit Counter Using SNAP (continued)

```
                Output of Waveform Version 1.80
              Date: 01/20/92        Time: 17:26:48
              Input File Name     :   COUNTER.SCL
              Rule File Name :    Scl Rule
              Output File Name    :   COUNTER.SCL
```

```
P RESET, CLOCK, OUT0, OUT1, OUT2, OUT3, OUT4, OUT5, OUT6, OUT7
PCO
S 0 (40) RESET
S 0 (7,14, ETC) CLOCK
SU time = 5220
F
```

```
File:  counter.res                Delay = 0ns        Marker = 0ns        Sec/Div = 20ns
   (Model)
```




```
PHILIPS © 1992      WFA 1.80        MODE = 1                    F1 : HELP, F10 : EXIT
```

Design Brief #3: 8-bit Counter Using ABEL

```
module_count;

countb device 'ec20ev8a';

clock,reset           pin 3,1;
out0,out1,out2,out3   pin 4,18,21,8;
out4,out5,out6,out7   pin 17,5,20,7;


count = [out7..out0];
H,L,C = [1,0,.C.;


equations

count := count + 1;
count.ar = !reset;

test_vectors
( [clock  ,reset ]-->      count);
  [   L   , L    ]-->      0;
  [   C   ,L     ]-->      0;
  [   C   ,H     ]-->      1;
  [   C   ,H     ]-->      2;
  [   C   ,H     ]-->      3;
  [   C   ,H     ]-->      4;
  [   C   ,H     ]-->      5;
  [   C   ,H     ]-->      6;
  [   C   ,H     ]-->      7;
  [   C   ,H     ]-->      8;
  [   C   ,H     ]-->      9;
  [   C   ,H     ]-->      10;
  [   C   ,H     ]-->      11;
  [   C   ,H     ]-->      12;
  [   C   ,L     ]-->      0;

end
```

Design Brief #4: Octal Shifter Using SNAP

```
@PINLIST
CLOCK I;DATIN I;RESET I;
OUT0 O;OUT1 O;OUT2 O;OUT3 O;OUT4 O;OUT5 O;OUT6 O;OUT 7 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OUT0.D =      DATIN;
OUT1.D =      OUT0;
OUT2.D =      OUT1;
OUT3.D =      OUT2;
OUT4.D =      OUT3;
OUT5.D =      OUT4;
OUT6.D =      OUT5;
OUT7.D =      OUT6;
OUT0.CLK =    CLOCK;OUT1.CLK = CLOCK;OUT2.CLK = CLOCK;OUT3.CLK = CLOCK;
OUT4.CLK =    CLOCK;OUT5.CLK = CLOCK;OUT6.CLK = CLOCK;OUT7.CLK = CLOCK;
OUT0.RST =    RESET;OUT1.RST = RESET;OUT2.RST = RESET;OUT3.RST = RESET;
OUT4.RST =    RESET;OUT5.RST = RESET;OUT6.RST = RESET;OUT7.RST = RESET;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

PIN FILE

Device = 0X20EV8
Pin1 = RESET
Pin2 = DATIN
Pin3 = CLOCK
Pin4 = OUT0
Pin5 = OUT1
Pin7 = OUT2
Pin8 = OUT3
Pin17 = OUT4
Pin18 = OUT5
Pin20 = OUT6
Pin21 = OUT7

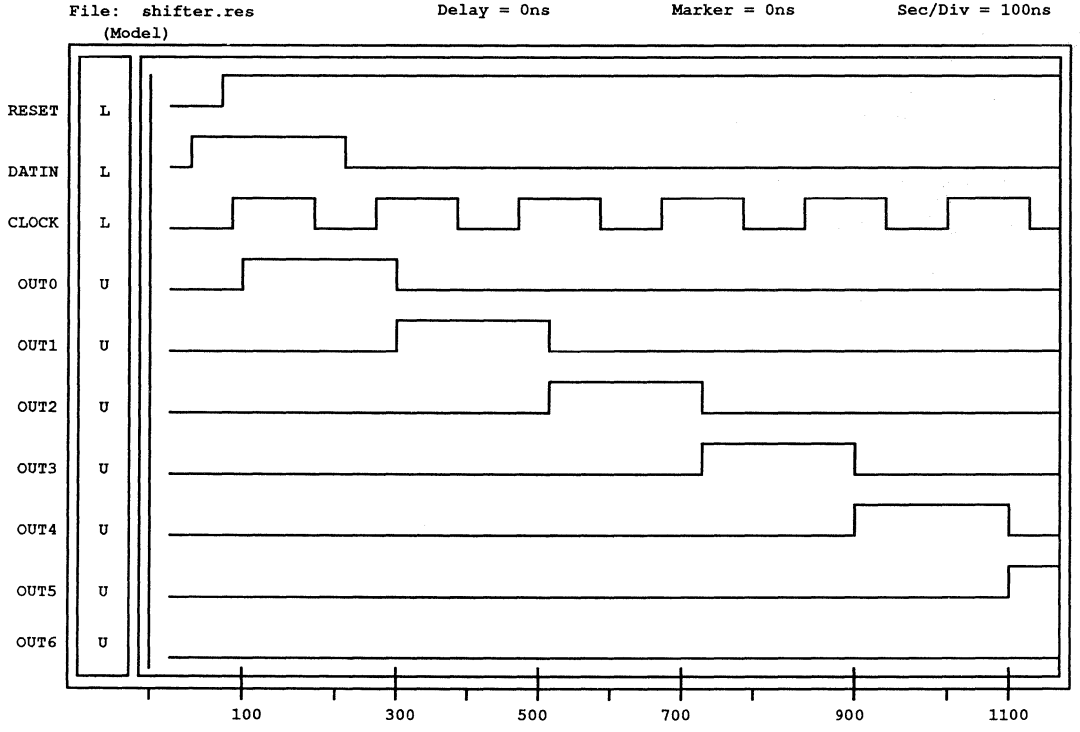
DESIGN FROM SHIFTER.N2 FOR DEVICE 10X20EV8

Cell Name	Used/Total	%
CKEV8	1 / 1	100%
DINEV8	1 / 28	3%
NINEV8	8 / 28	28%
AND	17 / 90	18%
OR	8 / 8	100%
OLMDIR	0 / 8	0%
OLMINV	0 / 8	0%
OLMREG	8 / 8	100%
DFFEV8	8 / 8	100%
OUTEV8	8 / 8	100%

Design Brief #4: Octal Shifter using SNAP (continued)

Output of Waveform Version 1.80		
Date: 01/20/92	Time: 17:15:42	
Input File Name	:	SHIFTER.SCL
Rule File Name :		Scl Rule
Output File Name	:	SHIFTER.SCL

```
P RESET, DATIN, CLOCK, OUT0, OUT1, OUT2, OUT3, OUT4, OUT5, # OUT6, OUT7
PCO
S 0 (80) RESET
S 0 (40,220,1640) DATIN
S 0 (100,200,ETC) CLOCK
S 1 (5880) VCC
S 0 (5880) GND
SU time = 5880
F
```



Design Brief #4: Octal Shifter Using ABEL

```
module_shift;

shifter device 'ec20ev8a';

clock,reset,datain    pin 3,1,2;
out0,out1,out2,out3   pin 4,5,7,8;
out4,out5,out6,out7   pin 17,18,20,21;
output = [out7..out0];
H,L,C = [1,0,.C.;


output istype 'buffer';


equations
out0.d = datain;
out1.d = out0;
out2.d = out1;
out3.d = out2;
out4.d = out3;
out5.d = out4;
out6.d = out5;
out7.d = out6;

[output].ar = !reset;

test_vectors
( [clock,reset,datain    ]  ⟶  [out0  ,out1  ,out2  ,out3,out4,out5    ,out6  ,out7 ]);
   [   0   ,   0   ,   0    ]  ⟶  [  L   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   0   ,   1    ]  ⟶  [  L   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   0    ]  ⟶  [  L   ,  H   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  L   ,  H   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   0    ]  ⟶  [  L   ,  H   ,  L   ,  H  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  L   ,  H   ,  L  ,  H  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   0    ]  ⟶  [  L   ,  H   ,  L   ,  H  ,  L  ,  H   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  L   ,  H   ,  L  ,  H  ,  L   ,  H   ,  L   ];
   [   C   ,   1   ,   0    ]  ⟶  [  L   ,  H   ,  L   ,  H  ,  L  ,  H   ,  L   ,  H   ];

   [   C   ,   0   ,   1    ]  ⟶  [  L   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  L  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  H  ,  L  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  H  ,  H  ,  L   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  H  ,  H  ,  H   ,  L   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  H  ,  H  ,  H   ,  H   ,  L   ];
   [   C   ,   1   ,   1    ]  ⟶  [  H   ,  H   ,  H   ,  H  ,  H  ,  H   ,  H   ,  H   ];
   [   C   ,   0   ,   1    ]  ⟶  [  L   ,  L   ,  L   ,  L  ,  L  ,  L   ,  L   ,  L   ];


end
```

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031

INTRODUCTION

Performance specifications for PLDs usually include simple parameters like t_{PD} and f_{MAX}. These values oversimplify PLD performance. Specifying the reaction time for an output pin to an input signal (t_{PD}) is important, but t_{PD} doesn't tell the whole story. Similarly, the flip-flop f_{MAX} shows only the maximum upper limit toggle rate. Neither number tells whether the PLD can even solve the problem at hand. To get that answer, the designer has to go beyond these simple parameters and investigate the design's exact needs.

THE DECODE PROBLEM

A recent computer architecture book (Hennesy and Patterson) shows several guidelines for fast processor design. These principles are commonly associated with RISC processors.

The primary guideline is simple: make the common case fast. For microprocessors, this means make the basic machine cycle as fast as possible. Since the basic cycle is repeated continuously, it defines nearly 100% of what the processor is doing.

Trimming time off the basic instruction cycle results in the single highest payoff. This is simple, right? Take the instruction cycle to zero wait states and the problem is solved. The problem more often yields to a quicker, universal solution – money! Put the fastest possible high speed memory into the processor – memory loop and the design is done. Naturally, there are always competitors who try to make the same design for less cost. All designers must find trade-offs giving maximum speed for minimum cost.

Getting a zero wait state solution for the lowest cost is the goal. Naturally, the cheapest solution is also the slowest one. For this problem, the designer must use the slowest memory devices that meet the zero wait state goal. We'll focus simply on that goal. It gets more interesting if some transactions can be done with one or two wait states while others require zero.

SOLVING THE PROBLEM

What must be done? When the processor emits its address and control signals, the memory space should not respond until specific modules have been explicitly selected. This is the basic problem of fast address decode.

Address decode has several dimensions: speed, resolution and current drive. Speed is obvious: it is the simple t_{PD} of the part – with the output reacting in the low direction to drive a RAM chip enable. Secondary issues include switching edge rates, reflections and contention. Current drive is the next important dimension because most small systems need at least 24mA (I_{OL}) to drive an array of chip enables, and pc board traces. The resolution dimension is the most interesting for the following reasons.

Today's high performance microprocessors have 32 address bits, along with several status and control bits. Careful decode may require as many as 36 or more distinct inputs to select bytes from the four gigabyte address space. In the past, it was acceptable to waste whole memory sections if chip decodes couldn't pack memory tightly. I/O regions were mapped into the memory space, leaving address gaps. Alternately, I/O device addresses would have repeated response regions in memory. Both solutions are less than desirable.

This arrangement contradicts today's prevailing design philosophy. Tight, contiguous memory regions are now desirable, while maintaining design freedom. Most software developers also resent giving up any memory space due to poor hardware design.

How fast does address decode have to be? Today, 33MHz and 50MHz processors are being shipped. This sets memory cycles at 33 and 20 nanoseconds, respectively. To get zero wait states Fast cache (at 20ns or less) is needed. Fast cache cost rises dramatically with 5ns speed increments. Suppose a 25ns cache were used on a 33ns cycle time. This leaves 8ns to decode and drive signal capacitance. For this analysis, we'll assume no line ringing or other electrical phenomenon. A decode propagation delay time below 7ns is needed.

To solve this decoding problem, a new PLD – the PHD48N22 – has been developed with a minimal t_{PD} and an adequate current drive. The PHD48N22 I/O structure (Figure 1) is its main strength. As the name implies, the PHD48N22 combines the potential for 48 inputs with 22 outputs. Combining high-speed configurable NAND gates with a PAL® architecture, the PHD48N22 is dual optimized. It easily handles high speed SRAMs, EPROMs, DRAMs and I/O.

By decoding 32 bits of address lines and several control lines, the PHD48N22 distinguishes reads from writes and I/O from memory to a byte level of resolution. This meets the needs of any system today. If the 22 outputs are not enough, another PHD48N22 can be added, to complete the system decode. Let's describe the part, showing some of its less obvious capabilities.

A LOOK INSIDE THE PHD48N22

Figure 1 is a logic diagram of the PHD48N22. It reveals three distinct logic regions, all connected to the main programming array. First, there are 18 AND gates, then there are three seven-input PAL sites and one 12-input PAL site. The speed of the paths which follow an input pin to an AND gate, inverter and output pin is shown in Figure 2 as solid dots. The speed path passing through the PAL sites is shown as hollow dots.

The number of outputs driven versus t_{PD} is also shown in Figure 2. The address decode function often needs only one output selected at any point in time, so it makes sense to show the speed with only one output driven. One output driven is the fastest configuration and is also a common case. Pyramided solutions that stack gates in series are unacceptable for very fast decode.

Note that the PHD48N22 outputs are 3-State controlled with logic gates programmable from the array. These 3-State, controlled outputs permit PHD48N22 outputs to be wire-ANDed together.

Wire-ANDing permits two important benefits. First, the logic function through the programmable AND gates becomes the NAND-AND when two outputs are joined. This connection is logically equivalent to the AND-OR-INVERT. The AND-OR-INVERT is exactly the logic function made by the PAL16L8. Second, wire-ANDing the gates can double the output drive to 48mA if the outputs are enabled by the same control logic. This approach avoids feeding an output back into the array just to drive another output, or serial buffering. Serial buffering increases time delay. Alternately, drive current can be split from different output pins instead of wire-ANDing them. Current splitting also increases time delay, but much less so than serial buffering.

For designers that are reluctant to use the wire-AND, some logic flexibility can occur at the four PAL sites. The usual sum-of-products can be made at these sites, and the speed is acceptable for memory decode. The PAL sites are somewhat slower than the high speed AND paths, and are ideal for controlling the processor wait line for I/O.

The PAL site outputs can also be 3-State controlled so the logic function can expand if

® PAL is a registered trademark of AMD.

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031

needed. Any mix of the PAL sites and the high speed AND gates may be had at the output pins, but the designer should be aware that the final output function is limited to the wire AND. This means that any low driving output signal dominates at the pin, if its 3-State enable is asserted.

A very simple policy for the use of the AND terms is to put the same logic function on the 3-State controlling term as is applied to the AND gates' logic inputs (Figure 3). Exotic enabling conditions can be made by using the seven input PAL sites to make and gate enable signals – wrapping around the part through the I/O pins.

The gate resolution is variable up to 48 inputs. Typically, designers will use 32 or fewer. Even with this, designers can resolve a single byte location out of a four gigabyte address space. Memory mapped I/O will no longer fragment an otherwise clean address space. It is possible to drop I/O registers (UAR/Ts, disks, etc.) right into the middle of a large memory space giving up no addresses except those needed for I/O select.

BACK TO THE SYSTEM. . .

A common practice today is to make memory and I/O devices shadow each other in the same address space. This is a cinch with the PHD48N22.

The best address partitioning with the PHD48N22 is to simply use the fast AND paths to handle memory, and the PAL sites to handle I/O decode needs. The PAL sites can be used to control the wait line on the processor for slower decodes, handle interrupt input lines, DMA requests, or whatever.

OTHER APPROACHES

It makes sense to consider the other solutions. Let's start with the way most designers have been solving the problem over the last few years. The most obvious way is fast PALs. Until recently, these parts offered the finest address resolution with as many as 21 inputs. This is the case with the 22V10, configured as an address decoder.

The number of 22V10 I/O pins leaves much to be desired, as well as the best case speed of 7.5ns. The output current drive is also low – between 12mA and 16mA depending on the manufacturer. Similarly, 5ns PAL devices of other varieties have the current drive and speed, but not enough inputs and outputs.

Considering other parts optimized for fast decode, the TIBPAD-6 is such a simple programmable NAND circuit. This part is pinned compatible to the PAL16L8, and has the requisite current drive, but not enough input pins. The Philips Semiconductors PHD16N8 is pin compatible and faster with a t_{PD} of 5ns. The PHD16N8 is the little sister of the PHD48N22. A logic diagram is shown in Figure 8.

The Xilinx LCA 4000 offers four quick decoders at 10ns and up to 60 inputs. The speed is 10ns and the output current is only 12mA. If the output current must be increased with an extra buffer, the horserace is lost. Paralleling output pins uses up pins very quickly.

The Intel 85C508 offers a fast (7.5ns) registered comparator, but it again only provides 12mA of output current. None of these solutions offer a fully integrated system decoder in a single chip.

MORE TRICKS

The PHD48N22 can build up other functions than simple, fast address decoders. For instance, Figure 4 shows a 48N22 exploiting its input width working with a deep counter. When a decode value occurs on the counter outputs, the NAND gate generates a pulse to either reset or load the counter with a new value.

Any value – from 1 to 16 bits (shown here) – can expand to 48 bits.

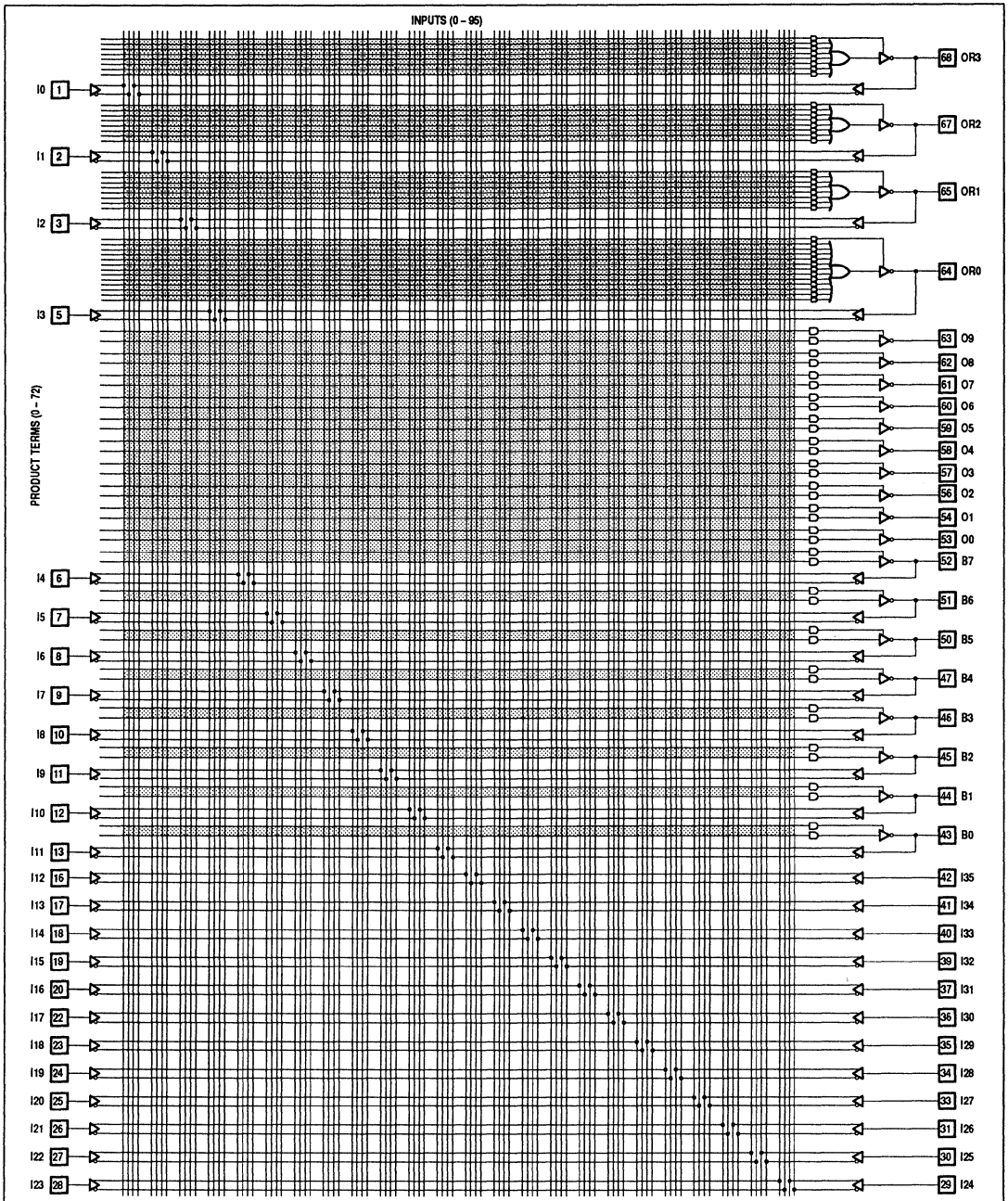
Fancy latches can be built, or custom arbiters based on the configuration shown in Figure 5 for multiple DMA contenders vying for selection.

Other logic structures not needing the signal width are possible to build with a 48N22, if any pins and functions remains after its primary use is accomplished. Figure 6 shows a simple sum of logical products and Figure 7 shows a multiplexer. Both require output signals fold back into the array and neither needs an external pull-up resistor.

In conclusion, the most common way to use the 48N22 is in its intended configuration, as a very wide input high speed address decoder. Literally, any high performance microprocessor can be used with the PHD48N22 resulting in a fast, clean design.

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒ Programmable connections

Figure 1. PHD48N22-7 Logic Diagram

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031

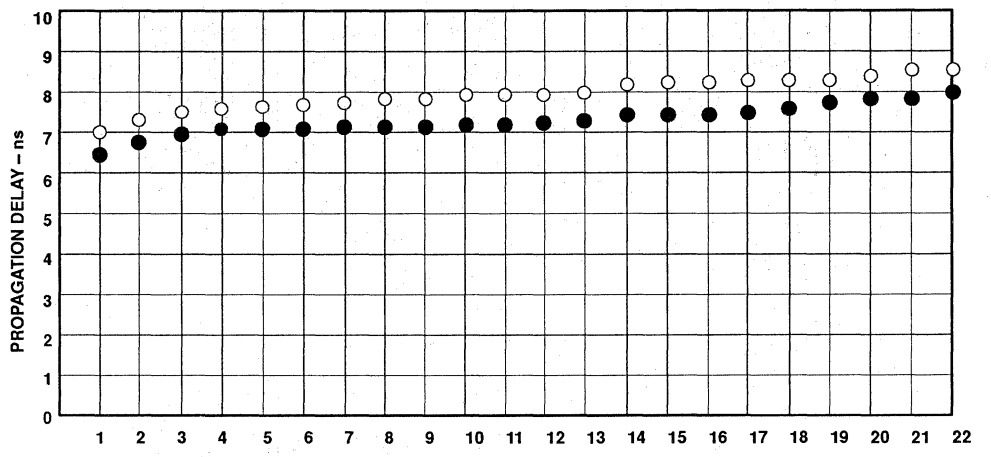


Figure 2. Worst Case Propagation Delay vs. Number of Outputs Switching

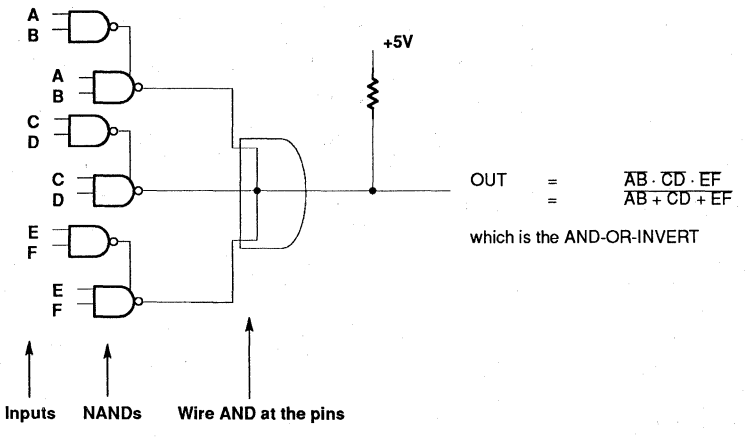


Figure 3. Wire ANDing

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031

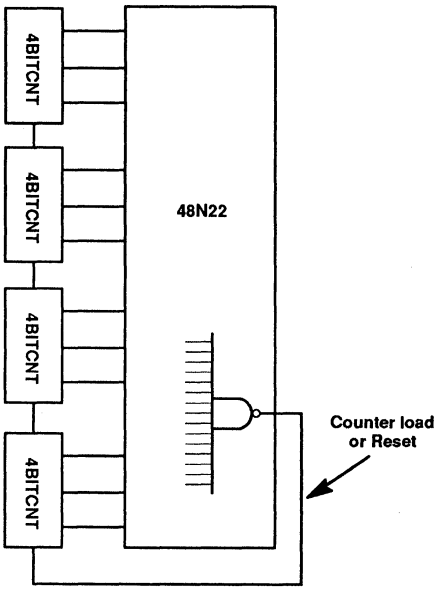


Figure 4. High Speed Count Decode

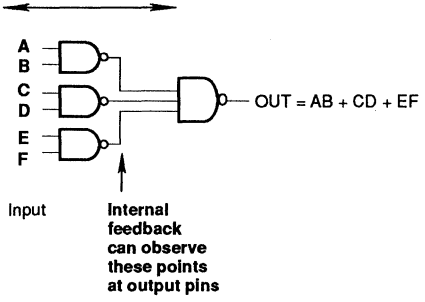


Figure 6. Two-level Sum of Products

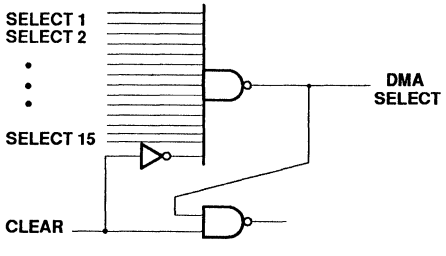


Figure 5. Wide Input Latches

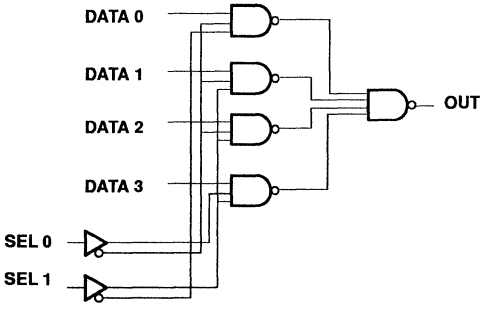
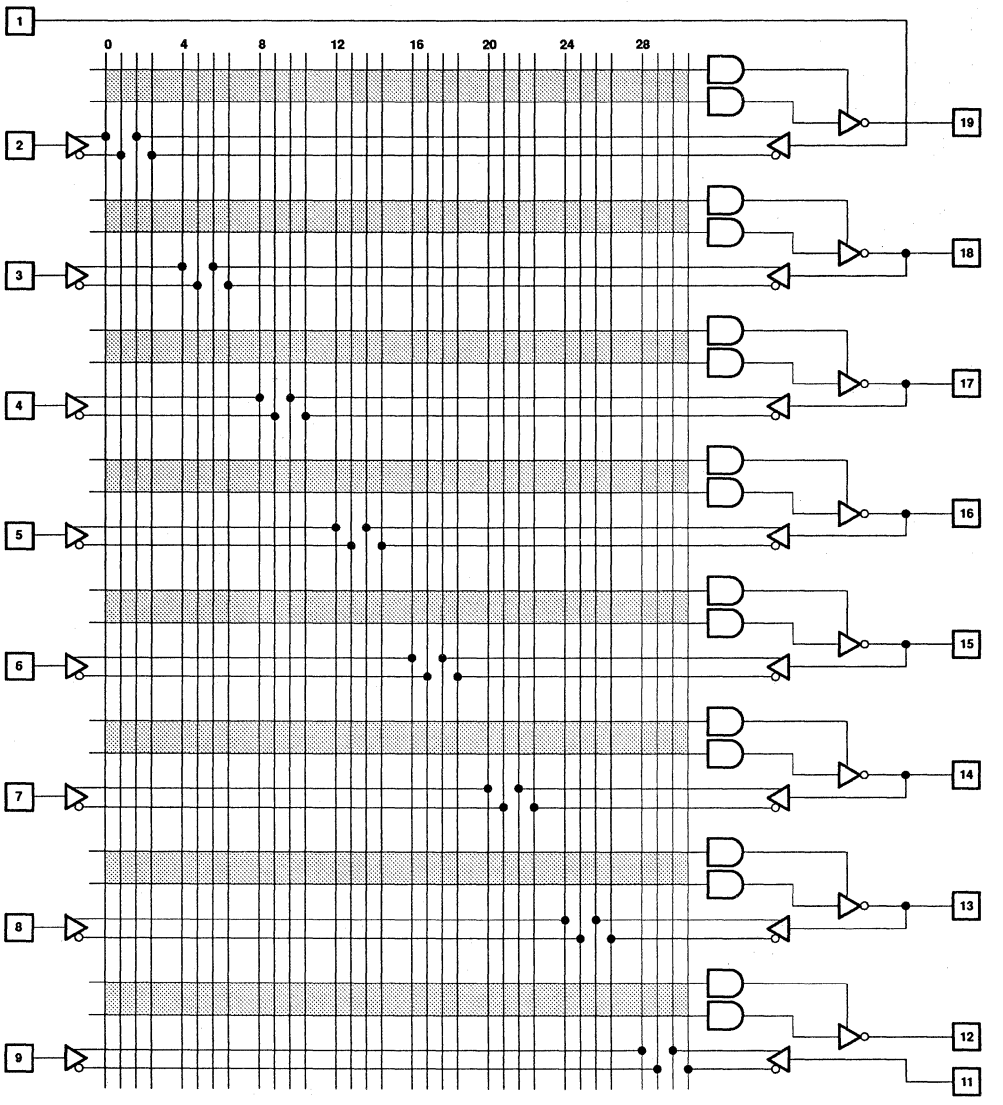


Figure 7. Four-input MUX

PHD48N22 high speed (7.5ns) 32-bit programmable decoder

AN031



NOTES:
1. All unprogrammed or virgin "AND" gate locations are pulled to logic "0"
2. ▒▒ Programmable connections

Figure 8. PHD16N8–5 Logic Diagram

68030 system decoding AN044

68030 ADDRESS DECODING

Philips Semiconductors PHD48N22 decoder is not only very high speed but also has a very wide input structure. This combination allows high-performance systems to be constructed in a minimal space without sacrificing decoding resolution. The following example demonstrates the use of the PHD48N22 as well as the PHD16N8 in a 33MHz 68030 system.

This example uses one each 48N22 and 16N8 PHD decoders to interface together a 33MHz 68030 processor, 33MHz 68882 Floating-Point Coprocessor, 128K bytes of 35ns static RAM, 64K bytes of 200ns EPROM and a SCN2692 Dual port UART. Additional logic required amounts to only one 16-pin and one 24-pin device.

The schematic is shown in Figure with SNAP listings of the PHD devices in Figures and . As can be seen from the schematic, the PHD48N22 handles all of the RAM, PROM and DUART decoding. Notice the number of 68030 address lines input into the 48N22: A31 through A8 and A1 and A0. Depending upon the application, the 48N22 can decode down to the byte level in 6.5ns. The PHD16N8 decodes an early chip-select signal for the 33MHz 68882 FPCP in less than the required 5ns.

Accesses to the RAM produce an immediate acknowledge (DSACK) from the 48N22 to the 68030, allowing it to run full speed in asynchronous mode. Since the EPROM and DUART operate at a much slower speed than the 68030, two additional devices, a 74F191 counter and a 74F646 transceiver/register were used in this example. If either the EPROM or DUART are accessed, the counter will count CPU clock cycles and delay the 48N22's assertion of DSACK signal. The 'F646 is used to quickly 3-State signals from the EPROM and DUART. Also, in conjunction with the counter and 48N22, a read of a DUART register first causes information to be read into a register of the 'F646, which is then read by the 68030 while the DUART's output is disabled. This was done since a DUART specification (t_{RWD}) requires 200ns between reads or writes. Therefore, accesses to the DUART are controlled by the counter, 48N22, and 'F646. Software restrictions are not required.

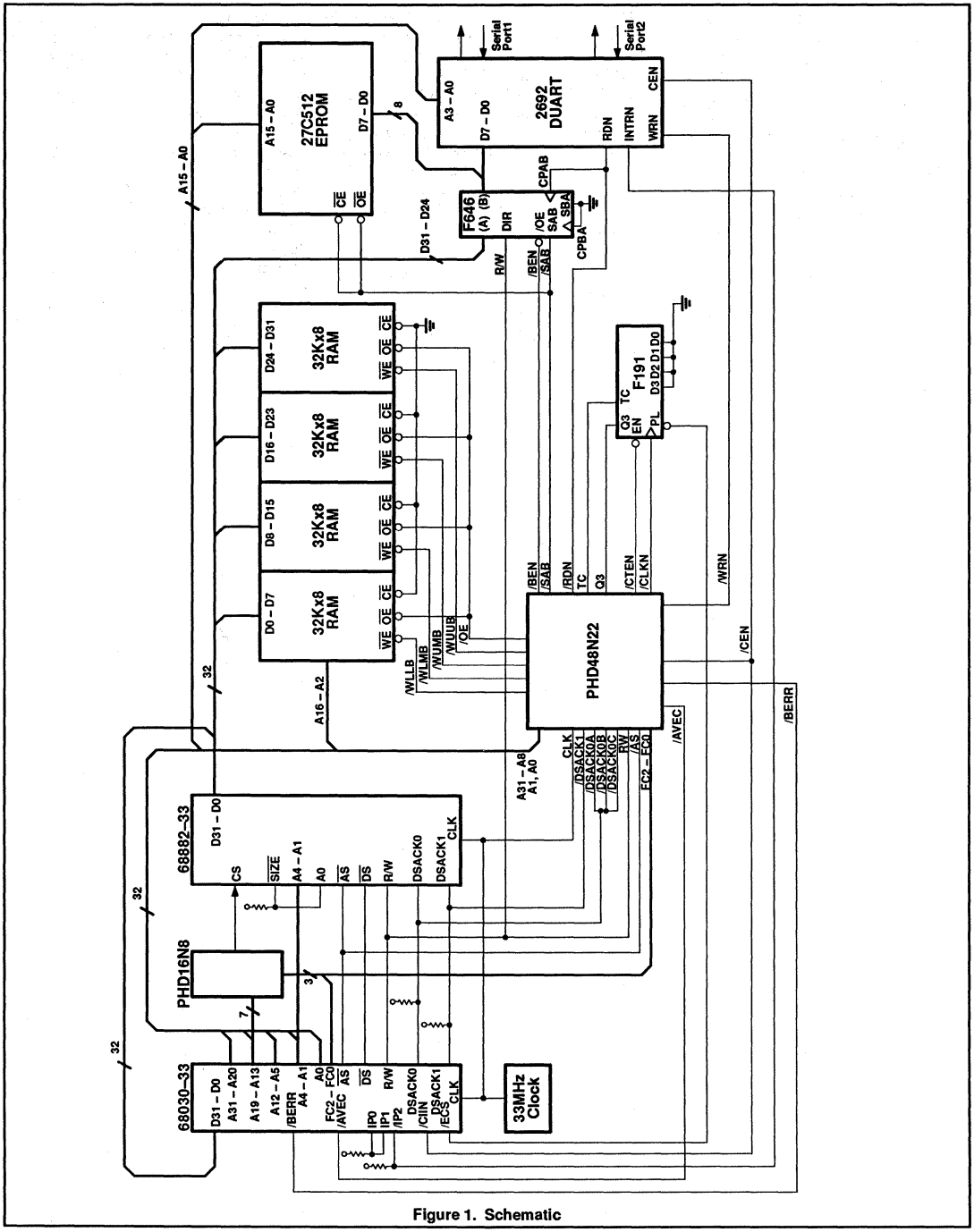


Figure 1. Schematic

```
@PINLIST
"Signal names preceding with a 'n' means active low signals"

A0              I;
A1              I;
nAS             I;
Q3              I;
TC              I;
CLK             I;
A8              I;
A9              I;
A10             I;
A11             I;
A12             I;
A13             I;
A14             I;
A15             I;
A16             I;
A17             I;
A18             I;
A19             I;
A20             I;
A21             I;
A22             I;
A23             I;
A24             I;
A25             I;
A26             I;
A27             I;
A28             I;
A29             I;
A30             I;
A31             I;
FC0             I;
FC1             I;
FC2             I;
SIZ0            I;
SIZ1            I;
RW              I;
nOE             B;
nCPU            B;
nSAB            B;
nASN            B;
nWUUB           B;
nCEN            B;
nBEN            O;
nRDN            O;
nWRN            O;
nDSACK0a        O;
nDSACK0b        O;
nDSACK0c        O;
nDSACK1         O;
nBERR           O;
nAVEC           O;
nCLKN           O;
nCTEN           O;
nWUMB           B;
nWLMB           B;
nWLLB           B;

@LOGIC EQUATIONS

                "RAM, PROM, and UART address definitions"
PROMADR = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A19*/A18*/A17*/A16;
RAMADDR = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A19*/A18*/A17;
UARTADR = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A19*/A18*/A17*A16*/A15*/A14*/A13*/A12*/A11*/A10*/A9*/A8;
UORPROM = /A31*/A30*/A29*/A28*/A27*/A26*/A25*/A24*/A23*/A22*/A21*/A20*
          /A19*/A18*/A17;
ASD     =  nAS * nASN;            "eliminate hazard ANDing nAS with nCPU"
```

Figure 2. PHD48N22-7 SNAP Listing (1 of 2)

```
nOE.oe = 1;
nCPU.oe = 1;
nSAB.oe = 1;
nASN.oe = 1;
nWUUB.oe = 1;
nCEN.oe = 1;
nCTEN.oe = 1;
nWUMB.oe = 1;
nWLMB.oe = 1;
nWLLB.oe = 1;

          "enables for DSACK0 and DSACK1"
nDSACK0a.oe = (PROMADR*nCPU*ASD*Q3);   "acknowledge PROM 8 bits"
nDSACK0b.oe = (UARTADR*nCPU*ASD*TC);   "acknowledge USART 8 bits"
nDSACK0c.oe = (RAMADDR*nCPU*ASD);      "acknowledge RAM 32 bits"
nDSACK1.oe  = (RAMADDR*nCPU*ASD);      "acknowledge RAM 32 bits"

nASN = /nAS; "delay nAS"

nDSACK0a = /(PROMADR*nCPU*ASD*Q3);   "acknowledge PROM 8 bits"
nDSACK0b = /(UARTADR*nCPU*ASD*TC);   "acknowledge USART 8 bits"
nDSACK0c = /(RAMADDR*nCPU*ASD);      "acknowledge RAM 32 bits"
nDSACK1  = /(RAMADDR*nCPU*ASD);      "acknowledge RAM 32 bits"

        "CPU address space decoding"
nCPU  = /(FC0*FC1*FC2);
nAVEC = /(FC0*FC1*FC2*A19*A18*A17*A16); "interrupt ack forces autovector"

        "RAM output enable"
nOE   = /(RAMADDR*nCPU*RW);              "enable all RAM outputs upon read"

        "RAM write strobes"
nWLLB = /(RAMADDR*/RW*A1*A0*nCPU*ASD       "directly addressed, any size"
        +RAMADDR*/RW*A0*SIZ1*SIZ0*nCPU*ASD "old alignment, 3 byte size"
        +RAMADDR*/RW*/SIZ1*/SIZ0*nCPU*ASD  "any RAM address, long word size"
        +RAMADDR*/RW*A1*SIZ1*nCPU*ASD);    "word aligned, word or 3 byte size"

nWLMB = /(RAMADDR*/RW*A1*/A0*nCPU*ASD        "directly addressed, any size"
        +RAMADDR*/RW*/A1*/SIZ1*/SIZ0*nCPU*ASD"word aligned, long word size"
        +RAMADDR*/RW*/A1*SIZ1*SIZ0*nCPU*ASD  "word aligned, 3 byte size"
        +RAMADDR*/RW*/A1*A0*SIZ0*nCPU*ASD);  "word aligned, word or long word"

nWUMB = /(RAMADDR*/RW*/A1*A0*nCPU*ASD      "directly addressed, any size"
        +RAMADDR*/RW*/A1*/SIZ0*nCPU*ASD    "word aligned,byte or 3 byte size"
        +RAMADDR*/RW*/A1*SIZ1*nCPU*ASD);   "word aligned, word or long word"

nWUUB = /(RAMADDR*/RW*/A1*/A0*nCPU*ASD);   "directly addressed, any size"

        "DUART and PROM address decoding"
nCEN = /(UARTADR*nCPU*ASD);

nRDN = /(UARTADR*nCPU*ASD*/Q3*RW);

nWRN = /(UARTADR*nCPU*ASD*/Q3*/RW);

nBEN = /(UORPROM*nCPU*ASD);         "enable F646 for USART or PROM"

nSAB = /(PROMADR*nCPU*ASD*RW);      "F646 transparent for PROM,
                                     registered for USART"

nCTEN = /(UARTADR*nCPU*ASD*/TC          "hold count upon TC for UART or"
        +PROMADR*nCPU*ASD*/Q3);         "Q3 for PROM access"

nCLKN = /(CLK*nAS);                     "clock for F191"

nBERR = /(nCEN*nWUUB*nWUMB*nWLMB*nWLLB*nOE*nCPU*nSAB*ASD);"nBERR if no access"
```

Figure 2. PHD48N22-7 SNAP Listing (2 of 2)

```
@PINLIST

FC2     I;
FC1     I;
FC0     I;
A19     I;
A18     I;
A17     I;
A16     I;
A15     I;
A14     I;
A13     I;
nCS     O;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

nCS.OE = 1;

CPU             = FC2 * FC1 * FC0;              "cpu space"
COPROCESSOR     = /A19 * /A18 * A17 * /A16;     "coprocessor communications"
CP_ID           = /A15 * /A14 * A13;            "cp-id one"

nCS = /(CPU * COPROCESSOR * CP_ID);
```

Figure 3. PHD16N8-5 SNAP Listing

High speed 8-bit parallel to serial converter AN045

INTRODUCTION

A common function in many systems is to convert parallel data into a serial data stream. A microcontroller may be programmed to shift a byte in a register out to a port, but this is a relatively slow procedure. A simple pre-loadable shift register could perform the basic conversion. However, for the function to be complete, additional circuitry to perform handshaking or control of the process is required. The entire function can be made to fit into a low cost Programmable Logic Device (PLD), including control circuitry tailored to meet specific application requirements.

DESCRIPTION

Figure 1 shows the desired waveforms for a typical implementation. First, a reset signal initializes the system and this circuit. Next, the parallel data to be serialized is applied to the device, possibly from a parallel port of a microcontroller, and a write strobe (WRS) signal pulsed. The PLD then raises a flag (BUSY) and puts the data, one bit at a time, on an output (SDAT) under control of a clock signal (CLK). Another output, (SCLK) is an inverted copy of the transmitting clock, ANDed with a control signal, so it only is active when data is actually being sent. It can be used by the receiving device to clock in the serial data.

How does one get a PLD to perform such a function? Preferably this design should fit into a simple, low cost device such as a 22V10 type PLD. A 22V10 has ten outputs which may be individually configured to be registered or combinatorial. It is possible to make a two input multiplexer circuit in front of eight of the D-type flip-flops. It could then be configured to shift data or load parallel data upon a control signal and clock. However, to provide the output control signal BUSY and gate SCLK, a 3-bit counter will be required to indicate when the last bit of data is shifted out. That would bring the total registers in the design up to eleven, one more than a 22V10 provides. Additionally, the write strobe (WRS) is a short duration asynchronous signal, so more circuitry is still required to synchronize it with the transmitting clock (CLK).

Another method of serializing data is to use a multiplexer (8 to1 for this example) and a counter. The counter controls which bit is to be output from the multiplexer. A count of zero connects input ID0 to the output, a count of one connects ID1, and so on. This will work only if the parallel input data is held stable throughout the serialization process. For this example, the data is applied from one port of a microcontroller and held stable until after the BUSY signal transitions from high-to-low, so a multiplexer will work for this case. An 8-to-1 multiplexer will use only one output, while the three-bit counter will use three outputs of a 22V10, which leaves us with six outputs for other functions. Let's use this technique to implement this example. Additional outputs are required for signals BUSY, SCLK, and some currently unspecified control signals.

A counter may be constructed very easily using a SNAP syntax equation of: "COUNT.D=COUNT#1H;". The "#" (pound) symbol means addition, the ".D" signifies an input to a D-type of flip-flop, and the "1H" is 1 hexadecimal. So the equation is simply COUNT equals COUNT plus 1. The actual equation in Figure 3 contains another term, but more on that later. In addition to the D inputs of the flip-flops, it is necessary to describe the flip-flops clock and reset connections. Those are listed in lines 57and 58 of Figure 3.

A multiplexer is also very easy to describe using SNAP syntax Boolean equations. For an 8-to-1 multiplexer with output SDAT and inputs ID7-ID0 it is:

```
SDAT = ID0 * (COUNT==0H)
     + ID1 * (COUNT==1H)
     + ID2 * (COUNT==2H)
     + ID3 * (COUNT==3H)
     + ID4 * (COUNT==4H)
     + ID5 * (COUNT==5H)
     + ID6 * (COUNT==6H)
     + ID7 * (COUNT==7H);
```

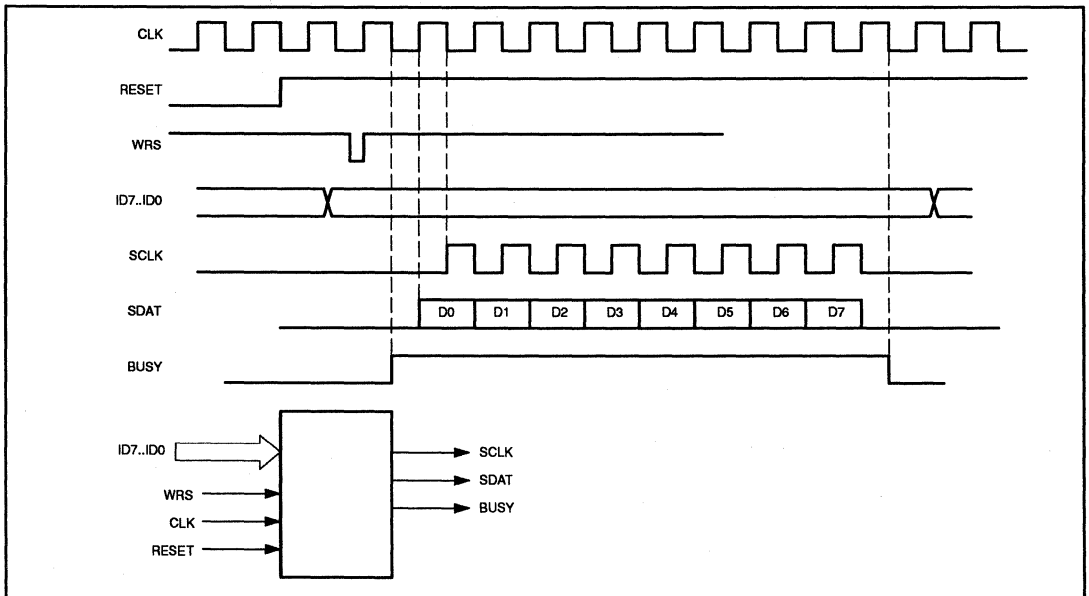


Figure 1. Desired Input and Output Waveforms

High speed 8-bit parallel to serial converter AN045

So far, we have a counter and multiplexer to serialize the data. The process of serialization begins with an asynchronous pulse on the write strobe input (WRS). It is therefore necessary to construct a latch to capture the pulse and then use two registers to synchronize the signal to the input clock. Figure 2 shows the desired operation of two intermediate signals Z and Z1. An extremely simple latch can be made with the equation: "Z = /WRS + Z". Once set with WRS low, it could never be reset. An additional signal named GATE, will be used as an extra term in the latch to reset it. From the waveforms of Figure 2, a table of the three signals may be constructed.

WRS	GATE	Z	Z+
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

	WRS,GATE 00	01	11	10
Z 0	1	1	0	0
1	1	1	0	1

Z+ is the "next state" or what value output Z should be, given the current inputs and the current state of Z. From the table, a Karnough map may be constructed (shown above) and the equation "Z=/WRS+Z*/GATE;" derived.

We have signal Z, which latches the input strobe, but we need to synchronize it to the input clock. That can be done with flip-flop Z1 and the following flip-flop, GATE, described later. For Z1, the equation is simply: "Z1.D=Z;" Z1 is clocked by the rising edge of CLK. Z1's output will go high upon the rising edge of CLK and Z high. It will go low upon a rising edge of CLK and Z low.

According to the original waveforms of Figure 1, a signal named BUSY is required to occur after the falling edge of CLK following a detected strobe (WRS). The internal D-type flip-flops of a 22V10 can only be clocked on the rising edge of the clock, so one of the 22V10's internal flip-flops cannot be used. However a Boolean equation may be used to describe this signal. The times and conditions when this signal is to be high will be noted from Figure 2 and a Boolean expression generated. From Figure 2, at time T2, BUSY should go high. Therefore, one term of the equation for BUSY will be: "Z*Z1*/CLK". When both Z and Z1 are high and CLK is low, then BUSY will go high. This product term will keep BUSY high until time T3.

At time T3, BUSY should remain high and adding a product term of "BUSY*Z1" can keep it high until time T5. This product term actually becomes active long before time T3 arrives, so there will be no glitching of the output. Adding yet another product term of "BUSY*GATE" will keep BUSY high from just after time T3, through time T5, until time T19. Finally, one last product term of "BUSY*CLK", keeps it high until the falling edge of the clock. The combined equation for BUSY is shown in Figure 3 lines 38 through 41.

The last signal to be described is GATE. It is used to control the gating of the inverted clock output SCLK, and also control the already described signals BUSY and Z. GATE can use one of the flip-flops inside the 22V10, as it should only switch after the rising edge of the input clock. It goes high after the first rising edge of CLK after BUSY goes high. Therefore, one of inputs to the GATE flip-flop has to be BUSY. GATE should go low after COUNT reaches seven, so the equation can be "GATE.D = BUSY * /(COUNT==7H);" The input to the GATE flip-flop will be high when BUSY goes high and COUNT is not equal to 7.

Signal GATE was also added to each of the terms in the multiplexer equation and it was added as a term in the counter equation. It was added to the multiplexer so SDAT would be low unless actual data was being sent . It was added to the counter so the counter would only count when GATE was high. This design used nine of the ten possible 22V10 outputs. The input RESET was added to many of the equations to force a proper initialization of the signals. From here it is just a matter of typing the equations into SNAP, running a simulation, and generating a JEDEC file for downloading to a device programmer. Figure 4 shows the SNAP simulation results and Figure 5 shows the associated simulation control language (SCL) file.

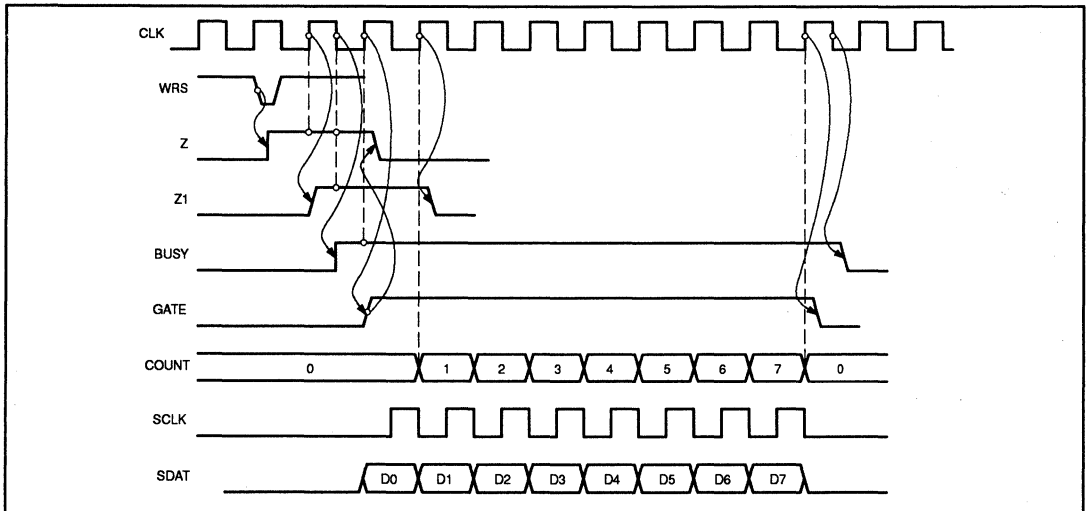
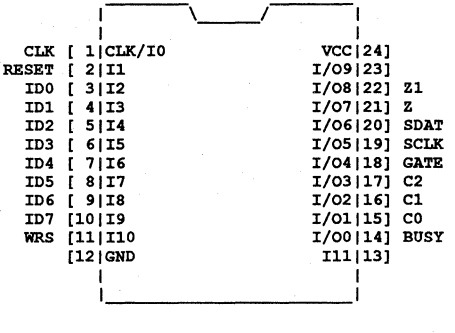


Figure 2. Waveform Timing Relationships

```
 1 |
 2 |"-------------------------------------------------"
 3 |"  High Speed 8-bit Parallel to Serial Converter  "
 4 |"-------------------------------------------------"
 5 |
 6 |@PINLIST
 7 |CLK              I;
 8 |ID[0..7]         I;
 9 |RESET            I;  "active low"
10 |WRS              I;
11 |
12 |BUSY             O;
13 |SCLK             O;
14 |SDAT             O;
15 |C[0..2]          O;
16 |GATE             O;
17 |Z                O;
18 |Z1               O;
19 |
20 |@GROUPS
21 |COUNT=[C2,C1,C0];
22 |
23 |@TRUTHTABLE
24 |@LOGIC EQUATIONS
25 |
26 |    "write strobe latch"
27 |
28 |Z = /WRS*reset+Z*/GATE*reset;
29 |
30 |    "first flip-flop to sychronize WRS to CLK"
31 |
32 |Z1.D   = Z;
33 |Z1.CLK = CLK;
34 |Z1.RST = reset;
35 |
36 |    "busy flag"
37 |
38 |BUSY = Z*Z1*/CLK*reset
39 |     + BUSY*Z1*reset
40 |     + BUSY*GATE*reset
41 |     + BUSY*CLK*reset;
42 |
43 |    "gate for control and 2nd synchronizing flip-flop"
44 |
45 |GATE.D   = BUSY*/(COUNT==7H);
46 |GATE.CLK = CLK;
47 |GATE.RST = reset;
48 |
49 |
50 |     "output clock"
51 |
52 |SCLK = /CLK*GATE;
53 |
54 |     "3-bit up counter"
55 |
56 |COUNT.D   = GATE==1 * COUNT#1H;   "count only when GATE is high"
57 |COUNT.CLK = CLK;
58 |COUNT.RST = reset;
59 |
60 |    "Multiplexer Equations"
61 |
62 |SDAT = ID0*(COUNT==0H)*GATE*reset  "if GATE is low then output a low"
63 |     + ID1*(COUNT==1H)*GATE*reset
64 |     + ID2*(COUNT==2H)*GATE*reset
65 |     + ID3*(COUNT==3H)*GATE*reset
66 |     + ID4*(COUNT==4H)*GATE*reset
67 |     + ID5*(COUNT==5H)*GATE*reset
68 |     + ID6*(COUNT==6H)*GATE*reset
69 |     + ID7*(COUNT==7H)*GATE*reset;
70 |
71 |@INPUT VECTORS
72 |@OUTPUT VECTORS
73 |@STATE VECTORS
74 |@TRANSITIONS
75 |
```



NOTE: Line numbers are for reference only, they are NOT part of the design file.

Figure 3. SNAP Listing and Pin File

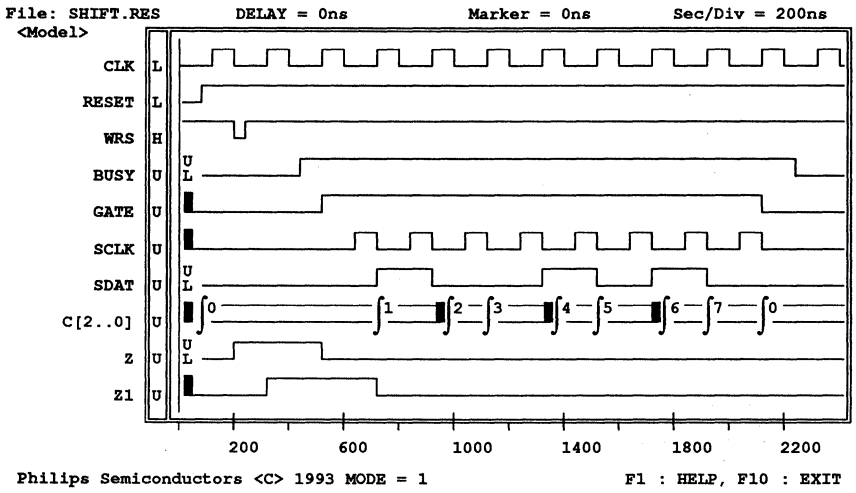


Figure 4. SNAP Simulation Waveforms

```
*****************************************************
*     Output of Waveform Version 1.90               *
* Date: 04/21/93                  Time: 16:14:48 *
*****************************************************
*                                                   *
* Input File Name   : SHIFT.SCL                     *
* Rule File Name    : Scl Rule                      *
* Output File Name : SHIFT.SCL                      *
*                                                   *
*****************************************************
 P ID0, ID1, ID2, ID3, ID4, ID5, ID6, ID7, CLK, RESET, WRS, BUSY,
#   GATE, SCLK, SDAT, C[2..0], Z, Z1
 PCO
 S 0 (5890) ID0
 S 1 (5890) ID1
 S 0 (5890) ID2
 S 0 (5890) ID3
 S 1 (5890) ID4
 S 0 (5890) ID5
 S 1 (9200) ID6
 S 0 (14290) ID7
 S 0 (100, 200, ETC) CLK
 S 0 (80, 5600, 6000) RESET
 S 1 (180, 240, 4000, 4200) WRS
 SU time = 14290
 F
```

Figure 5. SNAP Simulation SCL File

A Metastability Primer

AN219

INTRODUCTION

When using a latch or flip-flop in normal circumstances (i.e., when the devices set-up and hold times are not being violated) the outputs will respond to a latch enable or clock pulse within some specified time. These are the propagation delays found in the data sheets. If, however, the set-up and hold times are violated so that the data input is not a clear one or zero, there is a finite chance that the flip-flop will not immediately latch a high or low but get caught half way in between. This is the metastable state and it is manifested in a bi-stable device by the outputs glitching, going into an undefined state somewhere between a high and low, oscillating, or by the output transition being delayed for an indeterminable time.

Once the flip-flop has entered the metastable state, the probability that it will still be metastable some time later has been shown to be an exponentially decreasing function. Because of this property, a designer can simply wait for some added time after the specified propagation delay before sampling the flip-flop output so that he can be assured that the likelihood of metastable failure is remote enough to be tolerable. On the other hand one consequence of this is that there is some probability (albeit vanishingly small) that the device will remain in a metastable state forever. The designer needs to know the characteristics of metastability so that he can determine how long he must wait to achieve his design goals.

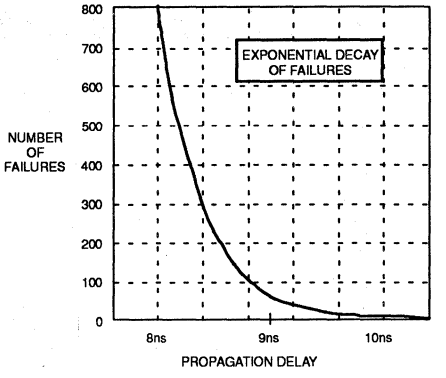


Figure 1.

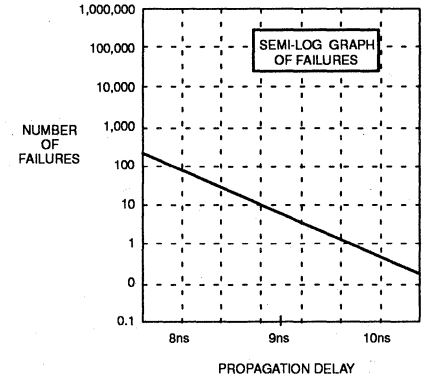


Figure 2.

THE CHARACTERISTICS OF METASTABILITY

In order to define the metastability characteristics of a device three things must be known: first, what is the likelihood that the device will enter a metastable state? This propensity is defined by the parameter T_0. Second, once the device is in a metastable state how long would it be expected to remain in that state? This parameter is tau (τ) and is simply the exponential time constant of the decay rate of the metastability. It is sometimes called the metastability time constant. The final parameter is the measured propagation delay of the device. Commonly, the typical propagation delays found in the data book are used for this and it is designated h in the equations (although most designers are familiar with this value as Tpd). Now lets see how tau and T_0 are determined by measurements.

A TEST METHOD

Suppose we wanted to measure the metastability characteristics of a fictitious edge-triggered D-type flip-flop and we had a test system that would count each time the flip-flop is found in a metastable state at some time after a clocking edge. The first thing we would like to know about the flip-flop would be the h or typical propagation delay. We could measure the delay or look it up in the data book (of course, measuring the actual delay would allow more precise results). This fictitious flip-flop has an h of 7ns. In this test we decide to use a clock frequency of 10MHz. This frequency is primarily a function of the test systems ability to assimilate the information. The data will run at 5MHz asynchronously to the clock and with a varying period. This frequency was chosen because at two transitions per cycle the data signal produces 10 million points each second where it is possible for the flip-flop to go into a metastable state, an average of one point for each clock pulse. An important point about the characteristic of the data signal in relation to the clock is that the data transitions must have an equal probability of occurring anywhere within the clock period or the results could be skewed. In other words, we need to have a uniform distribution of random data transitions (high and low) relative to the clocking edge.

The first measurement we take is to determine the number of times the device is still in a metastable state 8ns after the clock edge. With this device there are 792 failures after 1 billion clock cycles. Changing the time to 9ns we measure 65 failures after another 1

billion cycles. Because metastability resolves as an exponentially decaying function the two points define the exponential curve and they can be plotted as shown in Figure 1. An equivalent plot can be made using a semilog scale as in Figure 2. The slope of the line drawn through the two points represents tau. With these two points the tau can be determined by equation (1):

$$\tau = \frac{t_2 - t_1}{ln(N_1 \ / \ N_2)} \qquad (1)$$

where N_1 and N_2 are the number of failures at times t_1 and t_2, respectively.

Working through the numbers gives us a tau of 0.40ns. Tau of this order is representative of the FAST line of flip-flops.

Earlier we stated that T_O is an indicator of the likelihood that the device will enter a metastable state. Now we will attempt to explain it. At 9ns after the clock we observed 65 failures in 1 billion clock cycles. Since the data transits on average once per clock cycle and the period of this clock is 100ns, from equation (2) we can say that there appears to be an aperture about 0.0065 picoseconds wide at the input of the device that allows metastability to occur for 9 or more nanoseconds. Another way of explaining the same thing would be to suppose that if 1 billion data transitions were uniformly and randomly distributed over a clock period of 100ns: you would expect 65 of these transitions to cause the outputs to go into a metastable state and remain there for at least 9ns.

$$T_9 = \frac{N_9 \ P_C}{N_{C9}} \qquad (2)$$

Where N_{C9} is the number of clocking events at 9ns (in this instance, 1 billion), Pc is the period of the clock, and N_9 is the number of failures recorded at 9ns.

By the same reasoning the window at 8ns appears to be 0.0792 picoseconds wide. It seems to have grown because there are, of course, more failures after 8ns than after 9ns. This aperture has been normalized by researchers to indicate the effective size of the aperture at the clock edge, or time zero. Unfortunately the normalization process tends to obscure the interpretation of T_O. T_O can be calculated using equation (3). Figure 3 is an extension of Figure 2 and shows the relationship of T_O, h, and tau.

$$T_O = T_8 e^{\left(\frac{8ns}{\tau}\right)} \qquad (3)$$

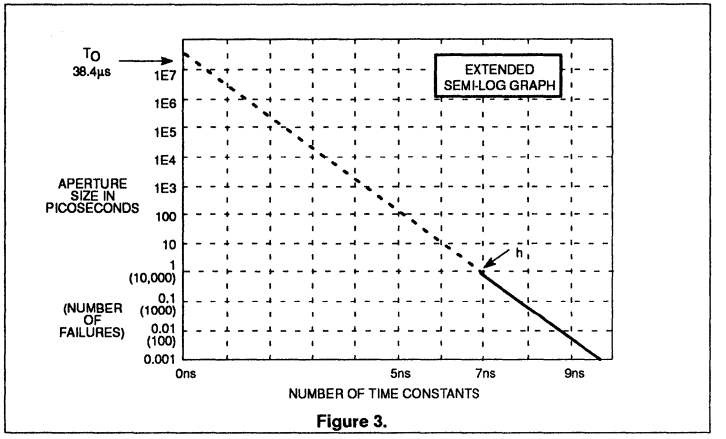


Figure 3.

or equivalently,

$$T_O = T_9 e^{\left(\frac{9ns}{\tau}\right)}$$

In this case To is 38.4µs and this value is again typical of the FAST line of products.

Figure 3 is an extension of Figure 2 and gives a graphic indication of T_O. The number of failures plots on the same scale as the aperture size but the number of failures is dependent on the number of clock cycles used in the test (we always used 1 billion in this paper) and the ratio of data transitions to clock pulses (1:1 in this paper). On the other hand, the aperture size is independent of these things.

MTBF

Having determined the T_O and tau of the flip-flop, calculating the mean time between failures (MTBF) is simple. Suppose a designer wants to use the flip-flop for synchronizing asynchronous data that is arriving at 10MHz, he has a clock frequency of 25MHz, and has decided that he would like to sample the output of the flip-flop 15ns after the clock edge. He simply plugs his numbers into equation (4) .

$$MTBF = e^{\frac{\left(\frac{t'}{\tau}\right)}{T_O \ f_C \ f_i}} \qquad (4)$$

In this formula f_C is the frequency of the clock, f_i is the average input event frequency, and t' is the time after the clock pulse that the output is sampled (of course t' > h). In this situation the f_i will be twice the data frequency because input events consist of both low and high data transitions. For the numbers above the MTBF is one million seconds or about one failure every 11.6 days. If the designer would have tried to sample the data after only 10ns the MTBF would have been 3.8 seconds.

Metastability literature can be very confusing because several companies use different nomenclature and often the fundamental parameters are obscured by scale factors, so it is important that the user understand MTBF. Lets try a thought experiment to determine the correct MTBF formula. We know the size of the aperture at 8ns so we need to know how often that window will occur. This is supplied by the clock period. This gives a ratio of window size to clock period and gives us the likelihood of a transition within the clock period causing a metastable state that lasts beyond the 8ns point. Now we need to know the number of input events per clock period to determine the MTBF at 8ns. This is supplied by the average input event period and produces the equation below where P_C and P_i are the periods of the clock and input events, respectively.

$$MTBF = \frac{1}{T_8 \frac{1}{P_C} \frac{1}{P_i}} = \frac{1}{T_8 \ f_C \ f_i} \qquad (5)$$

This gives the MTBF for 8ns, but how can the formula be developed to handle other times? It has been stated in this paper that the rate of decay of metastable events is an exponential function with a time constant of tau. Using this information gives the equation below where t' is the time after the clock pulse that the output is sampled.

$$MTBF = \frac{e^{\left(\frac{t'-8ns}{\tau}\right)}}{T_8\, fc\, fi} = \frac{e^{\left(\frac{t'}{\tau}\right)}}{T_8\, e^{\left(\frac{8ns}{\tau}\right)}\, fc\, fi} \quad (6)$$

$$= \frac{e^{\left(\frac{t'}{\tau}\right)}}{T_O\, fc\, fi}$$

A point should be made here about MTBF. This is the mean time between failures and as such does not indicate the average time between failures. In fact, in this situation, the MTBF is the time before which there is a 63.2% probability that a failure would have occurred. Suppose a device has an MTBF of one million seconds like the example above; because the MTBF is an exponential function there is a 9.5% probability that a failure will occur in the first 1.16 days of operation. This might cause the user to feel that the device is failing more than expected. The user would find that 50% of his failures would occur within 8 days. Figure 4 gives a visual interpretation of this idea: time constant one represents one million seconds in this case.

RECENT DEVELOPMENTS

The quest for better metastability characteristics in flip-flops has recently resulted in the development of flip-flops with taus significantly less than 0.40ns. Perhaps the most notable of these is the Philips Semiconductors 74F50XXX series with typical taus of 135ps. The specifications of these new products can cause confusion among the uninitiated because the typical T_O on these devices is 9.8 million seconds or about 113 days. This is an example of how the normalization process obscures the interpretation of T_O. In the newest products the taus have decreased faster than the normal propagation delays primarily due to speed limitations of the outputs.

Using the example above and calculating T_7 from equation (3) we see that the window at h is 0.965ps. Now lets assume that we have a device with the same size window (0.965ps) at h and an h of 7ns. The difference between this device and the previous example is that this device has a tau of 150ps. Clearly, if the device has the same h and the same size of window at h but a smaller tau, the device is better. But lets calculate the T_O.

$$T_O = T_7\, e^{\left(\frac{7ns}{\tau}\right)}$$

T_O = 178 million seconds!

Comparing the T_O of any two devices does not show which device is superior. However, one can expect that the device with the lower tau is superior in all but the most peculiar circumstances.

SUMMARY

This paper is intended to introduce the reader to the terms he will be dealing with regarding metastability and it is hoped that this introduction will help him to digest the more in-depth papers that he will be reading. Philips Semiconductors uses the parameters described by Thomas Chaney of Washington University in St. Louis, Missouri because they are fundamental and the better metastability papers generally use these parameters. For further reading on the subject, the article "Metastable behavior in digital systems", by Lindsay Kleeman and Antonio Cantoni published in *IEEE Design & Test of Computers* in December of 1987, is recommended.

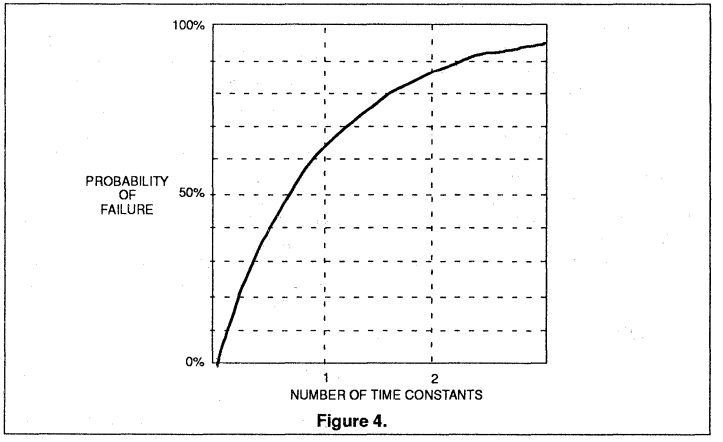


Figure 4.

PLA Devices

INTRODUCTION

Philips Semiconductors PLAs are particularly useful in the design of wide address decoders and random logic replacement. The primary advantage Philips Semiconductors brings to these applications with their PLA devices is product term sharing, which is made possible via the two programmable arrays graphically shown in Figure 1. The familiar PAL® architecture supports a programmable AND array, followed by a fixed OR array. Better than 90% of the PAL devices that are available today are limited to 8 input wide OR gates. When pursuing a solution to a complex address decoding scheme, this restriction is prohibitive. The Philips Semiconductors PLA devices support 100% connection of all product terms to one or more OR gates. Once a term is created, it can be shared with any or all of the output functions. No duplication of resources is incurred. The popular PLXX153 family support 32-input wide OR gates which are ideal for memory or I/O decoders. The addition of programmable output polarity also enhances design efficiency and logic minimization.

The two programmable array concept dominates the Philips PLD product line. With the exception of the PAL-type devices which have been geared for ultimate performance, all Philips PLDs have been architected with efficient and flexible PLA structures. With the largest breadth programmable product line in the industry, Philips believes the designer can truly fill his requirements from the several product lines – PLA, PAL and PLS. Two combinatorial logic PLA device descriptions follow. For information on PLA devices with registers please refer to the sequencer section of this manual.

PHILIPS SEMICONDUCTORS PLUS153

Figure 2 depicts the Philips PLUS153. This bipolar PLA is pin and functionally equivalent to the Philips PLS153, however is available with a maximum propagation delay time of 10ns from input to output.

The PLUS153 has eight dedicated inputs and 10 bidirectional pins. The bidirectional pins may be adapted to suite the user's specific needs. 20-pin DIP or PLCC packages are available.

The output structure of the PLUS153 includes programmable polarity control on each output. Either active HIGH (non-inverting) or active LOW (inverting) outputs are configurable via the EX-OR gate associated with each I/O. Individual 3-State control of the I/O is also supported with the ten direction control AND terms (D1-D9).

Other benefits to the PLUS153 include full pin compatibility with most 20-pin combinational PAL parts. The natural product term sharing capabilities of the PLA architecture yield complete freedom of configuration should the engineer implement a particularly creative decode configuration.

PHILIPS SEMICONDUCTORS PLUS173

Figure 3 depicts the Philips Semiconductors PLUS173. This bipolar PLA is functionally equivalent to the Philips PLS173. The 24-pin PLUS173 has four more inputs pins than the PLUS153. The user may adapt the bidirectional pins to suit particular decoding needs, but the propagation delay time is still no more than 10ns from stabilized input to stable output for a PLUS173-10 device.

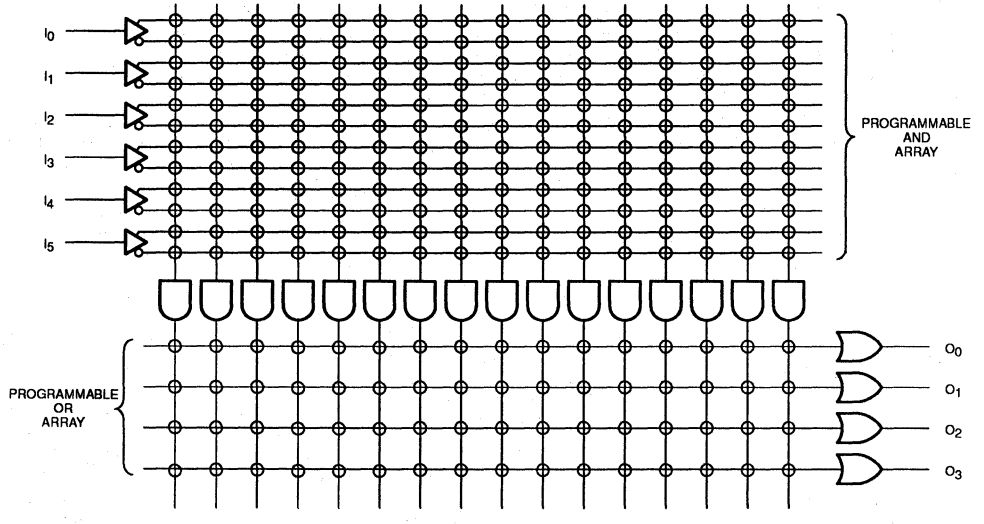
By having more inputs than the 153 part, the 173 can either resolve more input lines or generate more outputs functions for the same number of inputs. Distinct 3-State control over each output may be useful for controlling chip enables where contention (i.e., multiple access) may exist.

For speed and input width, the PLUS173 is probably the best single PLS available today for both memory and I/O decoding. Combining the 10ns T_{PD} with the distinguishable range of 12 to 21 inputs, the designer can easily decode say 16 input addresses as well as read/write qualifiers or encoded status signals. Output polarity control (Active-High or Active-Low) is achieved by programming the Exclusive-OR gate associated with each output.

The flexibility achieved with a PLA structure can be quickly appreciated by the designer who has experienced the frustration of the dedicated "OR" structures in PAL ICs. Currently, the only time penalty for the freedom granted by a PLA is a few nanoseconds!

® PAL is a registered trademark of Advanced Micro Devices

PLA Devices



O denotes programmable connection

Figure 1. PLA Architecture

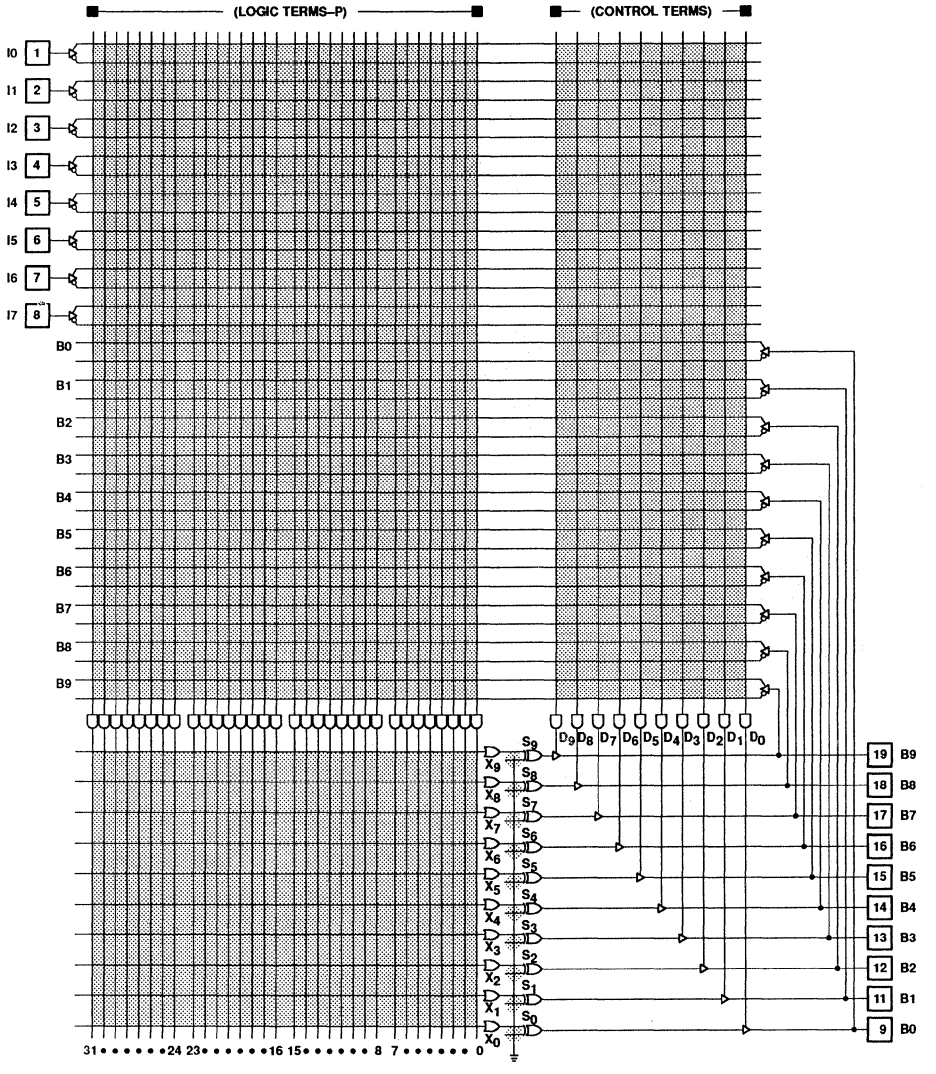


Figure 2. PLUS153

PLA Devices

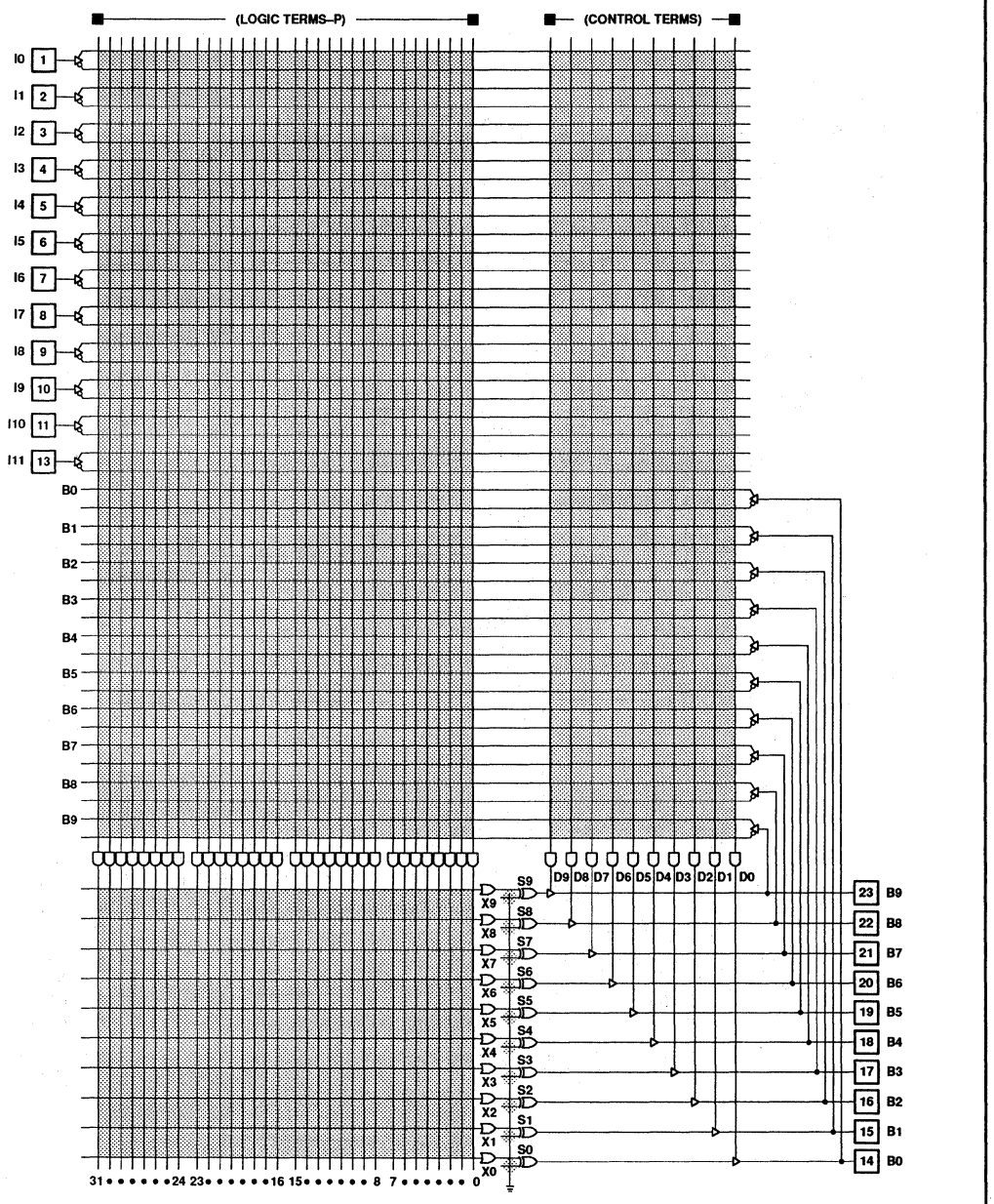


Figure 3. PLUS173

Quick PLA

AN046

INTRODUCTION

The PLUS153-10 is available in a 20-pin DIP or 20-pin PLCC package. The PLUS173-10 is available in a 24-pin DIP or 28-pin PLCC package. Both parts have t_{PD} no greater than 10ns.

Both parts provide 32 wide input product terms, whose outputs may be tied to the inputs of the sum terms (OR gates) below. There are no restrictions on this interconnect — any or all product terms may feed any or all sum terms. Thus each OR term can accept from 1 to 32 inputs without leaving the chip for a signal "wrap around". All ten outputs are bi-directional, so they may be traded off as inputs are used. Finally, each output may be polarity configured (exclusive-OR fused) and each is independently 3-Stateable from a separate product term (each) which is identical to the rest.

Although slightly slower (from pin to pin) than 7.5ns 20L8 structures, the following example demonstrates a simple case of how a 10ns PLA can be faster than a 7.5ns PAL.

Example 1: Glue Collection

This first example is an illustration of compressing glue logic. Figure 1 shows a piece of logic which performs one of two operations on two 8-bit numbers. These may come from different registers in a system, or be from two halves of a 16-bit bus. The goal is to perform the input operations (compare the bytes in one mode or multiplex one bit out in the other) in 10ns. Using MSI parts, this could have been done except there is no 16 to 1 MUX available in the 74F device series. There is a 74150 device available, but it has a propagation delay of 17ns. So this will not work. Figure 1 shows the solution using the 7.5ns PAL devices. Unfortunately, because architecture provides only seven product terms per sum term (16L8-7) multiple signal passes are required. This results in a solution needing over 20ns. It might be conjectured that a 15ns 22V10 could make it with 16 product terms on some outputs, but doing the MUX would only provide the output at "point 1" in 15ns. Additional time is needed to make the final out signal. A 10ns 22V10 could not make spec, with an additional 74F32 adding 4ns. Figure 3 shows the preferred solution — a single PLUS173 generating the final function in 10ns. Figures 4 and 5 show the pinout and SNAP equations for this solution.

Example 2: Cache Update Inhibit

Key to modern microsystem design has been simple, fast RISC processors with quick cache and single cycle high performance operation. Unfortunately even using the new cache control chips, exception handling results in clumsy designs. This may be one of the reasons simple, direct-mapped caches have also become popular. Exception handling is often resolving transactions which occur with data items that are non-cacheable. This occurs in a number of ways — first, EPROMs, I/O devices and special state registers are not cacheable items, so they will never be put into a cache memory. What happens when a non-cacheable item is referenced? The cache controller will miss and begin to update the cache. The transaction must be terminated before it overlays an I/O device onto the least recently used cache address.

The way to deal with the transaction overlay problem is straightforward — recognize all non-cacheable transactions and intercept them before the controller cleans house. How big of a problem is this? Figure 6 shows what might be an average engineering workstation. Each device (disk controller, LAN controller, keyboard, printer, etc.) usually has several internal registers each occupying a unique address. With two disks, a LAN, modem and printer, a system could instantly exceed 16 distinct I/O registers. It is best to assume a large number. Enter the PLA — the PLUS173 — for such a system. Each product term can be scattered all over memory if needed and decodes summed into a single output signal generating a composite inhibit. This process takes less than 10ns for up to 32 devices. Using a 20L8-7 requires trading off resolution (number of address bits resolved) and feeding through the chip multiple times, expanding to 13 devices in two passes (at 15ns for a 7.5ns device). Using a PLA keeps the RISC design very clean and fast.

Example 3: Interfacing Mixed Memory Types

Other sections of a microprocessor system can use the summation of a large number of decoded terms. For instance, the interrupt request, DMA request and the cycle extension WAIT line are contenders for a large number of decoded and summed inputs. Some are asserted low and some high so polarity control is vital. Some require a 3-State or open collector resistive pull-up, so the PLA enable fits well. These are situations where attention signals come into the processor.

It is not always necessary for the CPU to operate at full speed. Operating the CPU at a slower speed brings about a more economical and compact system. This is due to higher costs associated with fast memory and greater board area for wide memory configurations.

Some software routines where slower performance may be acceptable include power up initialization, diagnostic routines, or some exception routines. When speed is critical, an 8-bit bus is the most economical and compact because of readily available byte wide PROMs and RAMs. The 68030 is easily interfaced to 8-, 16- or 32-bit ports because it dynamically interprets the port size during each bus cycle. Figure 7 shows an example of interfacing both a slow 200ns 8-bit EPROM and a fast 35ns 32-bit RAM to a 68030. A PLUS173-10 was chosen for its high speed and large number of inputs and outputs. The EPROM occupies memory space 0–32K while the RAM occupies addresses 64–128K. Note that because not all of the upper memory address bits were decoded, the memory arrays will also appear at other addresses.

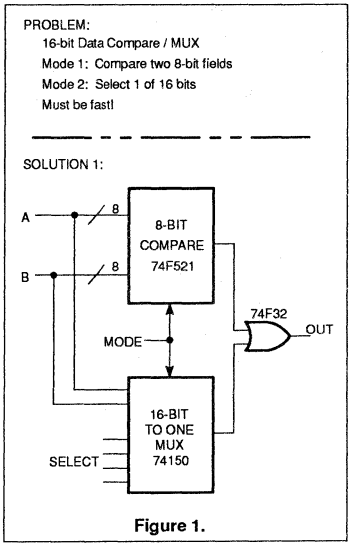
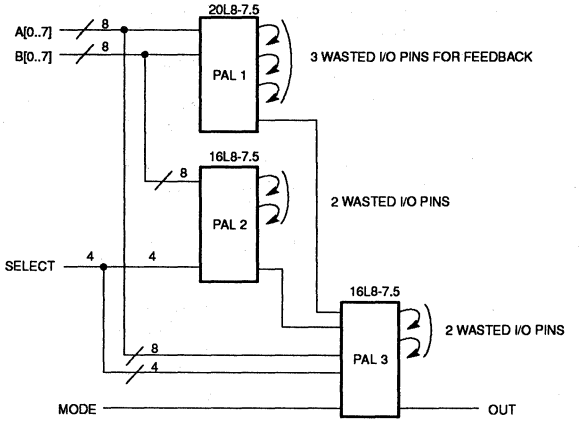


Figure 1.

Quick PLA

SOLUTION 2: MAKE IT OUT OF PALs



3) 7.5 PALs
1) 20L8-7.5
2) 16L8-7.5

COMPARE FUNCTION:

t_{PD} = 7.5 (1st PASS — #1)
= 7.5 (2nd PASS — #1)
= 7.5 (3rd PASS — #3)
= 22.5ns

MUX FUNCTION:

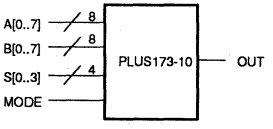
t_{PD} = 7.5 (1st PASS — #2 or 3)
= 7.5 (2nd PASS — #2 or 3)
= 7.5 (3rd PASS — #3)
= 22.5ns

GATE EFFICIENCY:
30 PALs
40 GATES USED
152 GATES WASTED

RESULT:
NOT FAST ENOUGH; SLOWER THAN 74F SOLUTION

Figure 2.

SOLUTION 3: MAKE IT OUT OF A PLUS173D



t_{PD} = 10ns
1 PACKAGE

COMPARE FUNCTION:
16 PRODUCT TERMS

MUX FUNCTION:
16 PRODUCT TERMS

INVERTERS:
INCLUDED

ORS:
INCLUDED

GATE EFFICIENCY:
1 PLA
32 GATES USED
NO GATES WASTED

RESULT:
THE BEST SOLUTION; ONE PACKAGE

Figure 3.

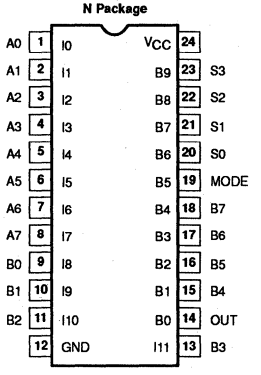


Figure 4. Comparator/MUX Pin Program

Quick PLA

```
@PINLIST
a[0..7]  i;
b[0..7]  i;
s[0..3]  i;
MODE  i;
OUT   o;

@GROUPS
sel     =  s[0..3];

@TRUTHTABLE
@LOGIC EQUATIONS

comp    =  a0 * b0 /mode
        +  /a0 * b0 /mode
        +  a1 * b1 /mode
        +  /a1 * b1 /mode
        +  a2 * b2 /mode
        +  /a2 * b2 /mode
        +  a3 * b3 /mode
        +  /a3 * b3 /mode
        =  a4 * b4 /mode
        +  /a4 * b4 /mode
        +  a5 * b5 /mode
        +  /a5 * b5 /mode
        +  a6 * b6 /mode
        +  /a6 * b6 /mode
        +  a7 * b7 /mode
        +  /a7 * b7 /mode;

mux     =  a0 * (sel == 0h) * mode
        +  a1 * (sel == 1h) *
mode    +  a2 * (sel == 2h) * mode
        +  a3 * (sel == 3h) * mode
        +  a4 * (sel == 4h) * mode
        +  a5 * (sel == 5h) * mode
        +  a6 * (sel == 6h) * mode
        +  a7 * (sel == 7h) * mode
        =  b0 * (sel == 8h) * mode
        +  b1 * (sel == 9h) * mode
        +  b2 * (sel == Ah) * mode
        +  b3 * (sel == Bh) * mode
        +  b4 * (sel == Ch) * mode
        +  b5 * (sel == Dh) * mode
        +  b6 * (sel == Eh) * mode
        +  b7 * (sel == Fh) * mode;

out = mux + comp;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 5. SNAP Equation Listing

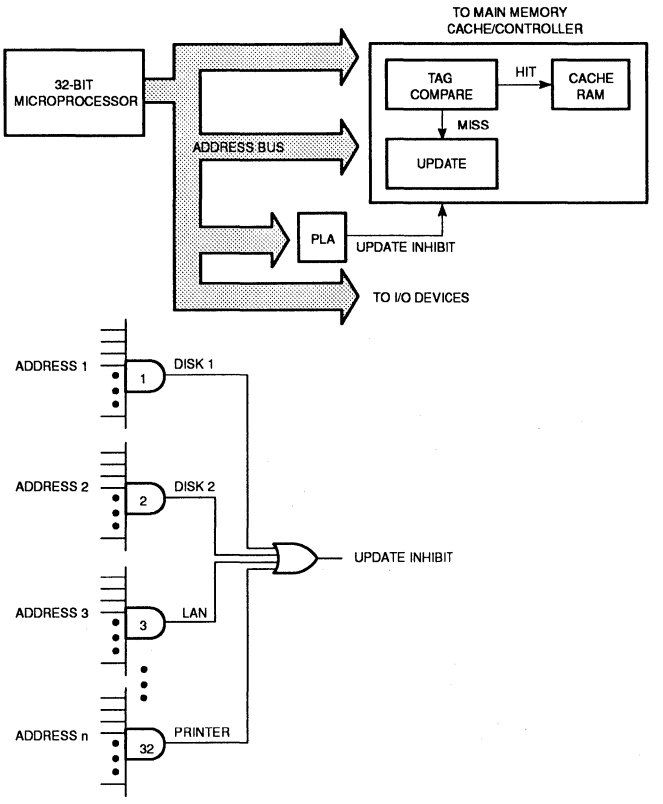


Figure 6. Cache Update Inhibit Using a PLA

Quick PLA

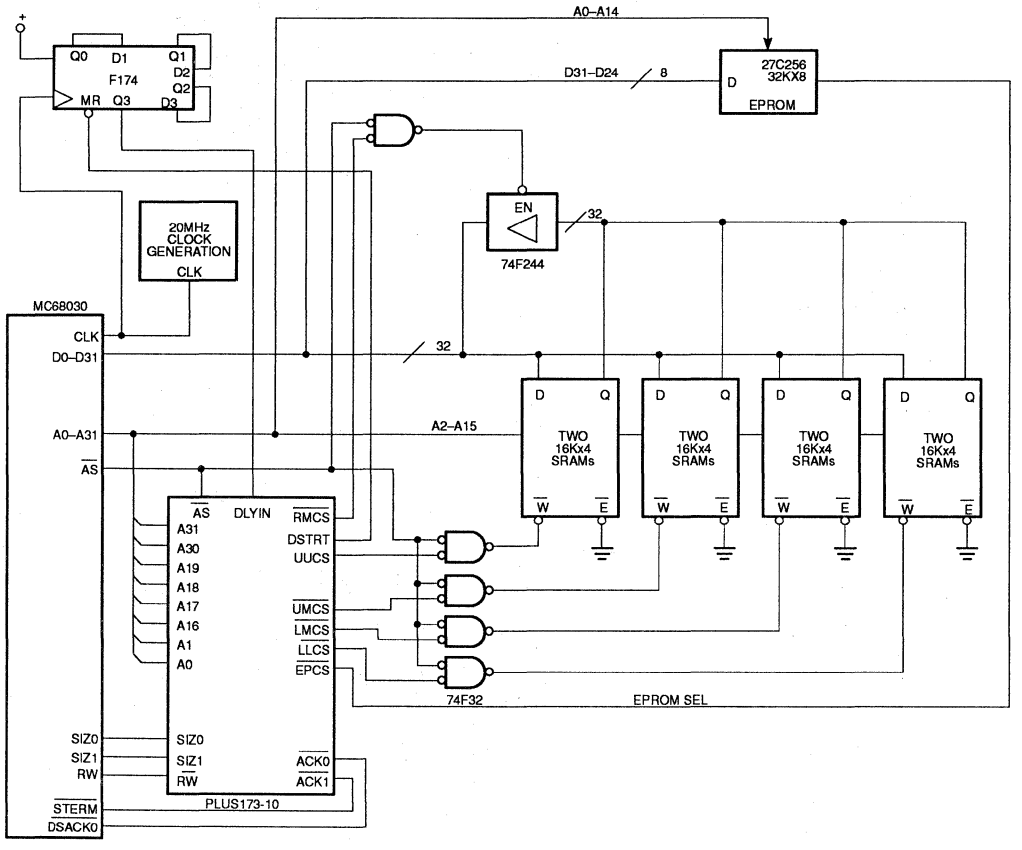


Figure 7. Example Interfacing Mixed Memory Types

```
DECODER FOR INTERFACING SRAMa AND EPROMs TO AN MC68030.  THIS DESIGN IS FOR A PLUS173 DEVICE

@PINLIST
dlyin     i;
nas       i;
a[31..30]   i;
a[19..16]   i;
a[1..0]   i;
siz0      i;
siz1      i;
rw        i;
nrmcs     o;
dstrt     o;
nuucs     o;
numcs     o;
nlmcs     o;
nllcs     o;
nepcs     o;
nack[1..0]  o
;
@LOGIC EQUATIONS
 "EPROM enable"
 nepcs = / ( /a31 * /a30 * /a19 * / a18 * /a17 * / a16 * / nas;

 "start shift register during EPROM access"
 dstrt = / ( /a31 * /a30 * /a19 * / a18 * /a17 * / a16 * / nas;

 "DSACKO after 4 clock cycles for EPROM access"
 nack0 = / (dlyin);

 "immediate STERM upon RAM access"
 nack1 = / ( /a31 * /a30 * /a19 * /a18 * /a17 * /a16);

 "Byte select signals for RAM writes"
 nuucs   =  / ( /a0 * /a1 *          /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31);

 nuucs   =  / ( a0 * /a1 *           /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 * /siz0 *           /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 * /siz1 *           /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31);

 nlmcs   =  / ( /a0 * /a1 *          /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 * /siz0 * /siz1 *   /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 * siz0 * /siz1 *    /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 * a0 * /siz0 *      /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31);

 nllcs   =  / ( a0 * a1 *            /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +   a0 * siz0 * siz1 *      /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +       / siz0 * /siz1 *    /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31
         +  / a1 *         siz1 *    /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31);

 nrmcs   =  / (                      /rw * a16 * /a17 * /18 * /a19 * /a30 * /a31);
```

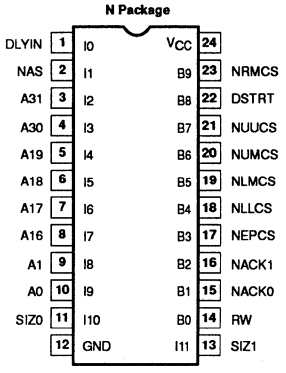


Figure 8. Equations for PLUS173 Shown in Figure 7

Latches and flip-flops with PLS153

AN014

DESCRIPTION

Using the simple AND, OR and INVERT logic functions of the PLS153, memory functions such as latches and edge-triggered flip-flops may be implemented with a relatively small part of the chip and without external wiring. In this application note, we will discuss the implementation of two R-S latches, a D-latch, an edge-triggered R-S flip-flop, an edge-triggered D flip-flop, and an edge-triggered JK flip-flop.

SIMPLE R-S LATCH

A simple R-S latch may be formed by cross-coupling two NAND functions together as shown in Figure 1.

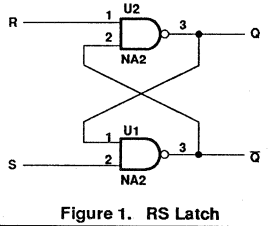


Figure 1. RS Latch

```
@pinlist
r    i;
s    i;
q    b;
qn   b;

@logic equations
qn = /(q*s);
q = /(r*qn);
qn · oe = 1;
q · oe = 1;
```

Figure 2. RS Latch SNAP Equations

ANOTHER SIMPLE R-S LATCH

Another way to implement a simple latch is shown in Figure 3, in which two NOR functions are cross-coupled to form a latch.

The SNAP equations are shown in Figure 4.

Since each AND-term of the PLS153 can accommodate up to 18 inputs (true or inverting inputs of eight from I_0 to I_7 and ten from B_0 to B_9), and each OR circuit can be connected to up to thirty-two AND-terms, we can add additional features such as those shown in Figure 5.

The programming of this design is left to the reader as an exercise.

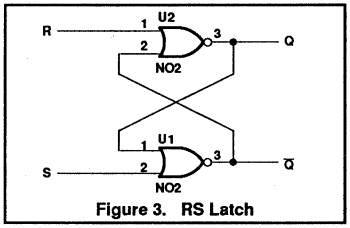


Figure 3. RS Latch

```
@pinlist
r    i;
s    i;
q    b;
qn   b;

@logic equations
qn = (/q*/s);
q = (/r*/qn);
qn · oe = 1;
q · oe = 1;
```

Figure 4. RS Latch SNAP Equations

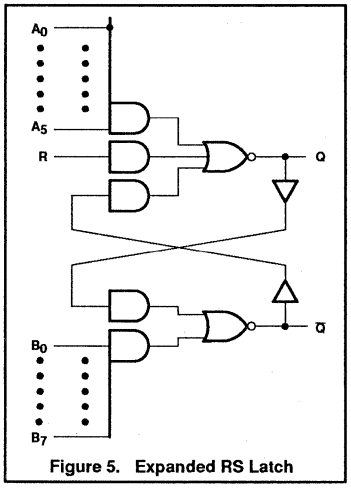


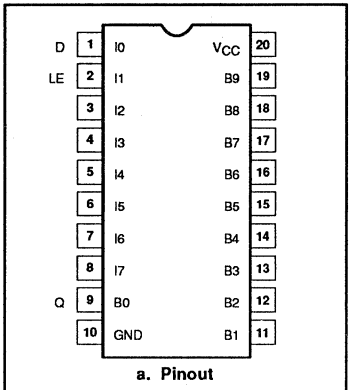
Figure 5. Expanded RS Latch

D-LATCH

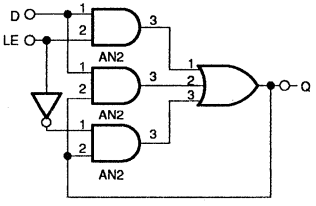
A simple D-latch can be constructed with a PLS153 as shown in Figure 6.

This circuit may be easily programmed into the PLS153 as shown in Figure 7. Note that according to the K Map of Table [1], there is a static hazard using only two gates, so the D * Q term is recommended.

This circuit may be expanded to have multiple D-latches using the same latch enable (LE).



a. Pinout



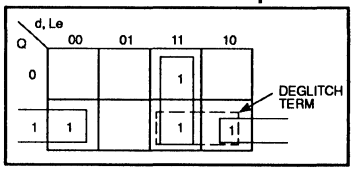
b. D-Latch Schematic

Figure 6.

```
@pinlist
d    i;
le   i;
q    o;

@logic equations
q = ((d * le)
  + (d * q)
  + (d * /le));
```

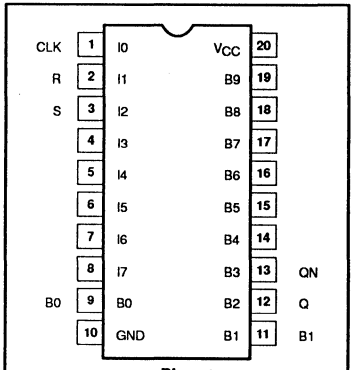
Figure 7. D– Latch SNAP Equations

Table 1. D-Latch K Map



R-S FLIP-FLOP

Two R-S latches may be combined to form a master-save flip-flop that is triggered at the rising-edge of the clock (or the falling-edge of the clock, if the designer so desires). Figure 9 shows a combination of two sets of cross-coupled NOR gates concatenated to form the flip-flop. The implementation of this circuit using SNAP equations is shown in Figure 8.



a. Pinlist

```
@pinlist
 clk  i;
 r    i;
 s    i;
 b0   b;
 b1   b;
 q    b;
 qn   b;

@logic equations
qn = / (q
     + (b1 * clk) );
 q = / ( (clk * b0)
     + qn);
b1 = / b0
     + (s * / clk);
b0 = / ( (r * /clk)
     + b1):

b0 . oe = 1;
b1 . oe = 1;
q . oe = 1;
qn . oe = 1;
```

Figure 8. SNAP Equations

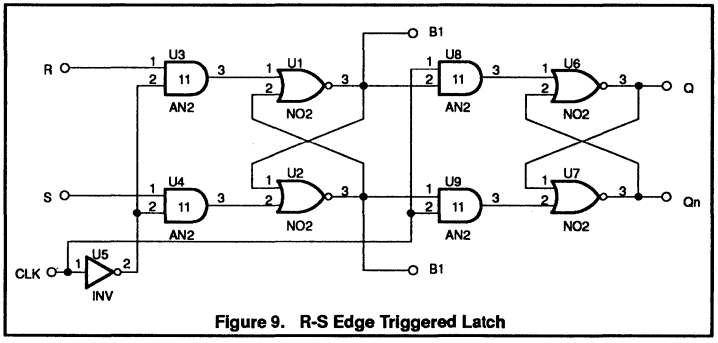


Figure 9. R-S Edge Triggered Latch

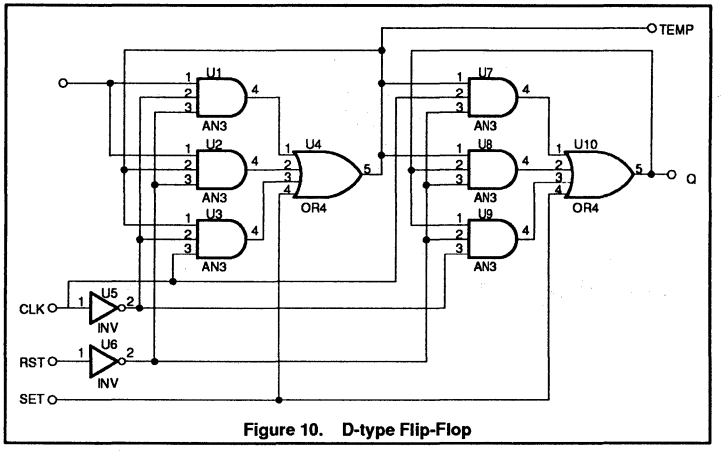
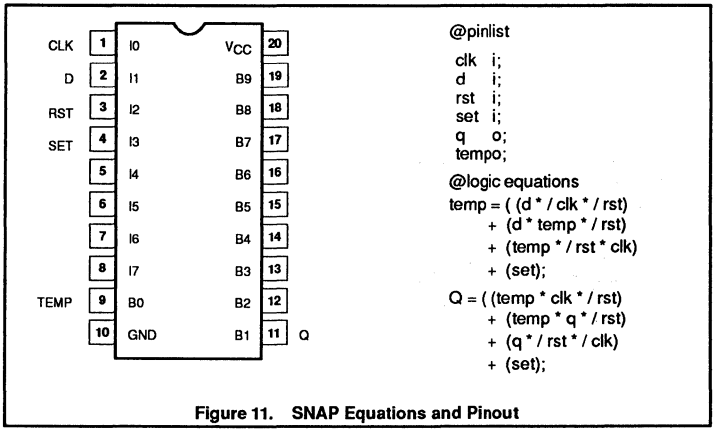


Figure 10. D-type Flip-Flop




```
@pinlist
 clk  i;
 d    i;
 rst  i;
 set  i;
 q    o;
 tempo;

@logic equations
temp = ( (d * / clk * / rst)
       + (d * temp * / rst)
       + (temp * / rst * clk)
       + (set);
Q = ( (temp * clk * / rst)
       + (temp * q * / rst)
       + (q * / rst * / clk)
       + (set);
```

Figure 11. SNAP Equations and Pinout

D FLIP-FLOP

An edge-triggered master-slave D flip-flop may be constructed with two D-latches in the manner shown in Figure 10.

A PLS153 may be programmed as shown in Figure 11 to implement the D flip-flop.

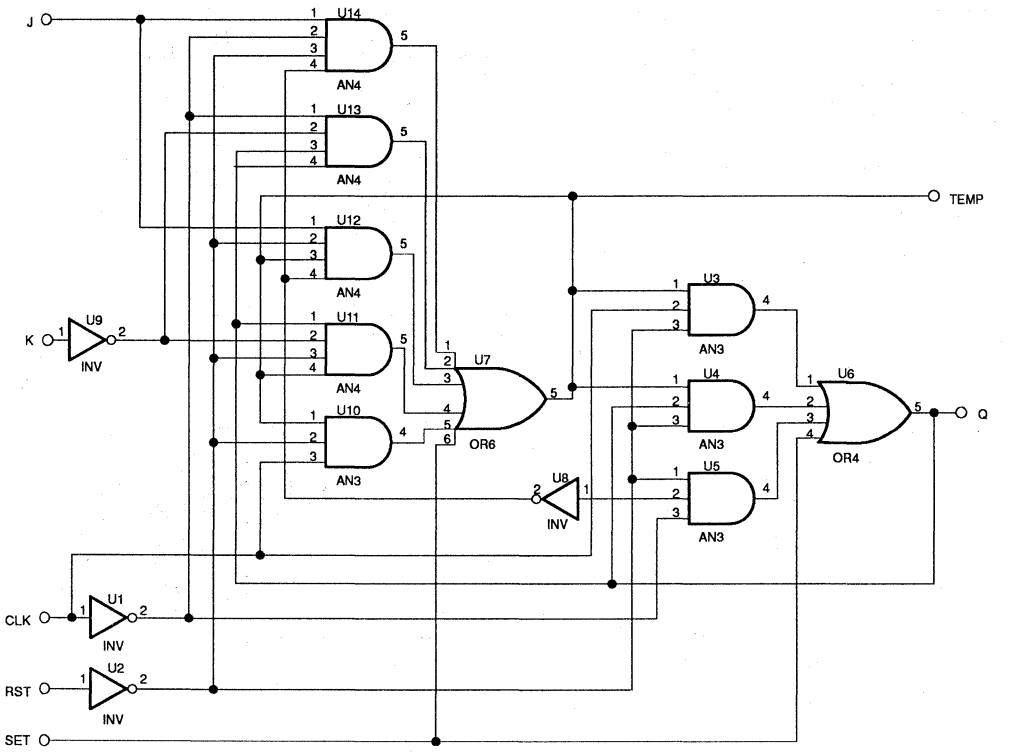
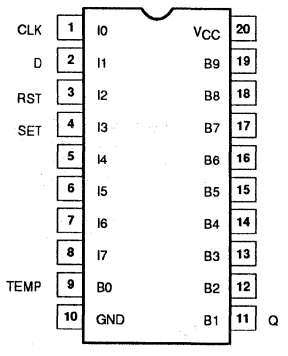


Figure 12. JK-type Flip-Flop




```
@pinlist
 clk   i;
 j     i;
 k     i;
 rst   i;
 set   i;
 q     o;
 tempo;

@logic equations
temp = ( j * / clk * / rst * /q)
      + (/clk * / k *q * / rst)
      + (j * / rst * temp * /q)
      + (temp * /rst * clk)
      + (set);
 q = ( (temp * clk * / rst)
      + (temp * q * / rst)
      + ( / rst  * q * / clk)
      + (set);
```

Figure 13. SNAP Equations and Pinout

JK FLIP-FLOP

An edge-triggered JK flip-flop schematic is shown in Figure 12. SNAP equations and pinout are shown in Figure 13.

PLS173 as a 10-bit comparator, 74LS460

AN024

DESCRIPTION

The PLS173 is a 24-pin PLA device which has 10 bidirectional outputs and 12 dedicated inputs. The output of the device is the sum of products of the inputs. The polarity of each output may be individually programmed as Active-High or Active-Low. A 10-bit comparator similar to the 74LS460 compares two 10-bit data inputs to establish if EQUIVALENCE or NOT EQUIVALENCE exists. The output has True and Complement comparison status outputs. The logic diagram of the comparator is shown in Figure 1.

The truth table is as shown in Table 1 where vectors a and b are 10-bit inputs to A9 to A0 and B9 to B0. If the input to A9-A0 is bit-to-bit equivalent to the input to B9-B0, the two input vectors are considered EQUIVALENT, and output EQ goes High and NE goes Low. If the two input vectors are not bit-to-bit equivalent, then EQ goes Low and NE goes High. The circuit is implemented with SNAP as shown in Figure 3.

Table 1. Function Table

$A_9 - A_0$	$B_9 - B_0$	EQ	NE
a	a	H	L
b	b	H	L
a	b	L	H
b	a	L	H

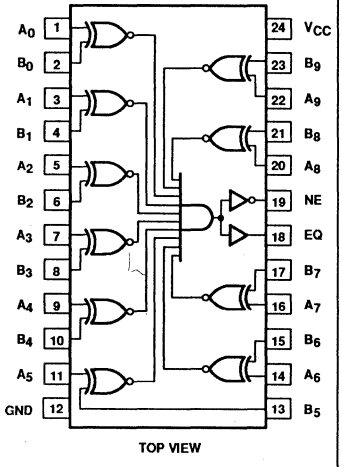


Figure 1. Logical Equivalent Circuit of 10-Bit Comparator

RESOURCES

This design used 20 product terms in the PLS173. As shown in Figure 4, expanded equations, each output needs 20 product terms. Since the product terms are the same, each output shares the 20 product terms with an output polarity fuse determining the proper output level.

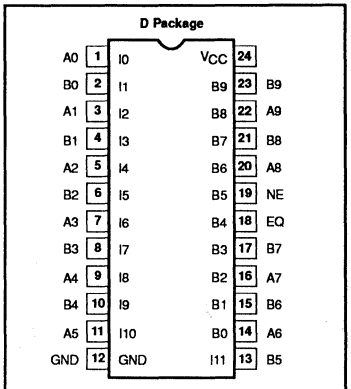


Figure 2. Pin Configuration

This circuit compares to two 10-bit inputs. If they are bit-to-bit equivalent, output EQ goes high and NE goes low. If they are not bit-to-bit equivalent, output EQ will be low while NE will be high.

```
@PINLIST
A[0..9]   i;
B[0..9]   i;
EQ        o;
NE        o;

@GROUPS
one       =  a[0..9];
two       =  b[0..9];

@TRUTHTABLE
@LOGIC EQUATIONS

EQ = one == two;
NE = one != two;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

These equations are written for Philips Semiconductors SNAP software.

Figure 3. Boolean Equations of 10-Bit Comparator

```
@LOGIC EQUATIONS
eq      =  /( ( /b9 * a9 )
        +  ( b9 * a9 )
        +  ( b8 * / a8 )
        +  ( / b8 * a8 )
        +  ( / b7 * a7 )
        +  ( b7 * / a7 )
        +  ( b6 * / a6 )
        +  ( / b6 * a6 )
        =  ( b5 * / a5 )
        +  ( b5 * / a5 )
        +  ( / b5 * a5 )
        +  ( / b4 * a4 )
        +  ( b4 * a4 )
        +  ( / b3 * a3 )
        +  ( b3 * / a3 )
        +  ( / b2 * a2 )
        +  ( b2 * / a2 )
        +  ( / b1 * a1 )
        +  ( b1 * / a1 )
        +  ( b0 * / a0 )
        +  ( / b0 *  a0 ) );


ne      =  ( ( /b9 * a9 )
        +  ( b9 * a9 )
        +  ( b8 * / a8 )
        +  ( / b8 * a8 )
        +  ( / b7 * a7 )
        +  ( b7 * / a7 )
        +  ( b6 * / a6 )
        +  ( / b6 * a6 )
        =  ( b5 * / a5 )
        +  ( b5 * / a5 )
        +  ( / b5 * a5 )
        +  ( / b4 * a4 )
        +  ( b4 * a4 )
        +  ( / b3 * a3 )
        +  ( b3 * / a3 )
        +  ( / b2 * a2 )
        +  ( b2 * / a2 )
        +  ( / b1 * a1 )
        +  ( b1 * / a1 )
        +  ( b0 * / a0 )
        +  ( / b0 *  a0 ) );
```

Figure 4. Expanded Equations

9-Bit parity generator/checker with PLS153/153A

AN021

INTRODUCTION

This application note presents the design of a parity generator using Philips Semiconductors PLD, PLS153 or PLS153A, which enables the designers to customize their circuits in the form of "sum- of-products". The PLA architecture and the 10 bi-directional I/O's make it possible to implement the 9-bit parity generator/checker in one chip without any external wiring between pins.

The parity of an 8-bit word is generated by counting the number of "1's" in the word. If the number is odd, the word has odd parity. If the number is even, the word has even parity. Thus, a parity generator designed for even parity, for example, will generate a "0" if the parity is even, or a "1" if parity is odd. Conversely, an odd parity generator will generate a "0" if the parity of the word is odd, or a "1" if the parity is even. This bit is then concatinated to the word making it 9-bits long. When the word is used elsewhere, its parity may be checked for correctness.

FEATURES

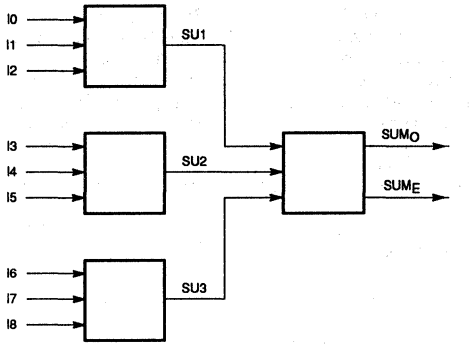
- Generates even and odd parities (SUME and SUMO)
- SUME = "1" for even parity, "0" for odd parity
- SUMO = "1" for odd parity, "0" for even parity
- Generate parity or check for parity errors
- Cascaded to expand word length

DESCRIPTION

The most straightforward way of implementing the parity generator/checker is to take the 9-input truth table (8 inputs for the 8-bit word, and 1 input for cascading the previous stage) and put it in a 256 × 4 PROM. Since there are 2^9 combinations and half of them is odd, the other half is even, the circuit will take 256 terms.

An alternative is to divide the 9-bits into 3 groups of 3-bits as shown in Figure 1. If the sum of the 3-bits is odd, then the intermediate output SU1, or SU2, or SU3 equals 1. Otherwise it equals 0. The intermediate results are grouped together and SUMO becomes "1" if the sum is odd, otherwise SUMO equals "0".

The circuit is implemented using SNAP as shown in Figure 3. SU1 is an intermediate output for inputs I_0, I_1 and I_2. In the same manner, SU2 and SU3 are intermediate outputs for I_3, I_4, I_5 and I_6, I_7, I_8.



I_0	I_1	I_2	SU1
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	0	1	0
1	0	0	1
1	1	0	0
1	1	1	1

I_3	I_4	I_5	SU2
0	0	0	0
0	1	1	0
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0
1	0	0	1
1	1	1	1

I_6	I_7	I_8	SU3
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0
0	0	1	1
0	1	0	1
1	0	0	1
1	1	1	1

SU1	SU2	SU3	SUMO	SUME
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Figure 1. Block Diagram of 9-Bit Parity Generator/Checker

RESOURCES

The design uses up 20 product terms and 5 control terms leaving 12 product terms and 4 bi-directional I/O's to implement other logic designs.

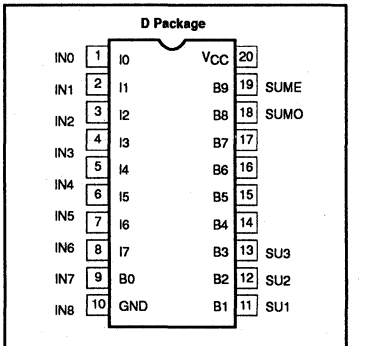


Figure 2. Pin Configuration

9-Bit parity generator/checker with PLS153/153A | AN021

```
@PINLIST          This circuit is a 9-bit parity generator/checker commonly used for error detection in high speed
                  data transmission/retrieval.
 in[0..8]    i;   The odd parity output (SUMO) is high when the sum of the data bits is odd. Otherwise it is low.
 su[1..3]    o;   The even parity output (SUME) is high when the sum of the data bits is even. Otherwise it is low.
 sumo        o;   SU3, SU2 and SU1 are intermediate terms.
 sume        o;   This design was done using the Truth Table Entry method of Philips Semiconductors SNAP
                  software.
@TRUTHTABLE
   [IN2,  IN1,  IN0  :  SU1]
    0     0     0    :  0;
    0     0     1    :  1;
    0     1     0    :  1;
    0     1     1    :  0;
    1     0     0    :  1;
    1     0     1    :  0;
    1     1     0    :  0;
    1     1     1    :  1;

   [IN5,  IN4,  IN3  :  SU2]
    0     0     0    :  0;
    0     0     1    :  1;
    0     1     0    :  1;
    0     1     1    :  0;
    1     0     0    :  1;
    1     0     1    :  0;
    1     1     0    :  0;
    1     1     1    :  1;

   [IN8,  IN7,  IN6  :  SU3]
    0     0     0    :  0;
    0     0     1    :  1;
    0     1     0    :  1;
    0     1     1    :  0;
    1     0     0    :  1;
    1     0     1    :  0;
    1     1     0    :  0;
    1     1     1    :  1;

  [SU3,  SU2,  SU1  :  SUMO,  SUME]
    0     0     0    :  0      1;
    0     0     1    :  1      0;
    0     1     0    :  1      0;
    0     1     1    :  0      1;
    1     0     0    :  1      0;
    1     0     1    :  0      1;
    1     1     0    :  0      1;
    1     1     1    :  1      0;
```

Figure 3. SNAP Truth Table Entry

```
@logic equations
su1     =  ( (in2 * in1 * in0)
           + (in2 * in1 * in0)
           + (/in2 * in1 * /in0)
           + (/in2 * /in1 * in0);

su2     =  ( (in5 * in4 * in3)
           + (in5 * in4 * in3)
           + (/in5 * in4 * /in3)
           + (/in5 * /in4 * in3);

su3     =  ( (in8 * in7 * in6)
           + (in8 * in7 * in6)
           + (/in8 * in7 * /in6)
           + (/in8 * /in7 * in6);

sum0    =  ( (su3 * su2 * su1)
           + (su3 * /su2 * /su1)
           + (/su3 * su2 * /su1)
           + (/su3 * /su2 * su1);

sume    =  ( (su3 * su2 * su1)
           + (su3 * /su2 * /su1)
           + (/su3 * su2 * /su1)
           + (/su3 * /su2 * su1);
```

Figure 4. Expanded Equations

Schmitt trigger using PLS153

AN018

Issued June 1988; revised October 1990; revised September 1993

INTRODUCTION

One of the many features of the PLS153 is the availability of individually controlled 3-State I/O pins. Taking advantage of this feature, a Schmitt trigger may be constructed using one input pin, two bidirectional I/O pins and additional components of three resistors. The two threshold voltages, as well as the hysterisis, are determined by the values of the three resistors and the parameters of the PLS153 device, which are 1) input threshold voltage, V_{TH}, 2) High output voltage, V_{OH}, and 3) Low output voltage, V_{OL}. The circuit may be simplified if Schmitt function is needed only on Low going High or High going Low, and if the hysterisis and threshold voltages are not important.

DESCRIPTION

A simplified block diagram of a non-inverting Schmitt trigger is shown in 1 where R_1, R_2, and R_1, R_3, form two pairs of voltage dividers one of which get into action at input voltage direction of High going Low and the other Low going High. Assuming that input voltage starts at zero volt, the output voltage is therefore at V_{OL} which causes Q_2 to pull R_3 towards ground. As the input voltage increases, only a fraction of the voltage is impressed upon the input buffer due to the dividing network R_1 and R_3. As soon as the input voltage reaches a point where $V_1=V_{TH}$ (V_{TH}=1.38V typical), the output switches to V_{OH} which, in turn, turns off Q_2 and turns on Q_1. V_1 will jump to a value greater than V_{TH} and Q_1 then pulls the input pin, through R_2, towards V_{OH}, which in turn locks the output to a High state even if the input voltage fluctuates, as long as it does not fluctuate outside of the designed hysterisis. When the input voltage goes from a High to a Low, the Schmitt function repeats itself except that Q_1 and Q_2 reverse their roles.

The triggering voltages, V_H (Low going High) and V_L (High going Low) are:

$V_H=V_{TH}\,[(R_1+R_3)/R_3] - V_{OL}\,(R_1/R_3)$;

$V_L=V_{TH}\,[(R_1+R_2)/R_2] - V_{OH}\,(R_1/R_2)$;

where, at room temperature, V_{CC} = 5.0V, I_{OH}/I_{OL}<1mA. V_{TH} is the threshold voltage of the device, typically 1.38V; V_{OL} is the output Low voltage of the device, typically 0.36V at $|I_{OL}|$ <1mA; V_{OH} is the output High voltage of the device, typically 3.8V at $|I_{OH}|$ <1mA.

The implementation of 1 using PLS153/153A is as shown in Figures 2a and 2b. A scope photo of the operation of the circuit is shown in Figure 6.

An inverting Schmitt triggered buffer may be constructed using the same principle. A simple block diagram of such inverter is shown in Figure 3a. The circuit is implemented as shown in Figures 3b and c.

If the voltage levels (V_L and V_H) and the hysterisis are not critical, one I/O pin may be used to pull the input pin High and Low. Therefore one I/O pin and a resistor may be saved. The drawback is that the range of V_H and V_L is quite limited. The circuit is as shown in Figure 4.

If Schmitt function is needed only in one direction, one of the resistor/output circuit may be eliminated. The circuit is as shown in Figure 5.

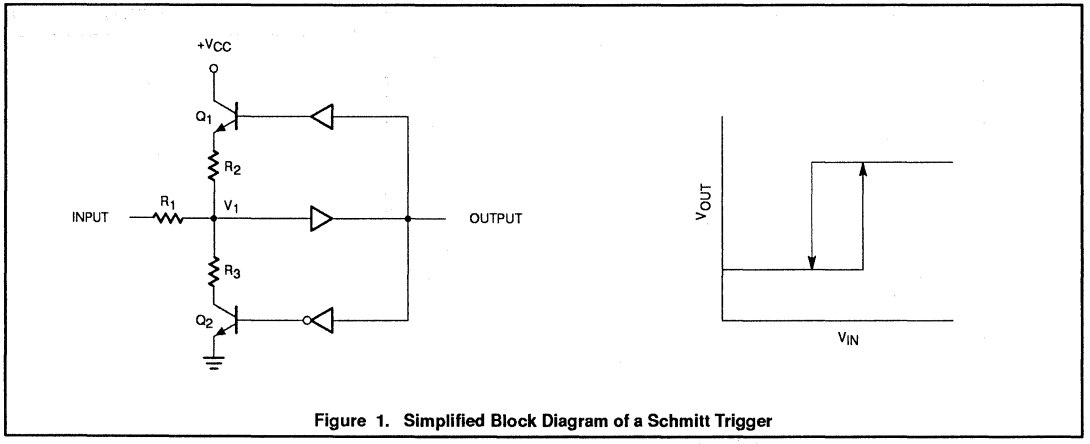
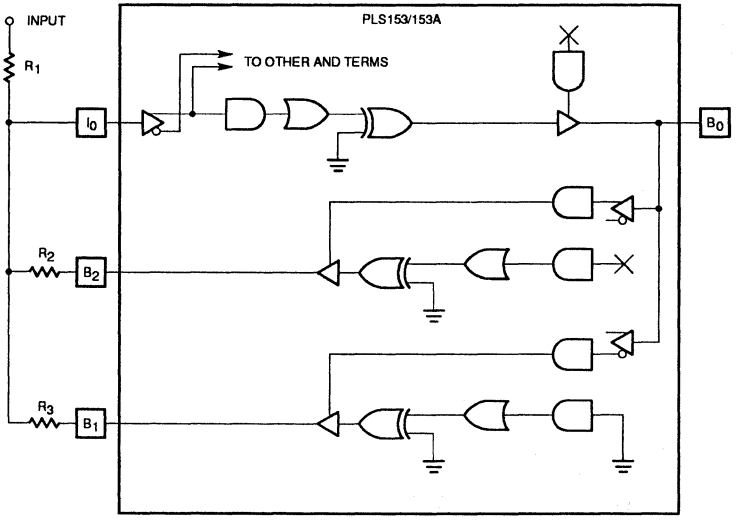


Figure 1. Simplified Block Diagram of a Schmitt Trigger

Schmitt trigger using PLS153 AN018



a. Using PLS153/153A

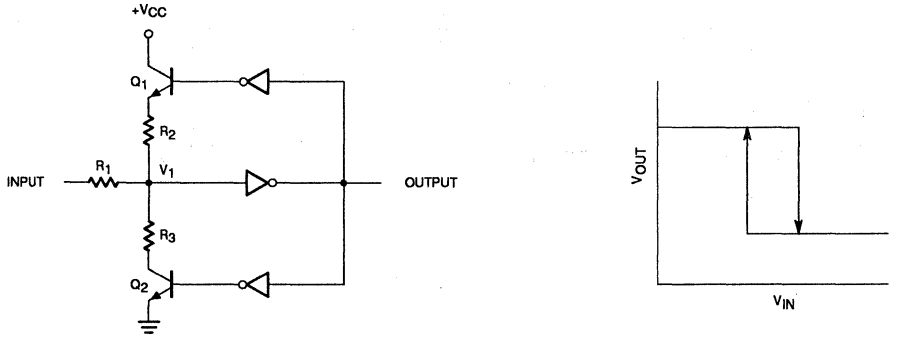
NOTE: Schmitt trigger output may be obtained from both I_0 and $B(I)_0$ to drive the AND-ARRAY.

SNAP LISTING FOR A SCHMITT TRIGGER FUNCTION. EXTERNAL RESISTORS ARE REQUIRED AS SHOWN ABOVE.

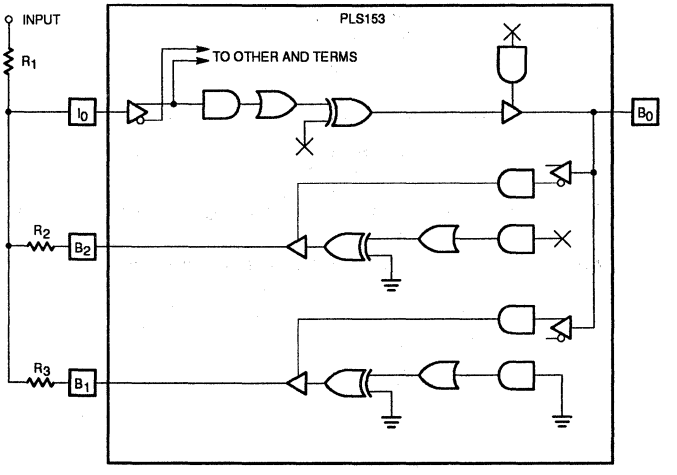
```
@pinlist
input     i;
output    b
B1        o;
B2        o;

@logic equations
output    =        input;
output.oe =        1

B1.oe     =        / output;
B1        =        0;

B2.oe     =        output
B2        =        1;
```

b. SNAP Equations

Figure 2. Schmitt Trigger



a. Simplified Block Diagram



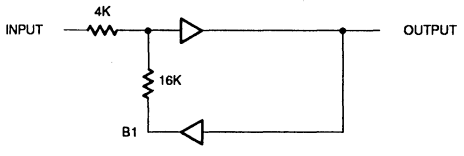
b. Using PLS153

SNAP LISTING SHOWING INVERTING SCHMITT TRIGGER FUNCTION.

```
@pinlist
input      i;
output     b
B1         o;
B2         o;

@logic equations
output     =        / input;
output.oe  =        1

B1.oe      =        output;
B1         =        0;

B2.oe      =        output
B2         =        1;
```

c. SNAP Equations

Figure 3. Inverting Schmitt Trigger



V_L = 0.78V, V_H = 1.6V

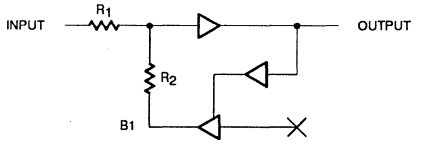
a.

SNAP LISTING SHOWING SCHMITT TRIGGER USING ONE I/O PIN TO PULL INPUT HIGH OR LOW.

```
@pinlist
input     i;
output    b
B1        o;

@logic equations
output    =        input;
output.oe =        1

B1        =        output;
B1.oe     =        1;
```

b.

Figure 4. Schmitt Trigger Using One I/O Pin

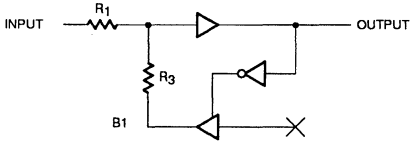


a. High Going Low Direction

SNAP LISTING FOR SCHMITT TRIGGER FUNCTION IN THE HIGH TO LOW DIRECTION ONLY.

```
@pinlist
input     i;
output    b
B1        o;

@logic equations
output    =        input;
output.oe =        1

B.oe      =        output;
B1        =        1;
```

c.



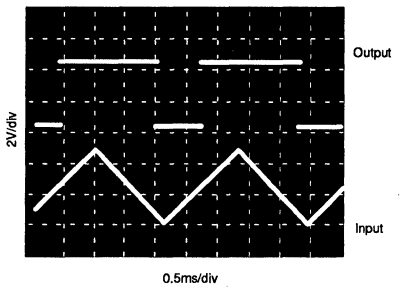
b. Low Going High Direction

SNAP LISTING FOR SCHMITT TRIGGER FUNCTION IN THE LOW TO HIGH DIRECTION ONLY.

```
@pinlist
input     i;
output    b
B1        o;

@logic equations
output    =        input;
output.oe =        1

B1.oe     =        /output;
B1        =        1;
```

d.

Figure 5. Schmitt Trigger



NOTE:
R_1 = 3.9kΩ, R_2 = 10.8k..kΩ, R_3 = 2.0kΩ,
V_{CC} = 5.0V, Ambient temperature = 25°C

Figure 6. A Non-Inverting Schmitt Triggered Buffer

Sequencer devices

INTRODUCTION

Ten years ago, in their search for a straightforward solution to complex sequential problems, Philips Semiconductors originated Programmable Logic Sequencers. Philips Semiconductors Programmable Sequencers represent a product line which combines the versatility of two programmable arrays (PLA concept) with flip-flops, to achieve powerful state machine architectures.

Each arrangement or "architecture" offers a variation of the basic concept which combines two programmable logic arrays with some flip-flops, in an undedicated fashion. The PLA product terms are not specifically dedicated to any particular flip-flop. All, none, or any mix in between may be connected to any flip-flop the designer chooses. The PLA structure therefore supports 100% product term-sharing as well as very wide OR functions preceding the flip-flops.

Philips Semiconductors line of Programmable Logic Sequencers has been further customized to accommodate specific types of state machine designs. Some have both registered and combinatorial outputs, specifically for synchronous and asynchronous Moore-type state machines. Others have state or buried registers, as well as output registers. These devices (PLUS105, PLC42VA12 and PLUS405) are ideal for synchronous Mealy-type applications.

J-K and S-R register functions are another benefit. The logic functions provided by these types of registers far exceed the capability of a D-type register. The functionality of the J-K allows the designer to optimize the logic used in generating state transitions. Ninety percent of PAL devices have D-type registers. All the sequencers are equipped with three state options for bussing operations, JK or SR flip-flops and some form of register Preset/Reset functions.

Finally, all PLS devices have a Transition Complement Array. This asynchronous feedback path, from the OR array to the AND array, generates "complement" transition functions using a single term. Virtually hidden in between the AND array and the OR array is the Complement Array. This single NOR gate is not necessarily "an array," however the inputs and outputs of this complement gate span the entire AND array. The input(s) to the Complement Array can be any of the product terms from the AND array. The output of the Complement Array will be the 'complement' of the product term input. If several product terms are connected to the Complement Array, their respective complements can also be generated. The output of the Complement Array is fed back to the AND array, whereby it can be logically gated through another AND gate and finally propagated to the OR array. The significance being that the complement state of several product terms can be generated using one additional AND product term. For example, if an efficient method of sensing that no inputs were asserted was needed, the designer could connect the output of appropriate AND gates to the complement NOR gate. The output of the NOR gate could then be used to condition and then set or reset a flip-flop accordingly. As well, he could detect a particular state variable combination and force a transition to a new state, independent of the inputs. Or he could combine input signals and state (AND) terms to generate a new composite term. In any of these applications, the Complement Array greatly reduces the number of state transition terms required.

In order to present the material in the most concise fashion, a brief state equation tutorial is presented first. The PLUS105 description immediately follows. In this capsule description, the level of detail is expanded, so read it first for basic understanding. Each additional presentation will be done with regard to the fundamentals described for the PLUS105. Figure 3 shows the detailed drawing of the PLUS105 in full detail. Figure 4 shows a compressed rendition of the same diagram so that the reader can understand the diagram notation. The compressed shorthand version will be used for the rest of the sequencers.

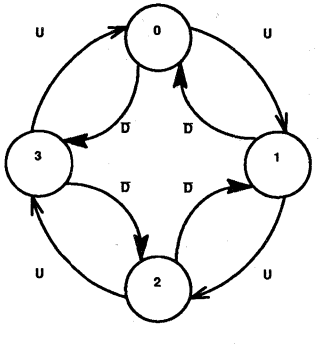


Figure 1. Up-Down Counter State Diagram

```
While     [STATE 0]
          IF      [U]     THEN        [STATE 1]
          IF      [/D]    THEN        [STATE 3]

While     [STATE 1]
          IF      [U]     THEN        [STATE 2]
          IF      [/D]    THEN        [STATE 0]

While     [STATE 2]
          IF      [U]     THEN        [STATE 3]
          IF      [/D]    THEN        [STATE 1]

When      [STATE 3]
          IF      [U]     THEN        [STATE 0]
          IF      [/D]    THEN        [STATE 2]
```

Figure 2. STATE EQUATIONS to Implement Up-Down Counter for JK or SR Type Flip-Flop Based Sequencer

Sequencer devices

State Equation Tutorial

STATE equation entry is a convenient way to describe elementary sequential machines in a manner which is directly related to a state diagram of the machine. The basic commands are few, but can be combined in a powerful fashion. Figure 1 shows a 4 state up-down counter for a machine with an U(up)/D(down) input line. Figure 2 shows the state equation syntax to implement Figure 1.

The basic meaning can be summarized in the following way. Simply, "while in state X" if input "Y" occurs, "transverse to state Z". This is a Moore machine model. Mealy may be accommodated by addition of the "with" operation which designates an output variable being associated as shown below:

A.) While		[CURRENT STATE]
	with	[OUTPUT VARIABLE]
	IF	[INPUT VARIABLE]
	then	[NEXT STATE]

or

B.) While		[CURRENT STATE]
	IF	[INPUT VARIABLE]
	then	[NEXT STATE]
	with	[OUTPUT VARIABLE]

If a latched output variable is desired, the addition of a prime notion (/) to the right of the output variable is required.

The designer must assign the binary values of choice to specific states for a state equation function to be implemented. The Philips Semiconductors SNAP manual details state equation solutions with more examples, but the advantage of state equations is that the designer can be less involved with the internal structure of the sequencer than required by other methods.

The PLUS105

This part (Figure 3) has sixteen logic inputs and eight outputs. It also has eight S-R flip-flops tied directly to those output pins through 3-State buffers (common control from pin 19). The user may select pin 19 to be an Output Enable signal or an asynchronous preset (PR) signal which is common to all flip-flops. Embedded into the device are 48 AND gates. All flip-flops are S-R type with an OR gate on both S and R. The designer may choose any number of product terms and connect them with any OR gate. The product terms can also be shared across any OR gate, as needed. Six of the 14 flip-flops are termed "buried registers" as their outputs are fed back to the AND array, regenerating both the Q and /Q state variables. There is no direct connection to an output. Both the input signals and the state variables Q and /Q are fed to the AND array through buffers which provided the TRUE (or noninverted) and Complement (inverted) renditions of the variable. This is critical for the efficient use of the AND array. The designer has all state and input variables necessary to generate any state transition signal to set and/or reset commands to the flip-flops. Because of this AND/OR arrangement, combined with complete freedom of configuration, all sequential design optimization methods are applicable.

There are many other feature capabilities suitable for creative usage. For example, it is common practice to use the 48 product terms with the 6-bit buried register, treating the output 8-bit register as an intermediate, loadable data register only. This provides a very good bus "pipeline" for the internal 6-bit machine. However, other logic options can be accomplished by combining internal state information (present state) with current input information, generating a next state which is different from the current internal state.

Sequencer devices

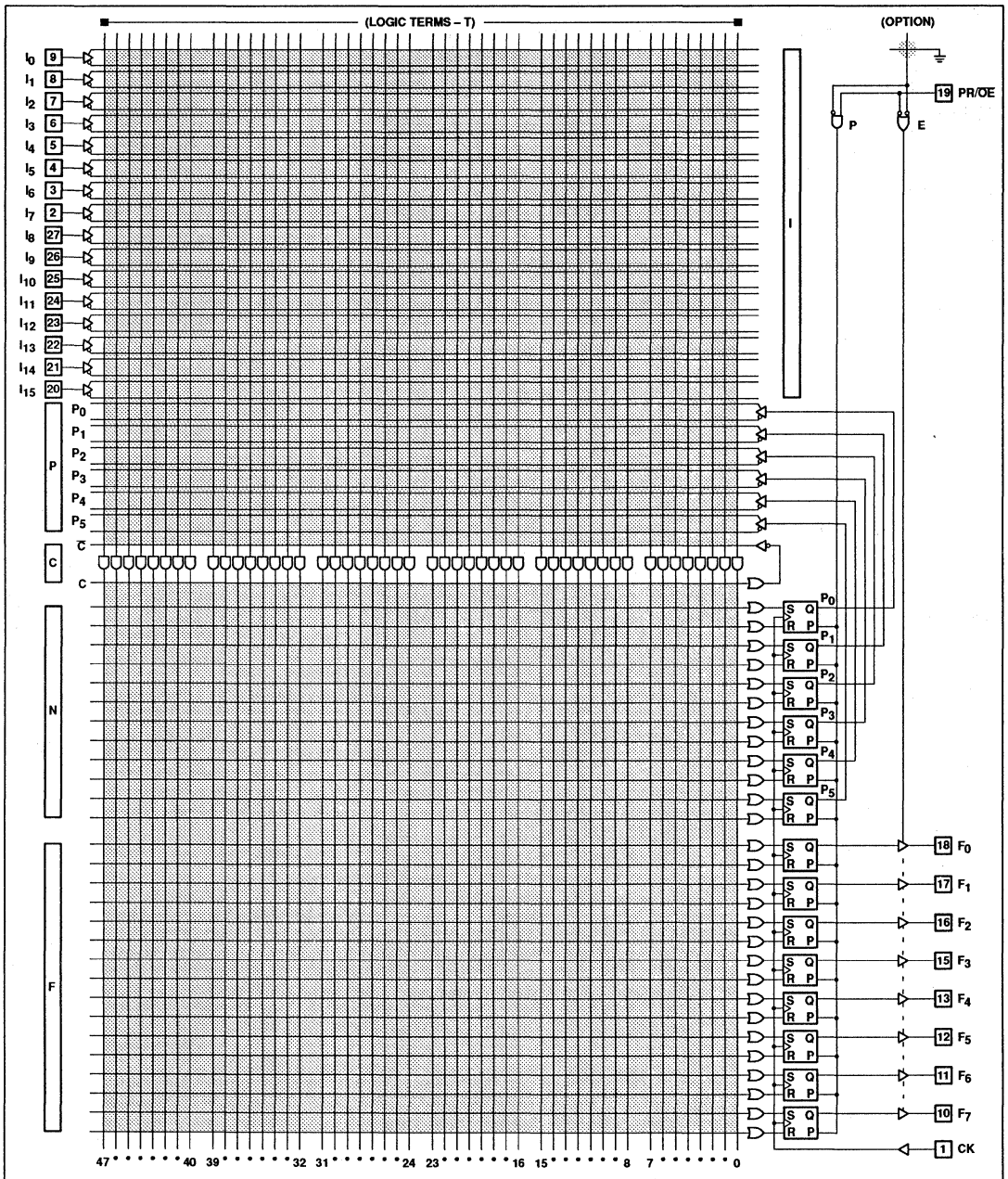


Figure 3. PLUS105

Sequencer devices

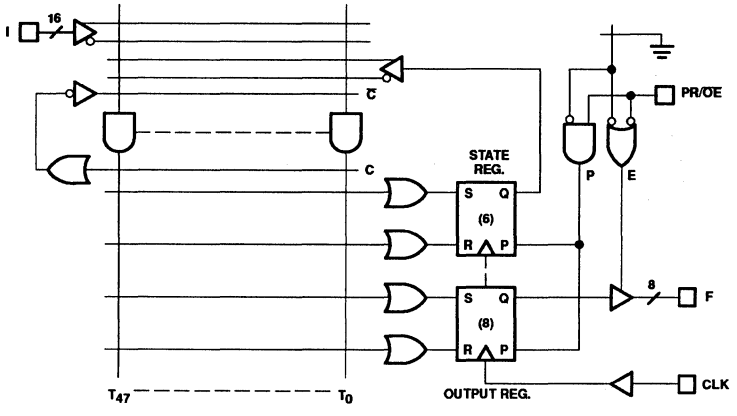


Figure 4. Compressed Drawing of PLUS105

Sequencer devices

The PLS155, 157, and 159A constitute a three part family of 20-pin sequencers that are well suited for high speed handshakers, counters, shift registers, pattern detectors and sequence generators. Additional applications include testability enhancement, demonstrated in the application examples of signature analysis and pseudo random number generation. The three devices are very similar in architecture. All have a total of 12 possible outputs. The difference is the ratio of combinational I/O to registered outputs available.

The PLS155

The PLS155 is a sequencer providing four J-K flip-flops with a PLA having 32 logic product terms and 13 control product terms. Eight combinational I/O are available in addition to the four registered outputs. All of the state variables and combinational variables are presented to the output pins by way of 3-State inverting buffers. The combinational and state variable outputs are fully connected (fed back) back to the AND array in both the True and Complemented form of the variable. The product includes a special feature that allows the user to configure the flip-flops as either J-K or D flip-flops on an individual basis. A Register Preload feature is supported via two product terms (La, Lb) which permit "back loading" of data into the flip-flops, directly from the output pins. The part can now be easily forced into any known state by enabling La, Lb, applying data at the outputs (previously "3-Stated"), and applying a clock pulse. Register Preset and Reset functions are controlled in 2 banks of 2 registers each. Note that control product terms are from the OR array.

The outputs of all variables are 3-State controlled by a unique partition. Pin 11 provides an Output Enable input (OE) which can be asserted with the EA and EB control product terms. EA controls the flip-flops F0 and F1, and EB controls F2 and F3. Each combinational output term has a distinct 3-State control term (D0 - D7) originating from the AND array of the PLA. Each combinational output variable can be programmed as inverting (active LOW) or non-inverting (active HIGH) by way of the output polarity EX-OR gate associated with each I/O pin.

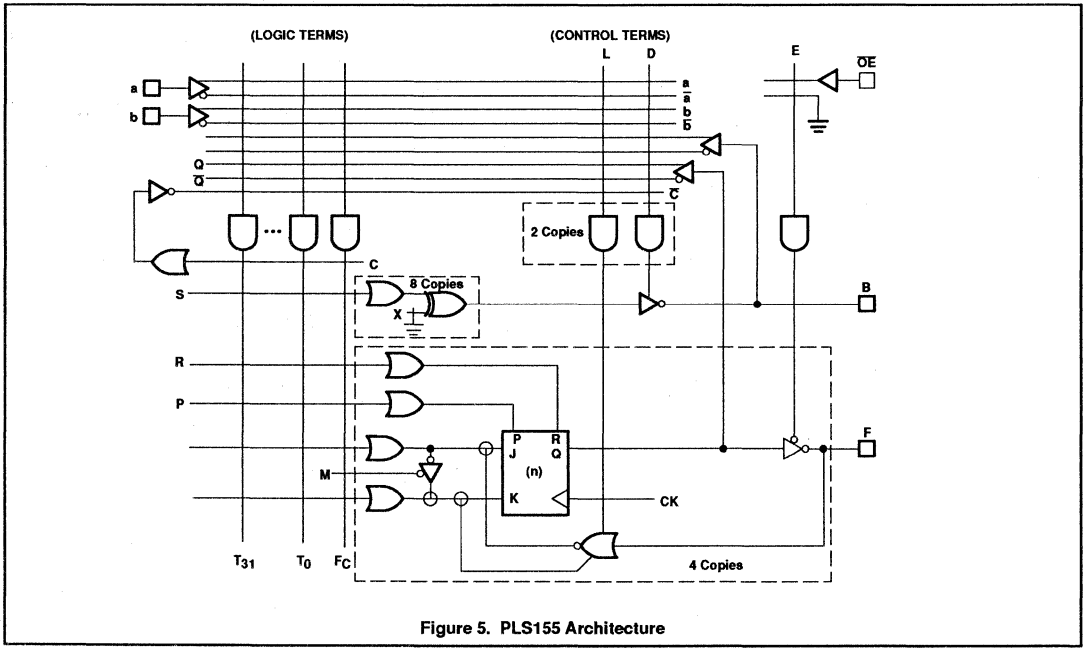


Figure 5. PLS155 Architecture

Sequencer devices

The PLS157

This sequencer features all the attributes of the aforementioned PLS155, however, two flip-flops have been added, at the expense of two of the combinational outputs. Pins 13 and 18 on the PLS157 are flip-flop driven, where the same pins on the PLS155 are combinatorial, driven from the PLA. Again, all variables (input, output, or state variables) fully connect over the PLA portion with both True and complemented versions supplied.

The number of product terms, the Complement array, Output Enable, 3-State configurations, Register Preload, etc., track the PLS155 part. As with the PLS155, distinct clock input on pin 1 is provided for synchronous operation. Register Preset and Reset are available in 2 banks. Pin F_4 and F_5 are controlled from the AND array (Product Terms P_B and R_B). The remaining 4 registers, F_0 - F_3, are controlled by the sum terms (from the OR array) P_A and R_A.

Designs requiring more than 16 states but less than or equal to 64 states are solid candidates for realization with the PLS157. It can be configured as a Moore machine for counter and shifter designs from the flip-flop outputs, or as high speed pulse generators or sequence detectors with the combinational outputs. Mixed solutions are also possible.

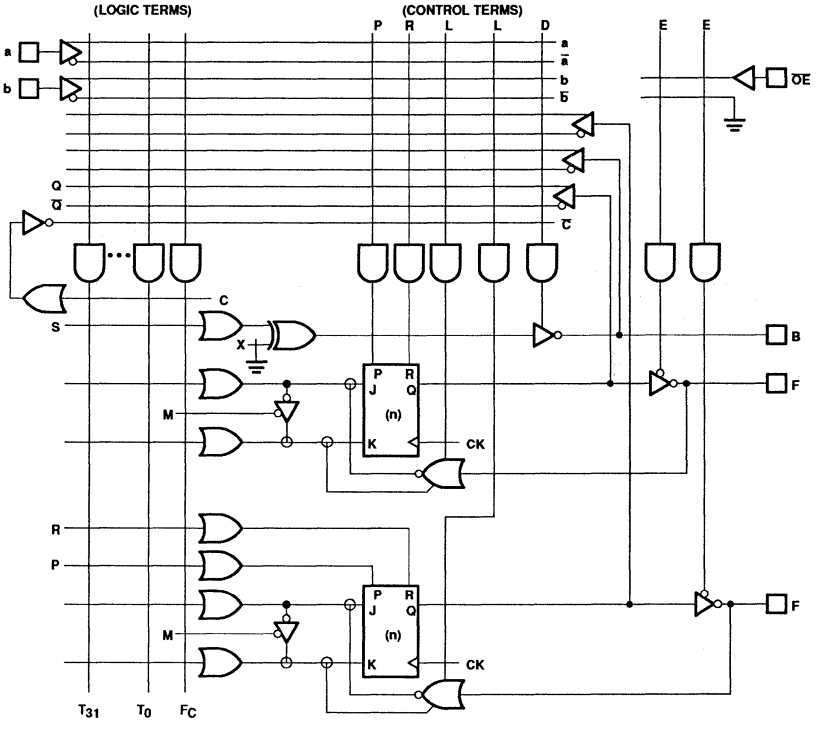


Figure 6. PLS157 Architecture

Sequencer devices

The PLS159A

By extending the PLS157 arrangement even further, the PLS159A can be derived. Again, maintaining identical input, product terms, Complement array and similar 3-State partitioning, the PLS159A also resides in a 20-pin package. The expansion to dual 4-bit banks of flip-flops, at the expense of 2 combinational outputs, enhances the number of available internal states while maintaining product term and pin compatibility. Note that all registers are controlled from the AND array in 2 groups of four.

The PLS159A is an octal part. It readily enters the environment of the 8-bit data operand as well as the bus oriented system.

For enhanced performance, the flip-flop outputs are inverted. To provide positive outputs for shifters and counters, the input variables and state feedback variables can be selectively inverted through an input receiver or the feedback path through the AND gate array.

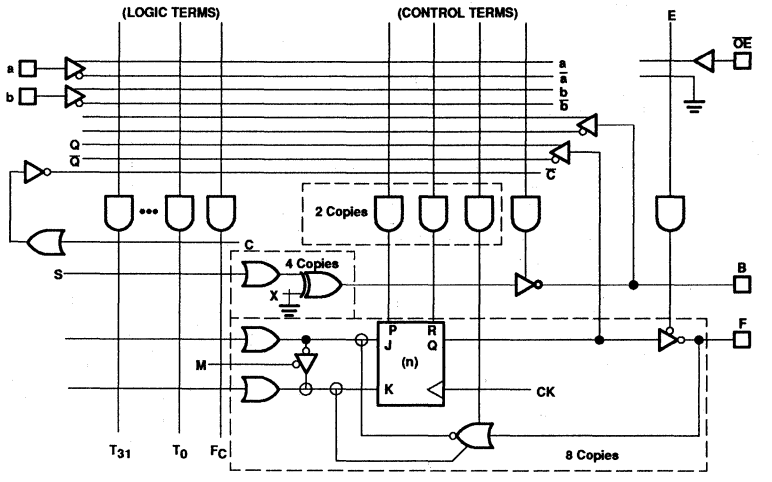


Figure 7. PLS159A Architecture

Sequencer devices

There are three basic members in the 24-pin package family: The PLS167A, the PLS168A, and the PLS179. The PLC42VA12 is discussed elsewhere.

The PLS167A

The PLS167A has 14 logic inputs and six registered outputs (S-R flip-flops). Six additional buried flip-flops reside beside the 48 product term AND array. This device can support state machine designs of up to 256 states—as two outputs feed back into the AND array, making a total of eight buried registers. There is complete feedback connectivity of the inputs and the state flip-flop outputs to the PLA AND gates. Organizationally it has much more in common with the PLS105A than the aforementioned 20-pin parts. The asynchronous Preset and the Output Enable are identical to the PLS105A.

By having the output latched state variable capability, it provides an automatic buffer for bus based systems. The current state may be presented, fully stable and synchronized to a bus—while the internal buried machine is transitioning to the next state based on current input conditions.

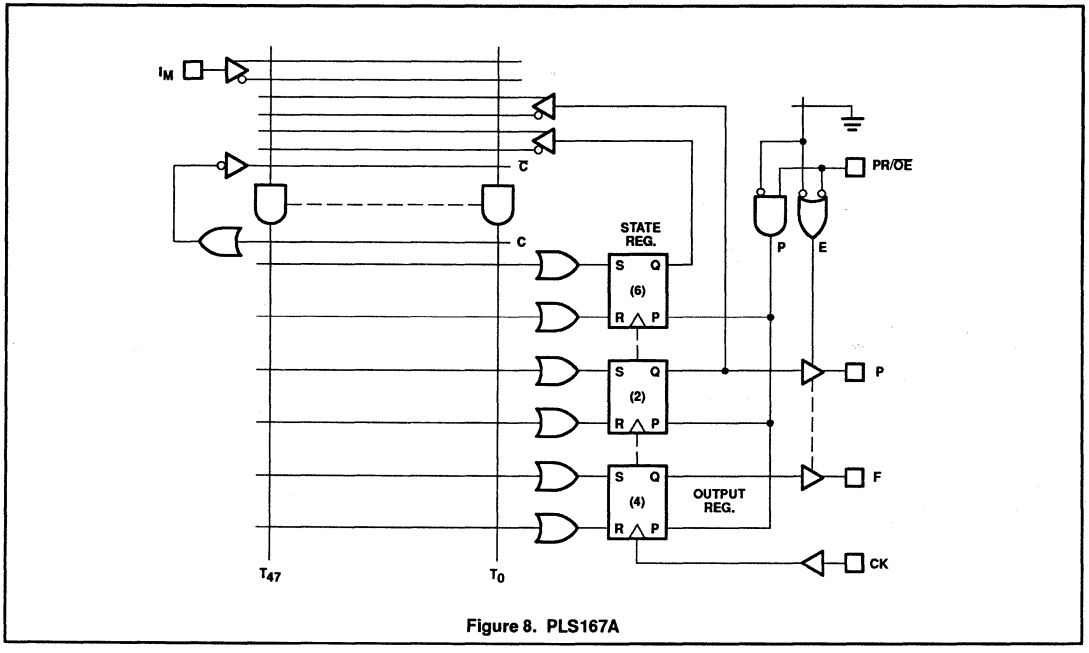


Figure 8. PLS167A

Sequencer devices

The PLS168A

This sequencer is a down-scaled version of the PLS105A. Having identical product terms, Complement array, asynchronous PRESET/Output Enable options, and 3-State controls, its primary difference is having 12 inputs compared to the PLS105A's 16 inputs. However, the PLS168A can become a state machine of up to 1024 states due to internal feed back of its six state registers, plus the feedback of four of the eight output registers. The PLS168A is packaged in a 300mil-wide 24-pin DIP or 28-pin PLCC.

This is also an octal part, providing an 8-bit register to a bus based system. State registers, interrupt vector synchronizers, counters, shifters, or just about any basic state machine can be generated and 3-State interfaced to a computer bus with a PLS168A. Outputs provided by the positive asserted sense make state transitioning and loading of state variables straightforward.

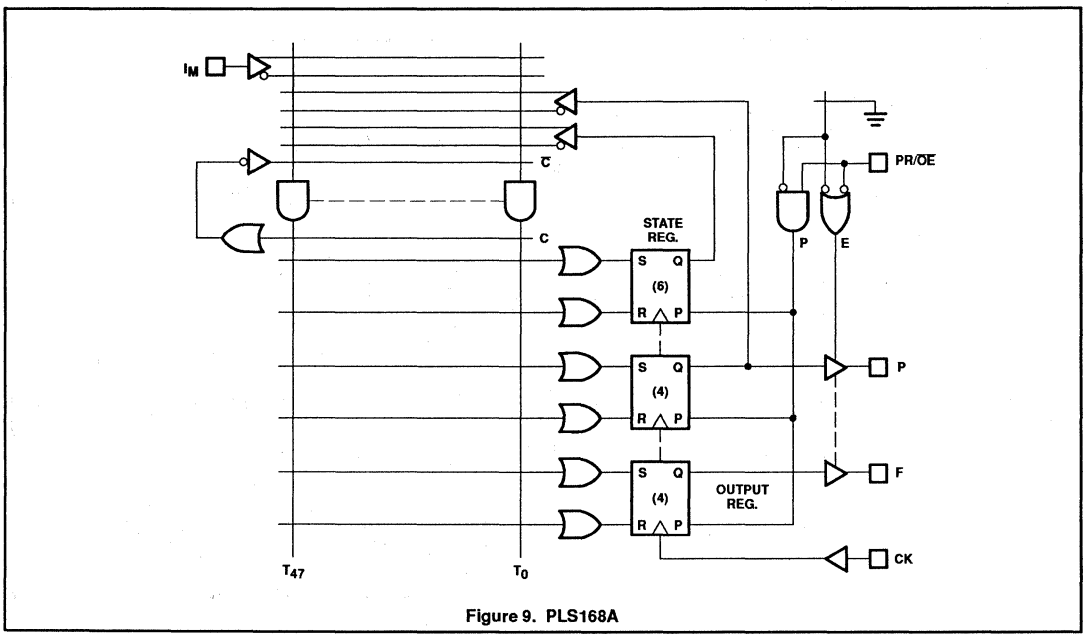


Figure 9. PLS168A

Sequencer devices

The PLS179

The PLS179 is architecturally similar to the PLS159A. The 3-State enable, number of product terms, flip-flop mode controls, register preload, etc., are all identical to the PLS159A. The four additional inputs are the dominant differentiating feature for this part as compared to the PLS159A. As with the PLS159A, the PLS179 Preset and Reset functions are controlled from the AND array in 2 groups of 4 registers each.

The PLS179 is also an octal part. Providing the state contents directly to the pin through 3-State buffers allows counters and other sequence generators direct access to an asserted low octal bus. Some design creativity will generate positive assertion through the pin inverters, for positive driven busses. Additional input pins expand the capability of the part beyond the PLS159A. Input combinations may be presented in a wider format, more fully decoded to the sequencer for faster reaction and less external circuitry than the PLS159A requires.

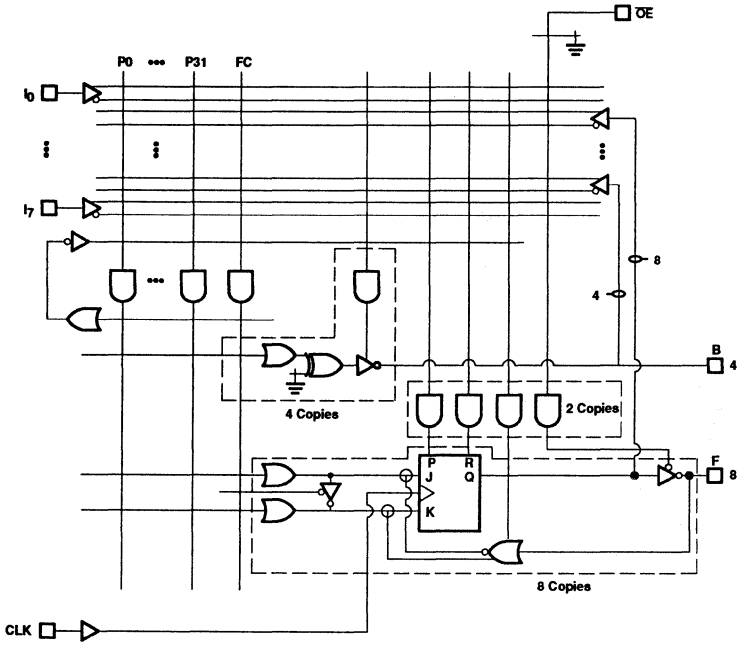


Figure 10. PLS179 Architecture

Sequencer devices

The PLUS405

The PLUS405 is a functional superset of the PLUS105. It is also much faster. The performance of the PLUS405 has been dramatically improved relative to the PLS105A. Available in two speed versions, the operating frequencies ($^1/_{t_{IS}} + t_{CKO}$) range from 37 to 45MHz (minimum guaranteed frequency). The clock frequencies, or toggle rate of the flip-flops, are 50MHz and 58.8MHz, respectively. The PLUS405 has 16 more product terms and two more buried state registers than the PLUS105. Equipped with two independent clocks, it is partitionable into two distinct state machines with independent clocks. And, it contains two independent Complement arrays, allowing full benefits over both machines.

The PLUS405 can be partitioned as one large state machine (16FFs) with 64 available p-terms using one clock and 16 inputs or alternately two state machines (8FFs each) with independent clocks, sharing 64 p-terms with 15 inputs in any combination the user desires. The Complement arrays can be used to generate the "else" transition over each state machine or alternately used as NOR gates. They can be coupled into a latch if needed.

The Asynchronous Preset option of the PLS105/167/168 architectures has been replaced with a Programmable Initialization feature. Instead of a Preset to all logic "1"s, the user can customize the Preset/Reset pattern of each individual register. When the INIT pin (Pin 19) is raised to a logic "1", all registers are preset/or reset. The clocks are inhibited (locked out) until the INIT signal is taken Low. Note that Pin 19 also controls the OE function. Either Initialization or OE is available, but not both.

A CMOS extension to the PLUS405 is Philips Semiconductors PLC415, which is pin compatible and a functional superset of the PLUS405 architecture.

Sequencer devices

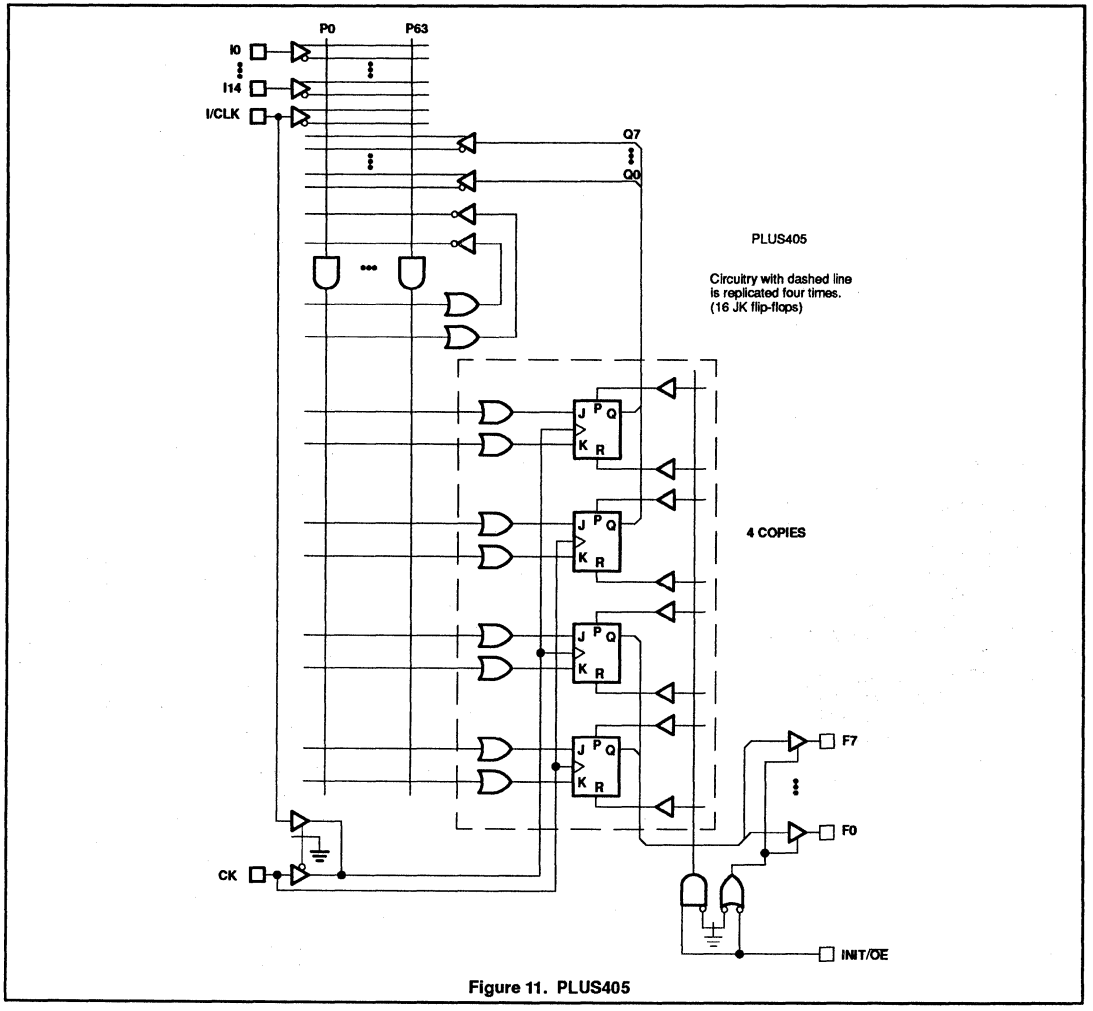


Figure 11. PLUS405

The Future is Here Now.

Recent architectural extensions are currently available from Philips Semiconductors. These include the PLC415 and PLC42VA12. These new "Super Sequencers" are available now for high-end new designs. Please check the data sheet section of this handbook for more information. See also the CMOS Sequencers section for more design examples using the PLC415 and the PLC42VA12.

PLS168/168A Primer

AN023

INTRODUCTION

The PLS168/168A is a bipolar Programmable Logic Sequencer as shown in Figure 1, which consists of 12 inputs, a 48 product term PLA and 14 R/S flip-flops. Out of the 14 flip-flops, six are buried State Registers (P_4-P_9), four Output Registers (F_0-F_3), and four Dual-purpose Registers (P_0-P_3), which may be used as Output or State Registers. All flip-flops are positive edge-triggered. They are preset to "1" at power-up, or may be asynchronously set to "1" by an optional PR/OE pin, which may be programmed either as a preset pin or as an Output Enable pin. Additional features includes the Complement Array and diagnostics features.

ARCHITECTURE

As shown in Figure 2, the device is organized as a decoding AND-OR network which drives a set of registers some of which, in turn, feedbacks to the AND/OR decoder while the rest serve as outputs. Outputs P_0 to P_3 may be programmed to feedback to the AND/OR decoder as State Registers and, at the same time, used as outputs. The user now can design a 10-bit state machine without external wiring. The AND/OR array is the classical PLA structure in which the outputs of all the AND gates can be programmed to drive all the OR gates. The schematic diagram of the AND-OR array is shown in Figure 3. This structure provides the user a very structured design methodology which can be automated by CAD tools, such as Philips Semiconductors SNAP software package. The output of the PLA is in the form of sum-of-products which, together with the RS flip-flops, is the ideal structure for implementation of state machines. (Refer to Appendix A for a brief description of synchronous finite state machines.)

Design Tools

State machines may be implemented easily with the assistance of a PLD design software package. The software, such as Philips SNAP package, allows for various methods of design description entry. State machines may be described using direct H/L table entry, schematic entry, Boolean equations, or state equations. The preferred method is, of course, state equation entry. The syntax for each software package differs but is generally of the form:

WHILE [present state]

IF [input condition] then [new state] with [output]

IF [input condition] then [new state] with [output]

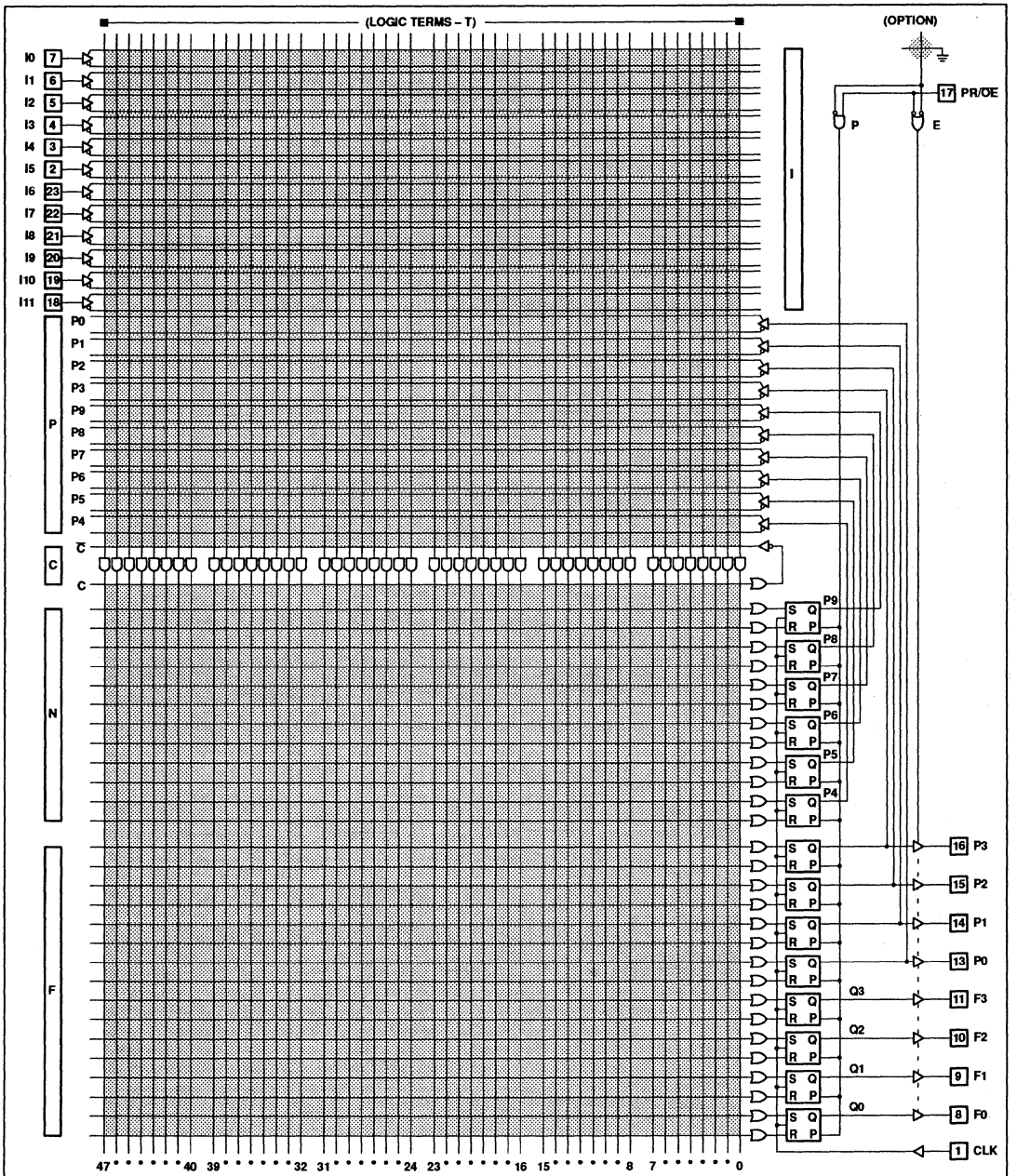
IF [input condition] then [new state] with [output]

Only one input condition should be active at any one time, otherwise two or more product terms trying to force the machine into different states may be active simultaneously. The result would be a state machine in an unexpected state, which would not be a desirable condition. The manual for your software package should be consulted for specific syntax rules and options.

Direct H/L table entry is not recommended for design entry, however looking at a table for verification of a design and for learning how state equations are implemented in a device is useful. SNAP does not provide access to a table from a menu entry. Instead, run the program 'FUSETABLE' from the DOS prompt. ABEL from Data I/O, provides a program called IFLDOC to display the JEDEC file in a H/L table format. Designs implemented in sequencer devices using JK or SR type flip-flops are easier to interpret in a H/L table format than those using D-type flip-flops.

The following examples illustrate how state equations are implemented in a PLS168 device. Two of them illustrate the functioning of the Complement Array and how it may be used to reduce the number of product terms used in a simple state machine design. The last example shows how only one Complement Array may be used in a state machine design which uses multiple ELSE statements.

A PLS168 H/L table as shown in the data sheet is shown in Table 1. The table is organized according to input and output of the PLA decoding network. The lefthand side of the table represent the inputs to the AND-array, which includes input from input pins and present state information from the feedback buffers which feedback the contents of the State Register. The righthand side of the table represents the output of the OR-array, which drives the State and Output Registers as the next state and output. Each column in the lefthand side of the table represents an input buffer, which may be inverting, non-inverting, disconnected or unprogrammed. Each column in the righthand side of the table represents a pair of outputs to the flip-flops, which may be set, reset, disconnected, or unprogrammed. The programming symbols are H, L, —, and 0. (See Figure 4 for details.) For inputs buffers, "H" means that the non-inverting buffer is connected, "L" means that the inverting buffer is connected, "—" means that both inverting and non-inverting buffers are disconnected, and "0" means that both inverting and non-inverting buffers are connected which causes that particular AND-term to be unconditionally Low. On the output side of the table, "H" means that the particular AND-term is connected to the OR-term on the "S" input of the particular flip-flop, "L" means that the AND-term is connected to the "R" side, "—" means that the AND-term is not connected to the flip-flop at all, and "0" means that the AND-term is connected to both the "S" and "R" sides. More details of the symbols and their meanings are shown in Appendix B. Each row in the table represents an AND-term. There are 48 AND-terms in the device. Therefore, there are 48 rows in the table. An example of implementing a transition from one state to another is shown in Figure 4a. The state diagram can be implemented by the PLS168 as shown in Figure 4b. The state diagram is translated into H/L format as shown in Figure 4c. The first column on the lefthand side of the table is for the Complement Array which will be discussed in detail in the next section.



NOTES:
1. All programmed "AND" gate locations are pulled to logic "1".
2. All programmed "OR" gate locations are pulled to logic "0".
3. ▒ Programmable connection.

Figure 1

Table 1. PLS168/168A Programming Table

CUSTOMER NAME _____
PURCHASE ORDER # _____
PHILIPS DEVICE # CF (XXXX) _____
CUSTOMER SYMBOLIZED PART # _____
TOTAL NUMBER OF PARTS _____
PROGRAM TABLE _____
REV _____
DATE _____

AND

INACTIVE	0
GENERATE	A
PROPAGATE	•
TRANSPARENT	—

C_n

INACTIVE	0
I, P	H
$\overline{I}$, $\overline{P}$	L
DON'T CARE	—

I_m, P_s

OR

INACTIVE	0
SET	H
RESET	L
NO CHANGE	—

Ns, Fr

OPTION

PRESET	H
$\overline{OE}$	L

P/E

	AND																				OPTION (P/E) OR														
TERM	C_n	INPUT (I_m)											PRESENT STATE (P_s) (P_x)										REMARKS	NEXT STATE (Ns) (P_x)						OUTPUT (Fr)							
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		9	8	7	6	5	4	3	2	1	0	3	2	1	0
0																																						
1																																						
2																																						
3																																						
4																																						
5																																						
6																																						
7																																						
8																																						
9																																						
10																																						
11																																						
12																																						
13																																						
14																																						
15																																						
16																																						
17																																						
18																																						
19																																						
20																																						
21																																						
22																																						
23																																						
24																																						
25																																						
26																																						
27																																						
28																																						
29																																						
30																																						
31																																						
32																																						
33																																						
34																																						
35																																						
36																																						
37																																						
38																																						
39																																						
40																																						
41																																						
42																																						
43																																						
44																																						
45																																						
46																																						
47																																						
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																		16	15	14	13	11	10	9	8
VARIABLE NAME																																						

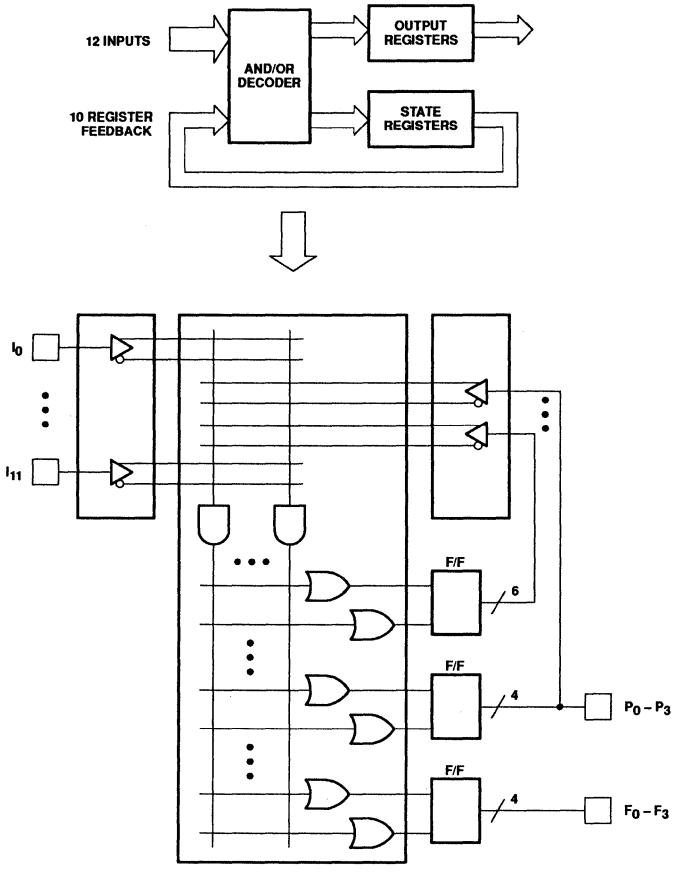


Figure 2. The Architecture of PLS168/168A

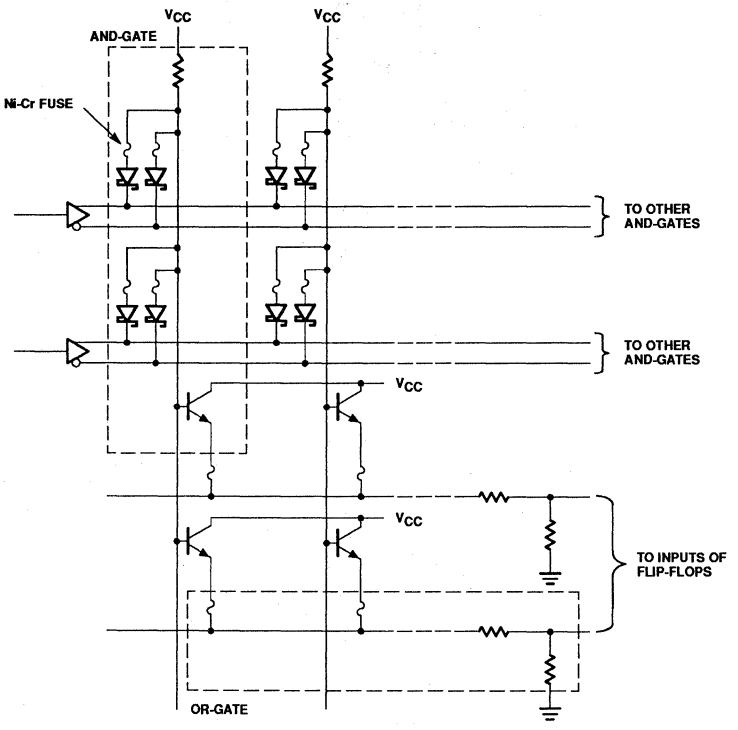
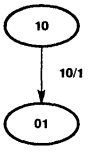
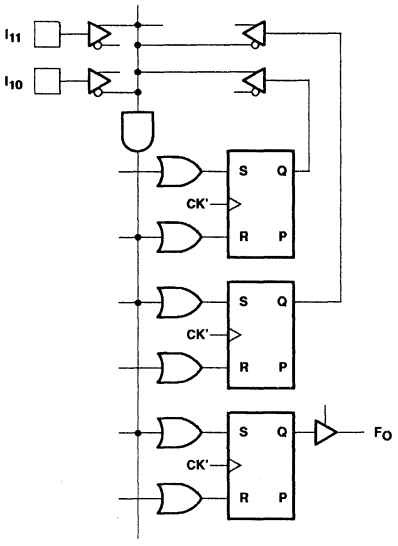


Figure 3. Schematic Diagram of AND-OR Array



a. State Diagram



b. Implementation of State Diagram (a) with PLS168

		AND																				OPTION (P/E)											
																						OR											
TERM	C_n	INPUT											PRESENT STATE									NEXT STATE								OUTPUT			
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0
00		–	H	L	–	–	–	–	–	–	–	–	–	–	–	–	–	H	L	–	–	–	–	–	–	–	–	L	H	–	–	–	–	H	–	–	–
01																																					
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																16	15	14	13	11	10	9	8
NAME		IN1	IN0																														OUT0			

c. PLS168 Programming Table

Figure 4. Implementing State Machine with PLS168

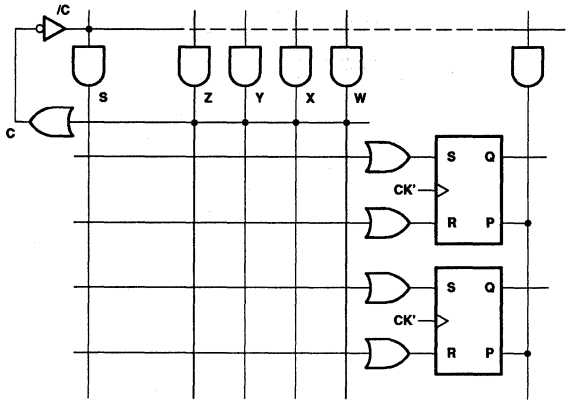


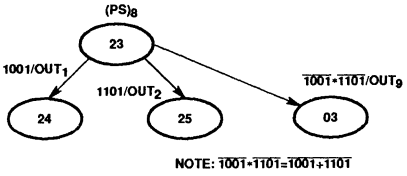
Figure 5. Logic Diagram of Complement Array

Complement Array

An additional feature is the Complement Array, which is often used to provide escape vectors in case the state machines get into undefined states during power-up or a timing violation due to asynchronous inputs. A logic diagram of the Complement Array is shown in Figure 5. The output of the Complement Array is normally Low when one or more AND-terms are High. If all of the AND-terms are Low, then the output of the Complement Array will be High. In this example, if each AND-term is a decoder for a particular state and input combination, and if the circuit gets into an undefined state, none of the AND-terms will be High. Therefore, the output /C will be High, which will then enable the AND-term S which in turn may be used to reset all registers to Low or High as predefined. The state machine thus escapes from being in an undefined state by using the Complement Array and one AND-term.

Without the Complement Array an alternate way of escaping from being in an undefined state is by defining all possible states which are not being defined. This method may require quite a few AND-terms depending on the design. Another application for the Complement Array is illustrated by the following example. As shown in Figure 6, when the machine is in state 23, if input vector equals 1001, it will go to next state 24. If the input is 1101, then go to state 25. But if the input is neither 1001 nor 1101, then go to state 03. It takes only two terms to implement the first two transition vectors.

To implement the third vector "go to state 03 if input is neither 1001 nor 1101", the Complement Array accepts the outputs of the first two AND-terms as inputs. If the input vector is neither 1001 nor 1101, then both terms will be Low, which causes the output of the Complement Array (/C) to be High. A third AND-term is used to AND state 24 and /C together to set the registers to state 03. The State Diagram is translated into SNAP syntax as shown in Figure 6b, where all vectors are in square brackets and the Complement Array is represented by the ELSE statement. The State diagram Figure 6a can also be expressed in the format of a program table as shown in Figure 6c. The complement array may be used to exit from different present states to different next states. It can be used many times in one state machine design as shown in Figures 7a, b, and c where the state diagram is implemented using the SNAP state equation syntax and a H/L format representation.



a. State Diagram

```
WHILE [23]
    IF [1001] THEN [24] WITH [OUT1]
    IF [1101] THEN [25] WITH [OUT2]
    ELSE [03] WITH [OUT9]
```

b. SNAP State Equation Syntax

		AND												OPTION (P/E) OR											
TERM	C	INPUT												PRESENT STATE								NEXT STATE								OUTPUT			
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0
00	A	H	L	L	H	–	–	–	–	–	–	–	–	H	L	L	L	H	H	–	–	–		H	L	L	H	L	L	–	–	–		L	L	L	H
01	A	H	H	L	H	–	–	–	–	–	–	–	–	H	L	L	L	H	H	–	–	–		H	L	L	H	L	H	–	–	–		L	L	H	L
02	•	–	–	–	–	–	–	–	–	–	–	–	–	H	L	L	L	H	H	–	–	–		L	L	L	L	H	H	–	–	–		H	L	L	H
03																																					
04																																					
05																																					
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																	16	15	14	13	11	10	9	8
NAME		IN3	IN2	IN1	IN0																													OP3	OP2	OP1	OP0

COMPLEMENT ARRAY

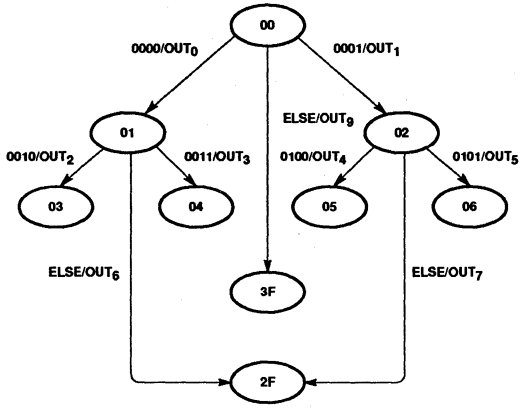
PRESENT STATE=23(HEX)

NEXT STATE: 24(HEX)
25(HEX)
03(HEX)

OUT1=0001
OUT2=0010
OUTG=1001

c. H/L Format

Figure 6. Application of Complement Array



a. State Diagram

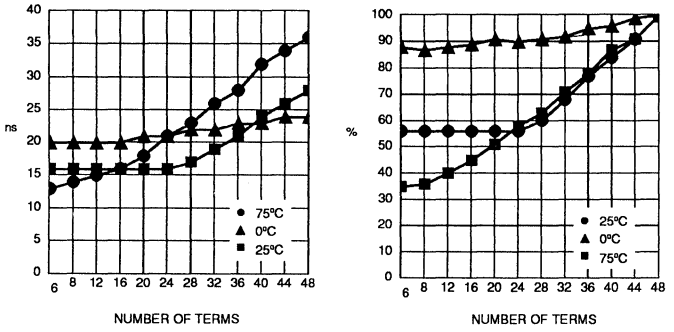
```
WHILE [00]
    IF [0000] THEN [01] WITH [OUT0]
    IF [0001] THEN [02] WITH [OUT1]
    ELSE [3F] WITH [OUT9]
WHILE [01]
    IF [0010] THEN [03] WITH [OUT2]
    IF [0011] THEN [04] WITH [OUT3]
    ELSE [2F] WITH [OUT6]
WHILE [02]
    IF [0100] THEN [05] WITH [OUT4]
    IF [0101] THEN [06] WITH [OUT5]
    ELSE [2F] WITH [OUT7]
```

b. SNAP State Equation

	AND																				OPTION (PR/OE) OR													
TERM	C	INPUT											PRESENT STATE									NEXT STATE							OUTPUT						
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0
00	A	L	L	L	L	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	L
01	A	L	L	L	H	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H
02	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	L	L	–	–	–	–	H	H	H	H	H	H	–	–	–	–	H	L	L	H
03																																					
04	A	L	L	H	L	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	L	H	H	–	–	–	–	L	L	H	L
05	A	L	L	H	H	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	L	L	L	H	L	L	–	–	–	–	L	L	H	H
06	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	L	H	–	–	–	–	H	L	H	H	H	H	–	–	–	–	L	H	H	L
07																																					
08	A	L	H	L	L	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H	L	H	–	–	–	–	L	H	L	L
09	A	L	H	L	H	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	L	L	L	H	H	L	–	–	–	–	L	H	L	H
10	•	–	–	–	–	–	–	–	–	–	–	–	–	L	L	L	L	H	L	–	–	–	–	H	L	H	H	H	H	–	–	–	–	L	H	H	H
11																																					
PIN NO.		18	19	20	21	22	23	2	3	4	5	6	7																16	15	14	13	11	10	9	8
NAME		IN3	IN2	IN1	IN0																												OP3	OP2	OP1	OP0

c. PLS168 Programming Table

Figure 7. Applications of Complement Array



a. PLS168 Setup Time vs. P-Term Loading

b. PLS168 Setup Time vs. P-Term Loading

Figure 8

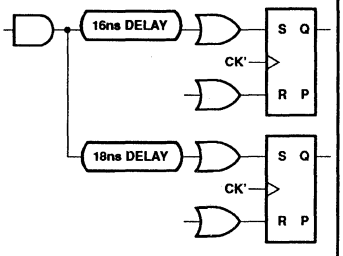


Figure 9. Difference in Propagation Delay Due to Different P-Term Loading

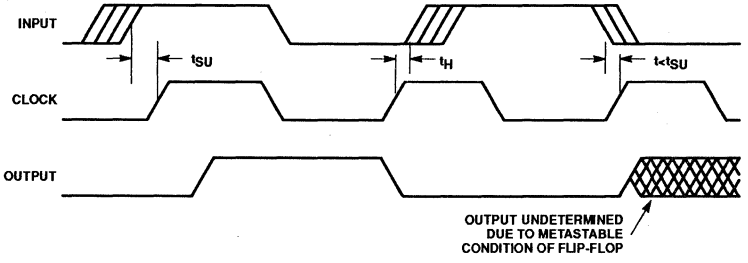


Figure 10. Change of t_{CKO} Due to Metastable Condition of Flip-Flop

Optional Preset/Output Enable

The PR/OE pin provides the user with the option of either using that pin to control the 3-State output buffers of the Output Registers, or have that pin to asynchronously preset all registers to High. The purpose of the preset function is to provide the system a way to set the PLS168 to a known state, all Highs. The output enable function are sometimes used where the state machine is connected to a bus which is shared by other output circuits. It is also used during power-up sequence to keep the PLS168 from sending power glitches to other circuits which it drives. By programming the PR/OE pin to control the 3-State output buffers, the preset function is permanently disabled. By programming the PR/OE pin to control the asynchronous preset of the registers, the output buffers are permanently enabled. While using the preset function to asynchronously preset the register, if a rising edge of the clock occurs while the preset input is High, the registers will remain preset. Normal flip-flop operation will resume only after the preset input is Low and the rising edge of the next clock. Setting the registers to a predefined pattern other than all Highs may be accomplished by using a dedicated p-term, which is activated by an input pin which will also inhibit all other p-terms which are being used. The inhibiting of other p-terms eliminates the problem of undetermined state of an RS flip-flop caused by having Highs on both R and S inputs.

Diagnostic Features

In debugging a state machine, sometimes it is necessary to know what is the content of the state register. The buried State Register may be read by applying +10V on I_0, which will cause the contents of register bits P_4 to P_5, P_6 to P_9 to be displayed on output pins F_2 to F_3 and P_0 and P_3 respectively. While the device can handle the +10V on pin I_0, prolonged and continuous use will cause the chip to heat up since more power is being dissipated at +10V. To facilitate more expedient functional tests, synchronous preset vectors as described above may be used to set the State Register to different states without having to go through the entire sequence.

Timing Requirements

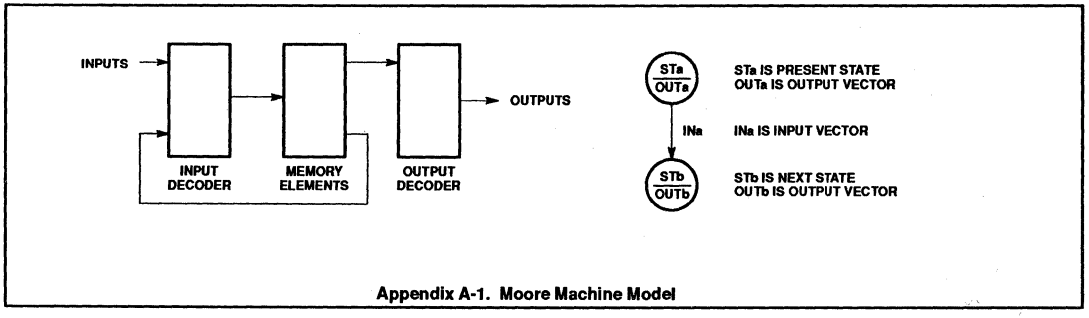
Since the PLS168 is intended to be a synchronous finite state machine, the inputs are expected to be synchronous to the clock and set-up and hold time requirements are expected to be met. In general, the set-up time requirement is measured at its worst case as having the entire AND-array connected to the OR-term being measured and there is only one active AND-term to drive the entire line. The set-up time decreases from there as less p-terms are used. This is due to the capacitance of the unused AND-terms being removed from the line. Figure 8a shows the typical set-up time requirement of a PLS168A device. Figure 8b shows the normalized set-up time as a percentage of the worst case, which is with 48 terms connected. In a typical state machine design, some flip-flops will change states more frequently than others. Those that change more frequently will have more p-term loading on its OR gates than those that change states less frequently. The different loadings on the OR-terms cause different delay on the inputs of the flip-flops as shown in Figure 9. If an input fails to meet the set-up time specification, it is possible that the resultant of the input change gets to one set of flip-flops before the rising edge of the clock while it gets to other flip-flops during or after the clock's rising edge. The result is that some flip-flops have changed states and some have not, or some get into metastable condition as shown in Figure 10. The state machine is now either out of sequence or is in an undefined state. This problem often occurs with asynchronous inputs which is generated totally independent of the clock on the system. A common remedy for the problem of asynchronous inputs is to use latches or flip-flops to catch the input and then synchronously feed it to the state machine. This minimizes the problem with the different propagation delays due to different p-term loading. But there is still a finite probability that the external latches or flip-flops will get into metastable condition, which may be propagated into the state machine. Nevertheless, the window for the flip-flops in state machine to get into undefined states or metastable condition is narrowed by a great extent.

APPENDIX A

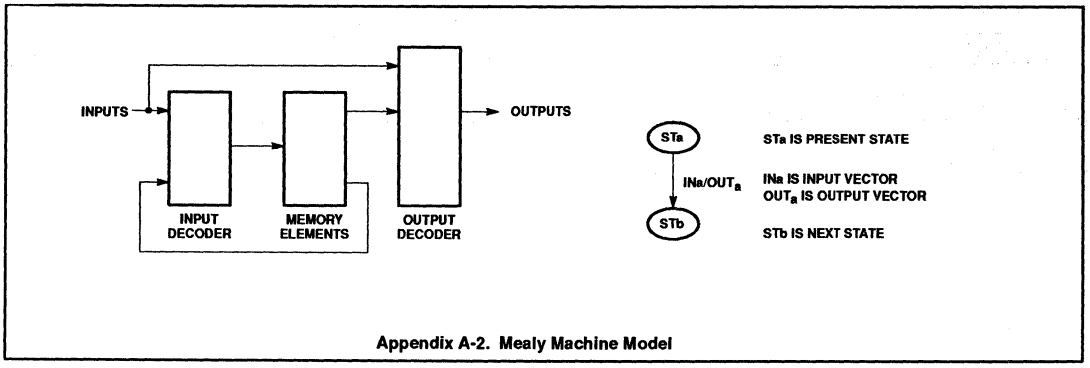
INTRODUCTION TO STATE MACHINE

A state machine is a synchronous sequential circuit which interprets inputs and generates outputs in accordance with a predetermined logic sequence. It is analogous to running a computer program with a computer. The state machine, with its sequence coded in hardware, can run much faster than a computer running the sequence in software. Therefore, it is often used in controller applications where speed is important.

Generally, state machines may be classified as Mealy or Moore machines as shown in Figures 1a and 1b. The fundamental difference of the two types are: the output of a Moore machine is a dependent of only the state of the memory elements whereas the output of a Mealy machine is a dependent of both the state of the memory elements and the inputs to the state machine. The figures also show graphic representations of the logic sequence in the form of state diagram in which the bubbles represent state vectors, and the arrows represent transitions from present states to next states.



Appendix A-1. Moore Machine Model



Appendix A-2. Mealy Machine Model

APPENDIX B

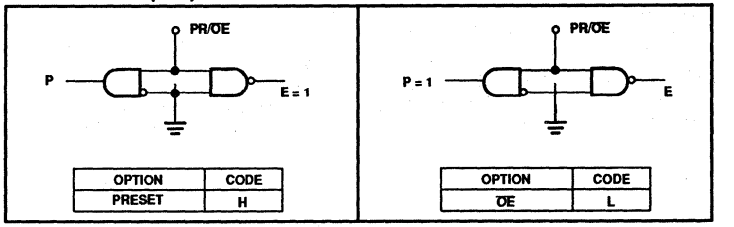
LOGIC PROGRAMMING

The PLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term T_n, is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

PRESET/$\overline{OE}$ – (P/E)



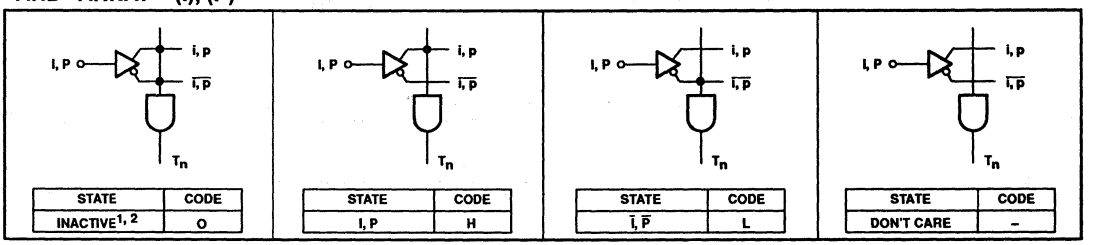
OPTION	CODE
PRESET	H

OPTION	CODE
$\overline{OE}$	L

PROGRAMMING:

The PLS168/A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

"AND" ARRAY – (I), (P)



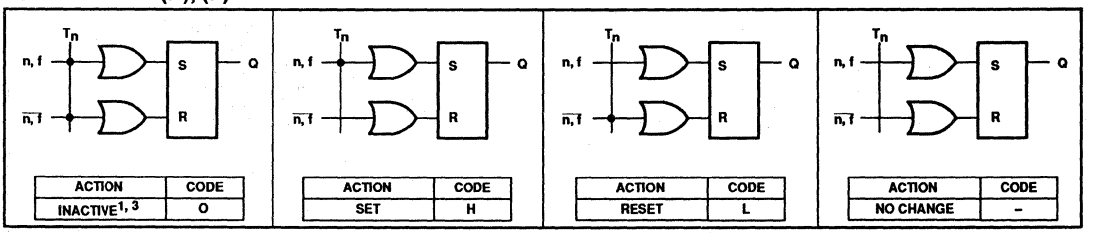
STATE	CODE
INACTIVE[1, 2]	O

STATE	CODE
I, P	H

STATE	CODE
$\overline{I, P}$	L

STATE	CODE
DON'T CARE	–

"OR" ARRAY – (N), (F)



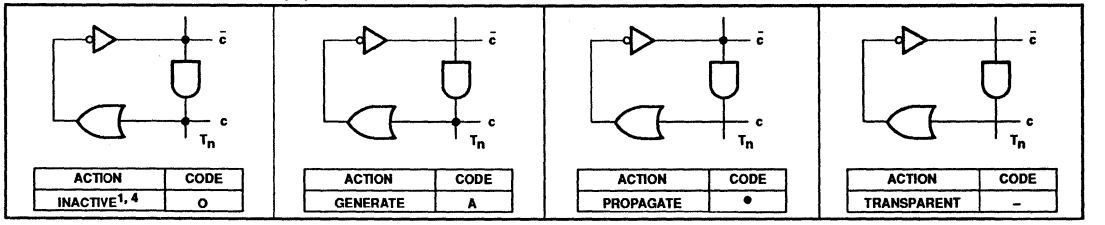
ACTION	CODE
INACTIVE[1, 3]	O

ACTION	CODE
SET	H

ACTION	CODE
RESET	L

ACTION	CODE
NO CHANGE	–

"COMPLEMENT" ARRAY – (C)



ACTION	CODE
INACTIVE[1, 4]	O

ACTION	CODE
GENERATE	A

ACTION	CODE
PROPAGATE	•

ACTION	CODE
TRANSPARENT	–

NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate T_n will be unconditionally inhibited if both the true and complement of any input (I or P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates T_n (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates T_n.

Alarm Controller

AN047

A Programmable Alarm System – PLS168

A basic alarm controller can be considered as a black box with several inputs and several outputs (Figure 1). Some inputs are used for detection and others for control. Detect inputs are driven from a variety of alarm transducers such as reed switches, smoke detectors, pressure mats, etc. An *ARM* input switches the system into a state which allows detection of the various alarm conditions and a *RESET* input is used to reset the system after an alarm has been triggered and dealt with or on re-entering the protected area. Outputs from the system include a sounder, a beacon and status indicators.

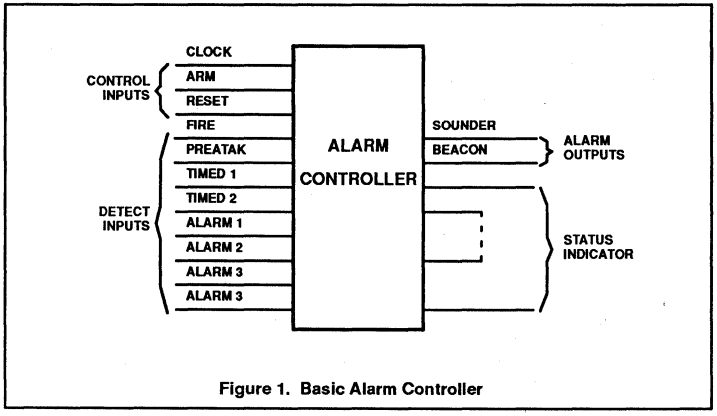


Figure 1. Basic Alarm Controller

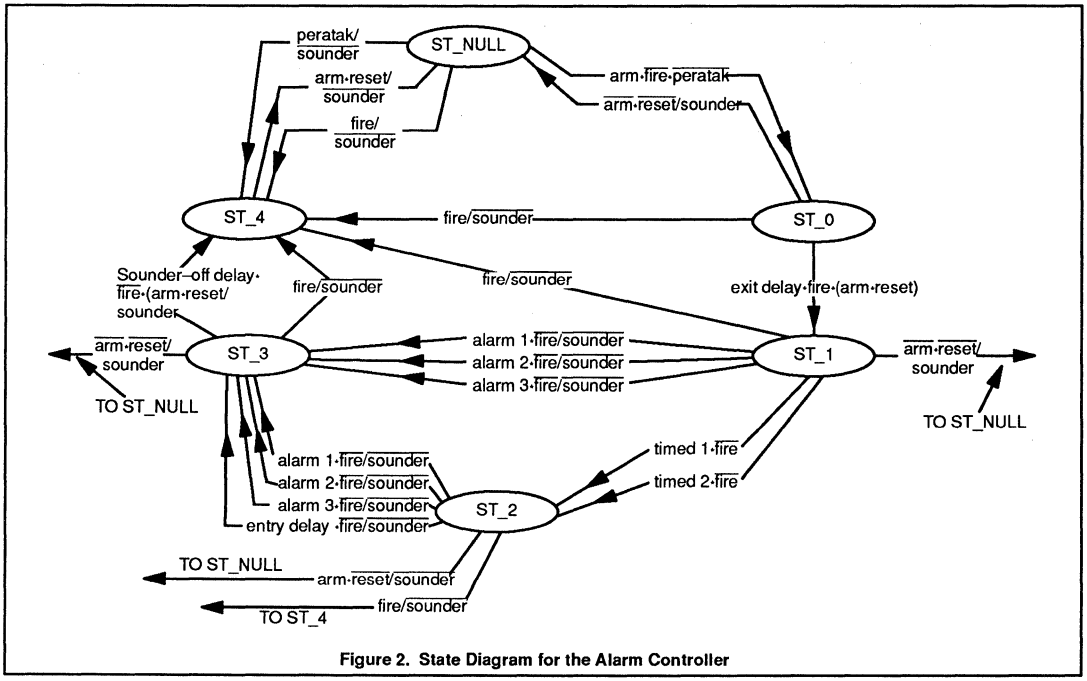


Figure 2. State Diagram for the Alarm Controller

Detect inputs can be divided into timed, untimed, fire and personal attack inputs. Timed circuits allow entry/exit delay circuits for front and rear doors, to delay operation of the alarm for approximately 16 seconds. Untimed circuits cause the alarm to operate immediately when an alarm condition occurs. These would be used to protect unusual means of entry, such as windows. Both the timed and untimed circuits should operate only if the system is armed.

The personal attack circuit is a special case untimed circuit and should operate only when the system is disarmed. The fire-detect circuit is again a special case untimed circuit and should operate regardless of whether the system is armed or not.

Outputs from the controller drive an external sounder and beacon. After 128 seconds, the sounder should turn off if the alarm has been triggered by either a timed or general untimed circuit. However, when a fire or personal attack triggers the system, the sounder should not turn off until the system is reset and the alarm condition removed.

State Machine Implementation

This design is best implemented as a state machine. The state diagram is derived from the verbal system description. Please note from Figure 2 the controller can be in one of six possible states. Examine the transitions from *ST_NULL* as an example. If a personal attack or fire condition occurs while in this state, a transition to *ST_1* takes place as indicated by the arrows on the diagram. Also at this time the sounder and beacon are activated, thus giving the alarm. If the fire and personal attack conditions have not occurred and the *ARM* SWITCH is set, then a transition to *ST_0* takes place.

Similarly, other arrows on the state diagram represent transitions between other states when specified input conditions occur. Output parameters are shown to the right of the slash line. Where there are no output parameters specified in a transition term, this indicates that no output changes are desired during this transition. That is, an output will hold its present value until told to change.

PLD Implementation

Having defined the desired system operation it is now time to select the required device to implement the desired system function from the PLD Data Manual. In this case, the device selected is the PLS168. Figure 3 shows the pinning information for the alarm controller. A 10-bit counter within the controller produces the entry/exit and sounder turn-off delays since this makes more efficient use of the PLD facilities than implementing the delays as part of the state machine. This counter uses seven internal registers with feedback and three without. For those registers without feedback, external wiring feeds their outputs back into the device to complete the 20-bit counter. Pins five to ten are used for this purpose. Output T7 also forms part of the counter.

Three other registers form the state registers and are labeled SR0, SR1 and BEACON. State vectors for these registers have to be chosen with care to ensure that the beacon output is activated at the correct time. Other inputs and outputs are as already discussed. Note that the PR/OE pin is not used. SNAP defaults its use to a register PRESET function. This pin should be tied to ground in the final circuit.

The EQN file of SNAP is separated into sections. First, in the @PINLIST section all of the signal names connected directly to the pins and their function is listed. If a signal name is used later in the file and not listed in the @PINLIST section, that signal is assumed to represent an internal node. The @PINLIST and @LOGIC EQUATIONS sections of the EQN file are shown in Table 1. The remaining state machine portion of the EQN file is shown in Table 2. Register SR0 halts and clears the counter while the controller is in certain states. This needs to be considered when defining the state vectors.

TABLE 1. SNAP EQUATIONS

```
"-----------------------------------------"
"           ALARM CONTROLLER              "
"-----------------------------------------"
@PINLIST
clock i;
arm   i;
reset i;
peratak i;
t8in  i;
t9in  i;
t10in i;
alarm3 i;
alarm2 i;
alarm1 i;
timed2 i;
timed1 i;
fire   i;

t10     o;
t9      o;
t8      o;
t7o     o;
sounder o;
beacon  o;
sr1     o;
sr0     o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
 "ten-bit counter for delay"

t1.s = /t1*/sr0;
t1.r =  t1*/sr0 + sr0;
t1.clk = clock;
t2.s =  t1*/t2*/sr0;
t2.r =  t1* t2*/sr0 + sr0;
t2.clk = clock;
t3.s =  t1* t2*/t3*/sr0;
t3.r =  t1* t2* t3*/sr0 + sr0;
t3.clk = clock;
t4.s =  t1* t2* t3*/t4*/sr0;
t4.r =  t1* t2* t3* t4*/sr0 + sr0;
t4.clk = clock;
t5.s =  t1* t2* t3* t4*/t5*/sr0;
t5.r =  t1* t2* t3* t4* t5*/sr0 + sr0;
t5.clk = clock;
t6.s =  t1* t2* t3* t4* t5*/t6*/sr0;
t6.r =  t1* t2* t3* t4* t5* t6*/sr0 + sr0;
t6.clk = clock;
t7.s =  t1* t2* t3* t4* t5* t6*/t7*/sr0;
t7.r =  t1* t2* t3* t4* t5* t6* t7*/sr0 + sr0;
t7.clk = clock;
t7o=t7;
t8.s =  t1* t2* t3* t4* t5* t6* t7*/t8in*/sr0;
t8.r =  t1* t2* t3* t4* t5* t6* t7* t8in*/sr0 + sr0;
t8.clk = clock;
t9.s =  t1* t2* t3* t4* t5* t6* t7* t8in*/t9in*/sr0;
t9.r =  t1* t2* t3* t4* t5* t6* t7* t8in* t9in*/sr0 + sr0;
t9.clk = clock;
t10.s=  t1* t2* t3* t4* t5* t6* t7* t8in* t9in*/t10in*/sr0;
t10.r=  t1* t2* t3* t4* t5* t6* t7* t8in* t9in* t10in*/sr0 + sr0;
t10.clk = clock;

sr0.clk = clock;
sr1.clk = clock;
beacon.clk = clock;
sounder.clk = clock;
```

(EQN file continued in Table 2)

State Equation Entry

The state equation entry portion of the EQN file uses a state-transition language, parameters of which are taken directly from the state diagram. Information is entered into this file in a free format. The only points to remember are that the square brackets should be used throughout to define the state registers and transitions, semicolons should be used to mark the end of vector definition. State vectors can be defined in the state equation entry file as shown in Table 2. State vectors are simply a means of labeling an arrangement of state registers which can be used later to define state transitions. Because we are using the *BEACON* output register as a state register also and SR0 is being used to halt and clear the 10-bit counter, particular care must be taken in defining the state vectors in this instance.

From the state diagram, the counter must begin counting during states *ST_0, ST_2* and *ST_3* and it must be cleared during states *ST_1, ST_4* and *ST_NULL*. State *ST_NULL* represents the power-up state of the PLS168 in which all register outputs are at logic one. Thus the inactive state of the counter is defined as being when SR0 is at logic one, therefore, SR0 must be at this level during states *ST_1* and *ST_4* and at logic zero during other states. The alarm beacon is considered to be active by an active-low signal and must be activated during states *ST_3* and *ST_4*. Register SR1 must therefore be chosen to ensure mutual exclusivity between state vectors. Input and output vectors can be defined in the same manner in terms of input and output label names. In this case, however, the label names are used directly. State transitions can now be derived directly from the state diagram. Entry/exit and sounder turn-off delay times are represented as a decoding of the 10-bit counter states. Thus to get the desired 16 second entry/exit delay. t7 must be decoded and to achieve the 128 second sounder turn-off delay t10in must be decoded.

TABLE 2. SNAP EQUATIONS

```
@INPUT VECTORS
@OUTPUT VECTORS
[sounder]srff
s_on  = 0b;
s_off = 1b;

@STATE VECTORS
[sr0, sr1, beacon]srff
st_null = 111b;
st_0    = 001b;
st_1    = 101b;
st_2    = 011b;
st_3    = 010b;
st_4    = 100b;

@TRANSITIONS

while [st_null]
 if [arm*/fire*/peratak]   then [st_0]
 if [peratak]              then [st_4]    with [s_on]
 if [fire]                 then [st_4]    with [s_on]

while [st_0]
 if [t7*/fire*(arm+reset)] then [st_1]
 if [/arm*/reset]          then [st_null] with [s_off]
 if [fire]                 then [st_4]    with [s_on]

while [st_1]
 if [timed1*/fire]         then [st_2]
 if [timed2*/fire]         then [st_2]
 if [alarm1*/fire]         then [st_3]    with [s_on]
 if [alarm2*/fire]         then [st_3]    with [s_on]
 if [alarm3*/fire]         then [st_3]    with [s_on]
 if [/arm*/reset]          then [st_null] with [s_off]
 if [fire]                 then [st_4]    with [s_on]

while [st_2]
 if [t7*/fire]             then [st_3]   with  [s_on]
 if [alarm1*/fire]         then [st_3]   with  [s_on]
 if [alarm2*/fire]         then [st_3]   with  [s_on]
 if [alarm3*/fire]         then [st_3]   with  [s_on]
 if [/arm*/reset]          then [st_null] with [s_off]
 if [fire]                 then [st_4]    with [s_on]

while [st_3]
 if [t10in*/fire*(arm+reset)] then [st_4] with [s_off]
 if [/arm*/reset]          then [st_null] with [s_off]
 if [fire]                 then [st_4]    with [s_on]

while [st_4]
 if [/arm*/reset]          then [st_null] with [s_off]
```

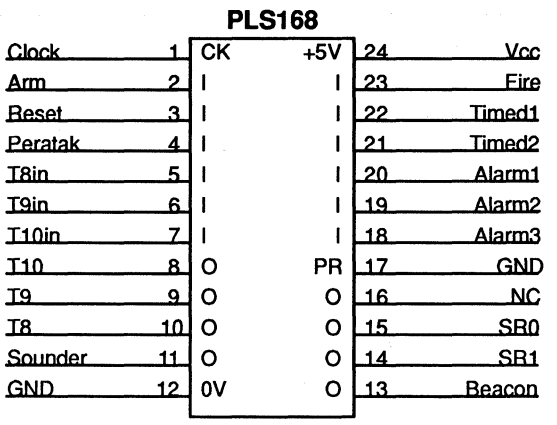


Figure 3. Pinning Information for the Alarm Controller

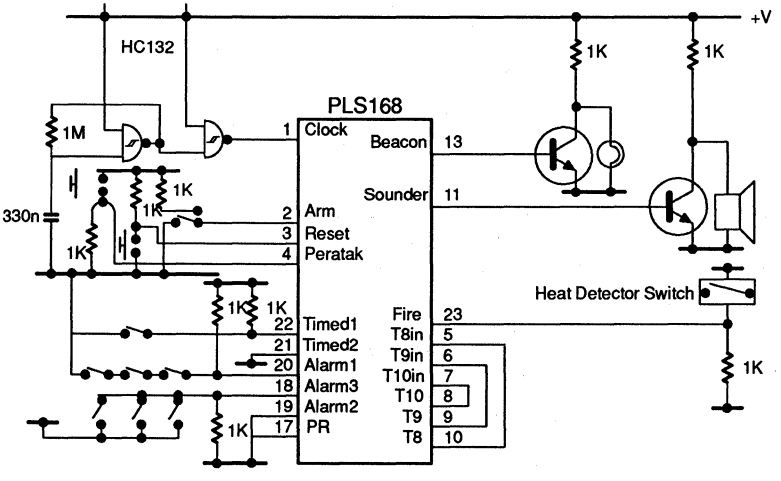


Figure 4. Alarm System based on the PLS168

With the system fully defined, simply assemble the design information using SNAP.

Functioning of the device can be verified with the SNAP simulator, which can also be used to check A.C. timings before downloading the pattern to a device programmer.

Programmability

The PLS168 device could now be used as the controller of an alarm system. As it stands, the device assumes that all the alarm inputs indicate an alarm condition when in the high state, logic one, and that the alarms are activated when the alarm outputs are active low (i.e., at logic zero).

Should an alarm input transducer be used which indicates an alarm condition as a low state, this can be catered for by altering the EQN file. For example, consider a smoke detector which outputs logic zero on detection of an alarm condition and assume that this transducer is driving the "fire" input of the device. By changing all references to 'fire' in the EQN file to '/fire' and all instances of '/fire' to 'fire' then the activation of the alarms will occur when logic zero is applied to this input and not when logic one is applied, as in the original case.

Polarity of the output signals cannot be altered as easily, as the device will always power-up with the outputs at logic one. This should not prove to be a problem since the outputs simply drive output transistors and

these can be used to produce the correct polarity signal for the beacon and sounder.

System Implementation

Figure 2 shows a typical alarm system based on this device. The system clock is produced by a relaxation oscillator built from 74HC132 Schmitt Triggers. Values of R_1 and C_1 shown result in a frequency of approximately 4Hz which will provide the desired entry/exit and sounder turn-off delays. These delays can be modified either by changing the external oscillator circuit or by decoding a different internal counter state. For example, to increase the entry/exit delay change all references to t7 in the EQN file to t8. Both normally-closed and normally-open loop implementations are shown. Due to the distances involved in an alarm system, the open-loop configuration may cause problems, being driven by the positive supply. to avoid this problem, input-detect polarity of the open-loop circuit can be changed by altering the EQN file.

Status indication can be provided by connecting LEDs as in Figure 5. When the reset button is pressed, any LED being lit will indicate an alarm condition for that input. This will not reset the alarm system unless the arm switch is off.

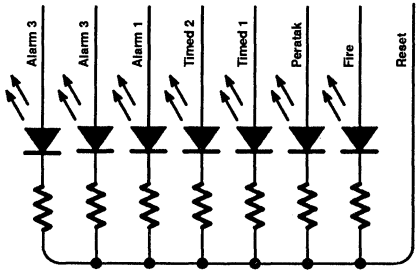


Figure 5. Status LEDs Connected to the alarm controller as shown provide status information

High-speed 12-bit tracking A/D converter using PLS179 AN028

INTRODUCTION

The general technique underlying the operation of this A/D converter is illustrated by the functional block diagram in Figure 1. The system consists of a D/A converter, a comparator circuit, and digital logic circuitry. The digital logic circuitry outputs a digital value which is converted to analog by the D/A converter.

The comparator senses when the output is greater or less than the input and causes the digital circuit to decrement or increment its digital output respectively. The initial conversion is completed in 13 clock cycles. If tracking mode is used, the A/D converter then tracks the input voltage as it changes by incrementing or decrementing 1-LSB per clock. The tracking function makes it possible to make an A/D conversion in one clock cycle if the input changes less than the value of 1-LSB per clock period. The conversion may be halted and the digital output, as well as the converted analog output from DAC, will hold their output constant indefinitely. This feature works well as sample-and-hold since its output voltage will not decay over time whereas the output of an analog sample/hold will decay due to charge leakages.

In order to avoid the violation of setup time by the output of the comparator, its output is latched. There is a built-in 2-phase clock in U2 which may be used to drive the logic circuitry and the latch of the comparator (see Philips Semiconductors NE5105 data sheet for details on output latches of voltage comparators).

The analog input voltage may be sampled and held by an analog sample/hold circuit to keep the input to the ADC from changing. The DONE output may be used to control the sample-and-hold if needed.

This paper discusses only the digital circuit which contains the SAR and the Up/Down Counter. The analog circuits are not within the scope of this paper.

SAR

Two PLS179s are connected together to form a 12-bit shift register and up/down counter. The schematic diagram of the A/D converter is shown in Figure 2. U2 contains bits 0 to 4 and U1 contains Bits 5 to 11. Interconnects are made as shown in the diagram. The digital output to the DAC is in natural binary format (e.g. 0000 0000 0000 equal zero, and 1111 1111 1111 is full scale or 4095). After the nST input becomes 0, at the rising edge of the next clock, the SAR is initialized to half-scale (1000 0000 0000) and the DONE flip-flop is reset to output 0 which causes the open-collector output nDone_OC to become high impedance.

The digital output is converted by the DAC and is compared to the analog input voltage by the comparator. If the digital output is greater than the analog input, the SAR shifts the 1 to next MSB on the right. The content of the SAR becomes (0100 0000 0000). If the digital output is still greater than the input, the SAR shifts right one bit again. The content of the SAR then becomes (0010 0000 0000). The shifting of 1 to the next MSB in equivalent to reducing by half the value of the bit under consideration. If the output is still too large, the SAR reduces it by half again by shifting to the right one more time. The SAR keeps shifting to the right until the digital output is less than the input. When the output is less than the input, the SAR adds one bit to the next MSB while keeping all the higher order bits unchanged. For example, if the current output is 0001 0000 0000 and the output is less than the input, the SAR adds one bit to the right at the next clock. The output becomes 0001 1000 0000. The output is again compared to the input. If the addition of that one bit is too much, it will be shifted to the right until the output becomes less than the input. When that happens, that SAR will again add one bit to the right. The algorithm of the SAR may be summarized as the following: If the output is greater than the input, shift to the right; otherwise add one bit to the right. This process continues until all 12 bits have been operated on. The last bit (Bit 0) is always changed from 0 to 1, which is used as the condition to set DONE to 1 which, in turn, sets open-collector output, /nDONE_OC, to 0.

UP/DOWN COUNTER

After DONE becomes 1, if /nST and nHOLD are 1 and nTRACK is 0, the SAR turns into a 12-bit up/down counter. If the analog input voltage increases, the counter will increment by 1 at every clock until it matches the input. If the input decreases, the counter will decrement by 1. When nHOLD becomes 0, the counter is inhibited and the output is held indefinitely. The counters consist of 12 toggle flip-flops and 2 p-terms per flip-flop for directional control. The counter will operate only after the approximation cycle is completed and DONE is 1.

Since the nST and nHOLD inputs may be asynchronous with the clock, in order to minimize the possibilities of having a metastable condition from happening, these inputs close-up are latched by flip-flops nSTART of U1 and nHLD of U2 respectively. Once they are latched, subsequent operation begins at the rising-edge of the next clock. The output of the comparator may be latched to prevent setup time violation. (Philips Semiconductors NE5105 is a high-speed comparator with an output latch. External latch may be used with other comparators.)

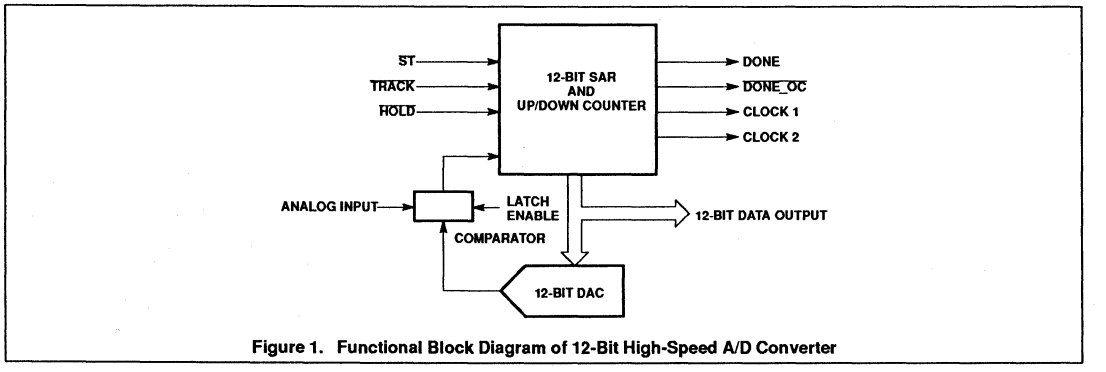


Figure 1. Functional Block Diagram of 12-Bit High-Speed A/D Converter

High-speed 12-bit tracking A/D converter using PLS179 AN028

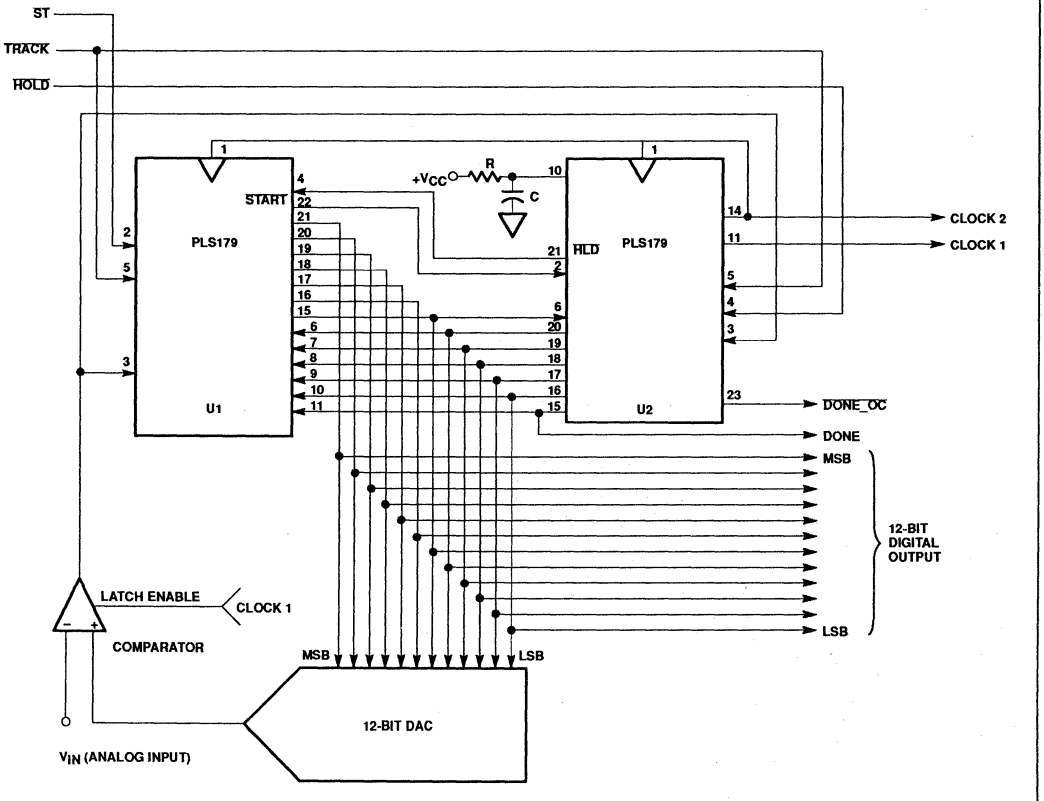


Figure 2. Schematic Diagram of 12-Bit High-Speed A/D Converter

High-speed 12-bit tracking A/D converter using PLS179 AN028

CLOCKS

U2 generates an optional 2-phase clock which may be used to control the latch of the comparator. The two clocks are basically 180° out of phase and CLOCK2 has an additional 25ns propagation delay behind CLOCK1. CLOCK2 is used to drive the clock-inputs of the PLS179 devices.

The clock frequency is controlled by R and C. Those who want to use the built-in clock should experiment with RC time constants for the best value. It is recommended that the capacitance should be less than 1000pF for best results.

DONE AND nDONE_OC

The output DONE is reset to 0 when nST is 0. It remains 0 until the approximation cycle is completed. After the least significant bit becomes 1, the DONE bit becomes 1 at the next clock. It remains 1 until it is reset again by input nST.

The nDONE_OC output is configured to emulate an open-collector output. The output is programmed to have a logic 0. When DONE is 0, the 3-State output buffer is set to Hi-Z condition. As soon as DONE equals 1, the 3-State buffer is enabled and nDONE_OC becomes 0.

In the initial phase of A/D conversion, 13 clock cycles are required. It is essential that the input voltage to the comparator remains unchanged while the SAR is converting. It may be necessary to have a sample/hold at the front end. The DONE output may be used to control the analog sample/hold circuit.

INPUT LATCHES

Flip-flop nSTART and 2 p-terms in U1 are configured as a non-inverting D flip-flop. The input, nST, and the output nSTART have the same polarities. Flip-flop nHLD and 2 p-terms in U2 also form a non-inverting D flip-flop. The output nHLD and the input nHOLD have the same polarities.

AMAZE IMPLEMENTATION

The implementation of the logic circuit using ABEL is as shown in the appendices. Two files (ADCS1.ABL and ADCS2.ABL) are generated for each PLS179s. State machine and Boolean equation design entries are used for both files.

The SAR circuit is designed as a state machine and the up/down counter, input latches, 2-phase clocks and the open-collector output are implemented by using Boolean equations. BIT0 to 4 are described in ADCS2 and BIT5 to 11 are in ADCS1. Toggle flip-flops are implemented by the JK flip-flops (when .J=.K). In order to combine two different design entries into one device, pin attribute assignment 'ISTYPE REG_JK' must be given to keep output registers type consistency (ABEL will default state machine to D-F/F).

Test vectors are also supplied with each design file to simulate SAR and up/down counter functions. Note that U2 will not start functioning until BIT5 and U1 becomes 1 and ripples into U2. When nST becomes 0,it clears all the state registers and next conversion cycle begins. The files are then compiled and simulated to produce the JEDEC files (ADCS1.JED and ADCS2.JED) and the simulation files (ADCS1.SIM and ADCS2.SIM).

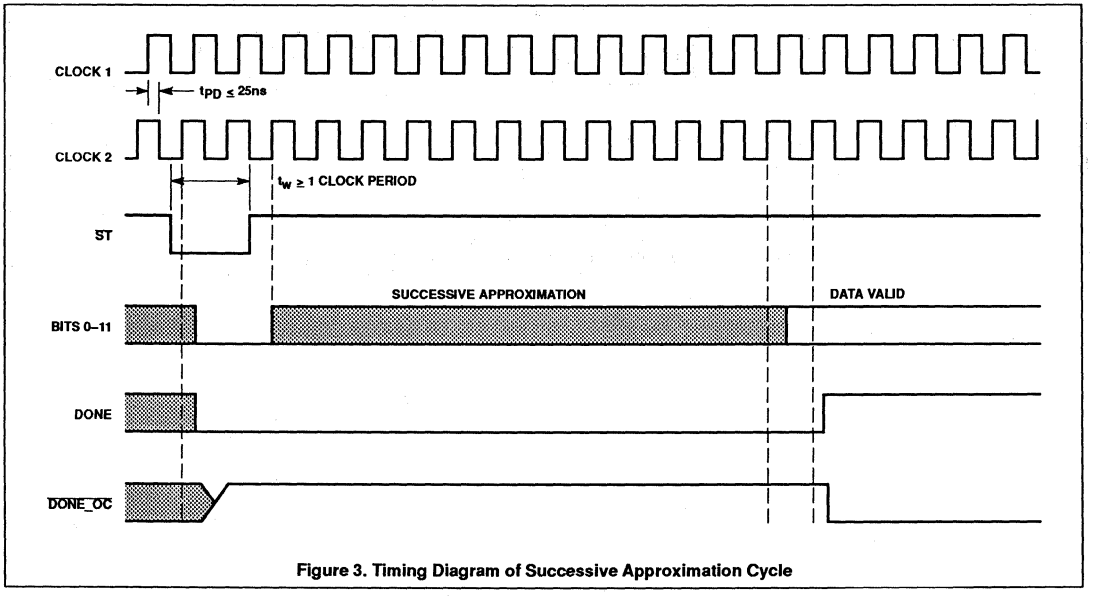


Figure 3. Timing Diagram of Successive Approximation Cycle

APPENDIX A: ABEL DESIGN FILE OF U1

```
MODULE  ADCS1;
TITLE   'SAR AND UPDOWN COUNTER FOR THE BIT5..BIT11
         PHILIPS SEMICONDUCTORS'

DECLARATIONS
        ADCS1 DEVICE 'F179';

"Signal names preceding with a 'n' means 'active low'.
"INPUTS
        CLOCK,nST,COMPARE,nHLD,nTRACK       PIN 1,2,3,4,5;
        BIT4,BIT3,BIT2,BIT1,BIT0,DONE       PIN 6,7,8,9,10,11;

"OUTPUTS
        BIT5,BIT6,BIT7,BIT8,BIT9        PIN 15,16,17,18,19  ISTYPE 'REG_JK';
        BIT10,BIT11,nSTART              PIN 20,21,22        ISTYPE 'REG_JK';

        H,L,CK,X = 1, 0, .C., .X.;

        "GREATER means digital output is greater than analog input, ...
        "LESS means digital output is less than analog inputs, ...
        GREATER = COMPARE&!BIT4&!BIT3&!BIT2&!BIT1&!BIT0&nST&!DONE;
        LESS =    !COMPARE&!BIT4&!BIT3&!BIT2&!BIT1&!BIT0&nST&!DONE;

        SREG = [nSTART,BIT11,BIT10,BIT9,BIT8,BIT7,BIT6,BIT5];

        BEGIN         = [X, X,X,X, X,X,X,X ];
        INIT          = [0, 0,0,0, 0,0,0,0 ];
        HALFSCALE     = [1, 1,0,0, 0,0,0,0 ];
        ST2048        = [1, 1,0,0, 0,0,0,0 ];
        ST1024        = [1, X,1,0, 0,0,0,0 ];
        ST512         = [1, X,X,1, 0,0,0,0 ];
        ST256         = [1, X,X,X, 1,0,0,0 ];
        ST128         = [1, X,X,X, X,1,0,0 ];
        ST64          = [1, X,X,X, X,X,1,0 ];
        ST32          = [1, X,X,X, X,X,X,1 ];

        AD1024        = [X, X,1,X, X,X,X,X ];
        AD512         = [X, X,X,1, X,X,X,X ];
        AD256         = [X, X,X,X, 1,X,X,X ];
        AD128         = [X, X,X,X, X,1,X,X ];
        AD64          = [X, X,X,X, X,X,1,X ];
        AD32          = [X, X,X,X, X,X,X,1 ];
        AD16          = [1, X,X,X, X,X,X,X ];

        SH1024        = [X, 0,1,X, X,X,X,X ];
        SH512         = [X, X,0,1, X,X,X,X ];
        SH256         = [X, X,X,0, 1,X,X,X ];
        SH128         = [X, X,X,X, 0,1,X,X ];
        SH64          = [X, X,X,X, X,0,1,X ];
        SH32          = [X, X,X,X, X,X,0,1 ];
        SH16          = [X, X,X,X, X,X,X,0 ];
```

APPENDIX A: ABEL DESIGN FILE OF U1 (Continued)

```
EQUATIONS

SREG.CLK = CLOCK;

"Non-Inverting Input Latch: nSTART = nST
nSTART.J = !nST;
nSTART.K = nST;

"UPDOWN COUNTER"
BIT5.J =  !nSTART.Q & !nTRACK & DONE & nHLD& COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4
          #
          !nSTART.Q & !nTRACK & DONE & nHLD& !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4;

BIT5.K =  !nSTART.Q & !nTRACK & DONE & nHLD& COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4
          #
          !nSTART.Q & !nTRACK & DONE & nHLD& !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4;

BIT6.J =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q;

BIT6.K =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q;

BIT7.J =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q & BIT6.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q & !BIT6.Q;

BIT7.K =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q & BIT6.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q & !BIT6.Q;

BIT8.J =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q & BIT6.Q & BIT7.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q & !BIT6.Q & !BIT7.Q;

BIT8.K =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0 & !BIT1 & !BIT2 & !BIT3 & !BIT4 & BIT5.Q & BIT6.Q & BIT7.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1 & BIT2 & BIT3 & BIT4 & !BIT5.Q & !BIT6.Q & !BIT7.Q;
```

APPENDIX A: ABEL DESIGN FILE OF U1 (Continued)

```
BIT9.J =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q;

BIT9.K =  !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0 & BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q;

BIT10.J = !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q& BIT9.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0& BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q& !BIT9.Q;

BIT10.K = !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q& BIT9.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0& BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q& !BIT9.Q;

BIT11.J = !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q& BIT9.Q& BIT10.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0& BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q& !BIT9.Q& !BIT10.Q;

BIT11.K = !nSTART.Q & !nTRACK & DONE & nHLD & COMPARE &
          !BIT0& !BIT1& !BIT2& !BIT3& !BIT4& BIT5.Q& BIT6.Q& BIT7.Q& BIT8.Q& BIT9.Q& BIT10.Q
          #
          !nSTART.Q & !nTRACK & DONE & nHLD & !COMPARE &
          BIT0& BIT1& BIT2& BIT3& BIT4& !BIT5.Q& !BIT6.Q& !BIT7.Q& !BIT8.Q& !BIT9.Q& !BIT10.Q;
```

APPENDIX A: ABEL DESIGN FILE OF U1 (Continued)

```
STATE_DIAGRAM SREG;

STATE BEGIN:
        if !nST then INIT;

STATE INIT:
        if nST then HALFSCALE;

STATE ST2048:
      if GREATER  then  SH1024
      if LESS then AD1024;

STATE ST1024:
      if GREATER  then  SH512
      if LESS then AD512;

STATE ST512:
      if  GREATER  then  SH256
      if LESS then AD256;

STATE ST256:
      if GREATER  then  SH128
      if LESS then AD128;

STATE ST128:
      if  GREATER  then  SH64
      if LESS then AD64;

STATE ST64:
      if GREATER  then  SH32
      if LESS then AD32;

STATE ST32:
      if  GREATER  then SH16
      if LESS then AD16;
```

APPENDIX A: ABEL DESIGN FILE OF U1 (Continued)

```
TEST_VECTORS
([nST,nHLD,nTRACK,BIT4,BIT3,BIT2,BIT1,BIT0,DONE,CLOCK,COMPARE]
- [nSTART,BIT11,BIT10,BIT9,BIT8,BIT7,BIT6,BIT5])

 [1,1,1,0,0,0,0,0,0,0 ,0] -> [1,1,1,1,1,1,1,1]; "1  power-on reset
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,1,1,1,1,1,1,1]; "2
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,1,1,1,1,1,1,1]; "3
 [0,1,1,0,0,0,0,0,0,CK,0] -> [0,0,0,0,0,0,0,0]; "4  nST resets outputs to 0s
 [0,1,1,0,0,0,0,0,0,CK,0] -> [0,0,0,0,0,0,0,0]; "5
 [0,1,1,0,0,0,0,0,0,CK,0] -> [0,0,0,0,0,0,0,0]; "6
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,1,0,0,0,0,0,0]; "7  State machine goes to HALFSACLE
 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,1,0,0,0,0,0]; "8  and SAR begins
 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,0,1,0,0,0,0]; "9
 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,0,0,1,0,0,0]; "10

 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,0,0,0,1,0,0]; "11
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,0,0,0,0,1,1,0]; "12
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,0,0,0,0,1,1,1]; "13
 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,0,0,0,1,1,0]; "14 End of SAR of bit5-11
 [1,1,1,1,0,0,0,0,0,CK,1] -> [1,0,0,0,0,1,1,0]; "15 Outputs keep unchange
 [1,1,1,0,1,0,0,0,0,CK,1] -> [1,0,0,0,0,1,1,0]; "16
 [1,1,1,0,1,1,0,0,0,CK,0] -> [1,0,0,0,0,1,1,0]; "17
 [1,1,1,0,1,0,1,0,0,CK,0] -> [1,0,0,0,0,1,1,0]; "18
 [1,1,1,0,1,0,1,1,0,CK,0] -> [1,0,0,0,0,1,1,0]; "19
 [1,1,1,0,1,1,1,1,1,CK,0] -> [1,0,0,0,0,1,1,0]; "20 End of SAR of bit0-4

 [1,1,0,1,1,1,1,1,1,CK,1] -> [1,0,0,0,0,1,1,0]; "21 Tracking (up/down counter) started
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,0,1,1,1]; "22 up
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,0,0]; "23 up
 [1,1,0,1,1,1,1,0,1,CK,0] -> [1,0,0,0,1,0,0,0]; "24 unchange
 [1,1,0,1,1,1,1,0,1,CK,0] -> [1,0,0,0,1,0,0,0]; "25 unchange
 [1,1,0,1,1,1,1,0,1,CK,0] -> [1,0,0,0,1,0,0,0]; "26 unchange
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,0,1]; "27 up
 [1,1,0,0,0,0,0,0,1,CK,1] -> [1,0,0,0,1,0,0,0]; "28 down
 [1,1,0,0,0,0,0,0,1,CK,1] -> [1,0,0,0,0,1,1,1]; "29 down
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,0,0]; "30 up

 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,0,1]; "31 up
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,1,0]; "32 up
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,0,1,1]; "33 up
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,1,0,0]; "34 up
 [1,1,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,1,0,1]; "35 up
 [1,0,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,1,0,1]; "36 nHLD goes LOW, hold the output value
 [1,0,0,0,0,0,0,0,1,CK,1] -> [1,0,0,0,1,1,0,1]; "37
 [1,0,0,1,1,1,1,1,1,CK,0] -> [1,0,0,0,1,1,0,1]; "38
 [1,0,0,1,1,1,1,1,1,CK,1] -> [1,0,0,0,1,1,0,1]; "39
 [0,1,1,0,0,0,0,0,0,CK,0] -> [0,0,0,0,0,0,0,0]; "40 Next conversion cycle begins

 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,1,0,0,0,0,0,0]; "41 HALFSACLE
 [1,1,1,0,0,0,0,0,0,CK,1] -> [1,0,1,0,0,0,0,0]; "42 SAR begins
 [1,1,1,0,0,0,0,0,0,CK,0] -> [1,0,1,1,0,0,0,0]; "43

END ADCS1
```

APPENDIX B: ABEL DESIGN FILE OF U2

```
MODULE  ADCS2;
TITLE   'SAR AND UPDOWN COUNTER FOR THE BIT0..BIT4
        PHILIPS SEMICONDUCTORS'

DECLARATIONS
            ADCS2 DEVICE 'F179';

"INPUTS
       CLOCK,nSTART,COMPARE,nHOLD,nTRACK    PIN 1,2,3,4,5;
       BIT5,RC,nDONE_OC                     PIN 6,10,23;

"OUTPUTS
       CLOCK1,CLOCK2                    PIN 11,14;
       DONE,BIT0,BIT1,BIT2              PIN 15,16,17,18 ISTYPE 'REG_JK';
       BIT3,BIT4,nHLD                   PIN 19,20,21    ISTYPE 'REG_JK';

       H,L,CK,X = 1,0,.C.,.X.;

       GREATER =  COMPARE; "IF DIGITAL OUTPUT IS GREATER THAN ANALOG INPUT.
       LESS    = !COMPARE; "IF DIGITAL OUTPUT IS LESS THAN ANALOG INPUT.

       SREG = [BIT4,BIT3,BIT2,BIT1,BIT0,DONE];

       BEGIN        = [X, X,X,X,X, X];
       INIT         = [0, 0,0,0,0, 0];
       ST16         = [1, 0,0,0,0, 0];
       ST8          = [X, 1,0,0,0, 0];
       ST4          = [X, X,1,0,0, 0];
       ST2          = [X, X,X,1,0, 0];
       ST1          = [X, X,X,X,1, 0];

       AD8          = [X, 1,X,X,X, X];
       AD4          = [X, X,1,X,X, X];
       AD2          = [X, X,X,1,X, X];
       AD1          = [X, X,X,X,1, X];
       IEND         = [X, X,X,X,X, 1];

       SH8          = [0, 1,X,X,X, X];
       SH4          = [X, 0,1,X,X, X];
       SH2          = [X, X,0,1,X, X];
       SH1          = [X, X,X,0,1, X];
       SH0          = [X, X,X,X,0, 1];
```

APPENDIX B: ABEL DESIGN FILE OF U2 (Continued)

```
EQUATIONS
SREG.CLK = CLOCK;
nHLD.CLK = CLOCK;

"Non-Inverting Input Latch: nHLD = nHOLD"
nHLD.J = !nHOLD;
nHLD.K = nHOLD;

"UP/DOWN COUNTER"
BIT0.J = (nSTART & !nTRACK & !DONE.Q & nHLD );
BIT0.K = (nSTART & !nTRACK & !DONE.Q & nHLD );
BIT1.J = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q);
BIT1.K = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q);

BIT2.J = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q & !BIT1.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q & BIT1.Q);
BIT2.K = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q & !BIT1.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q & BIT1.Q);

BIT3.J = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q & !BIT1.Q & !BIT2.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q & BIT1.Q & BIT2.Q);
BIT3.K = (nSTART & !nTRACK & !DONE.Q & nHLD & !COMPARE & !BIT0.Q & !BIT1.Q & !BIT2.Q
         #nSTART & !nTRACK & !DONE.Q & nHLD & COMPARE & BIT0.Q & BIT1.Q & BIT2.Q);

BIT4.J = (nSTART& !nTRACK& !DONE.Q& nHLD &!COMPARE &!BIT0.Q  &!BIT1.Q  &!BIT2.Q  & !BIT3.Q
         #nSTART& !nTRACK& !DONE.Q& nHLD &COMPARE  &BIT0.Q &BIT1.Q &BIT2.Q & BIT3.Q);
BIT4.K = (nSTART& !nTRACK& !DONE.Q& nHLD &!COMPARE &!BIT0.Q  &!BIT1.Q  &!BIT2.Q  & !BIT3.Q
         #nSTART& !nTRACK& !DONE.Q& nHLD &COMPARE  &BIT0.Q &BIT1.Q &BIT2.Q & BIT3.Q);

nDONE_OC = 0;
nDONE_OC.OE = !DONE.Q;

RC = 0;
RC.OE = RC;

CLOCK1 = RC;
CLOCK2 = !CLOCK1;

STATE_DIAGRAM SREG;

STATE BEGIN:
      if !nSTART then INIT;

STATE INIT:
      if BIT5 then  ST16

STATE  ST16:
      if GREATER then  SH8
      if LESS  then  AD8;

STATE  ST8:
      if GREATER  then SH4
      if LESS  then AD4;

STATE  ST4:
      if  GREATER  then  SH2
      if  LESS  then  AD2;

STATE  ST2:
      if  GREATER then  SH1
      if  LESS  then  AD1;

STATE  ST1:
      if  GREATER  then  SH0
      else IEND;
```

APPENDIX B: ABEL DESIGN FILE OF U2 (Continued)

```
TEST_VECTORS
([nSTART,nHOLD,nTRACK,BIT5,COMPARE,CLOCK] -> [BIT4,BIT3,BIT2,BIT1,BIT0,DONE,nHLD]);

[1,1,1,0,0,0 ] -> [1,1,1,1,1,1,1]; "1 Power_on reset
[1,1,1,0,0,CK] -> [1,1,1,1,1,1,1]; "2
[0,1,1,0,0,CK] -> [0,0,0,0,0,0,1]; "3 nSTART clears all the output registers
[1,1,1,0,1,CK] -> [0,0,0,0,0,0,1]; "4
[1,1,1,0,1,CK] -> [0,0,0,0,0,0,1]; "5
[1,1,1,0,1,CK] -> [0,0,0,0,0,0,1]; "6
[1,1,1,1,1,CK] -> [1,0,0,0,0,0,1]; "7 bit5=1 ripples into U2 to start SAR of bit0-4
[1,1,1,0,1,CK] -> [0,1,0,0,0,0,1]; "8
[1,1,1,0,1,CK] -> [0,0,1,0,0,0,1]; "9
[1,1,1,0,0,CK] -> [0,0,1,1,0,0,1]; "10

[1,1,1,0,0,CK] -> [0,0,1,1,1,0,1]; "11
[1,1,1,0,0,CK] -> [0,0,1,1,1,1,1]; "12
[1,1,1,0,0,CK] -> [0,0,1,1,1,1,1]; "13 end of SAR of bit0-4
[1,1,1,0,1,CK] -> [0,0,1,1,1,1,1]; "14
[1,1,1,0,1,CK] -> [0,0,1,1,1,1,1]; "15
[1,1,1,0,1,CK] -> [0,0,1,1,1,1,1]; "16
[1,1,0,0,1,CK] -> [0,0,1,1,0,1,1]; "17 Tracking started and counts down
[1,1,0,0,1,CK] -> [0,0,1,0,1,1,1]; "18 down
[1,1,0,0,0,CK] -> [0,0,1,1,0,1,1]; "19 up
[1,1,0,0,0,CK] -> [0,0,1,1,1,1,1]; "20 up

[1,1,0,0,0,CK] -> [0,1,0,0,0,1,1]; "21 up
[1,1,0,0,1,CK] -> [0,0,1,1,1,1,1]; "22 down
[1,1,0,0,1,CK] -> [0,0,1,1,0,1,1]; "23 down
[1,0,0,0,0,CK] -> [0,0,1,1,1,1,0]; "24 nHOLD becomes 0, hold
[1,0,0,0,0,CK] -> [0,0,1,1,1,1,0]; "25
[1,0,0,0,1,CK] -> [0,0,1,1,1,1,0]; "26
[0,1,1,0,0,CK] -> [0,0,0,0,0,0,1]; "27 Next conversion cycle begins
[0,1,1,0,0,CK] -> [0,0,0,0,0,0,1]; "28
[1,1,1,0,1,CK] -> [0,0,0,0,0,0,1]; "29
[1,1,1,1,1,CK] -> [1,0,0,0,0,0,1]; "30

[1,1,1,1,1,CK] -> [0,1,0,0,0,0,1]; "31
[1,1,1,1,1,CK] -> [0,0,1,0,0,0,1]; "32
[1,1,1,1,1,CK] -> [0,0,0,1,0,0,1]; "33

END ADCS2

H
```

Interrupt Handler — PLS179

As an example of designing a microprocessor family part, consider Figure 1, which depicts an interrupt handler. In particular, note that interrupt inputs will be latched into an 8-bit register. This in turn will be encoded to a 3-bit vector which may be appropriately enabled and applied to the microbus. Figure 1 shows the eight flip-flops as having J-K and /D inputs which will be generated with a PLS179 by switching the flip-flop control. Appropriate control signals for the various transactions might be as follows:

1. CLOCK – the system synchronous time base.
2. Interrupt Enable – when asserted high from the microprocessor, allows interrupts to be generated to the microprocessor.
3. Interrupt – a strobe or level defined to indicate a pending interrupt and a valid encoded vector.
4. Interrupt Acknowledge – a response signal from the microprocessor which may be used to enable the 3-bit vector onto the bus. As well, it may initiate clearing the currently asserted interrupt latch.
5. /INT0–/INT7 – eight possible interrupt request signals which must be asserted low and held there until service for that device has occurred.
6. Reset – this is a system override signal which will clear all flip-flops during initial operation.

Basic Operation

Initially, the part should be reset by asserting the RESET pin high, asynchronously. Then, when interrupts are enabled, the /D-inputs to the 8 flip-flops will be synchronously scanning for interrupt inputs (asserted low). This will put a nonzero value into the eight bit register which will generate an interrupt output, combinationally through the Complement array. In parallel, a 3-bit encoded vector will be applied on the VEC0, VEC1, VEC2 lines. Asserted high logic will be assumed for the vector. Presumably, a microprocessor will interrupt this, transfer control to a service routine and clear the interrupt. The clear will be accomplished by disabling interrupts and strobing the vector value back into the PLS179, using the IACK signal. Disabling the interrupts will put the registers into J-K mode. J is tied to zero and K is decoded from the specifically strobed vector. Therefore, synchronous clear of the high priority bit is done. Interrupts are then re-enabled and the process continues.

The PLS179 solution offers room for user alteration. For example, the IACK condition could be redefined as a combination of the Z80 IOREQ and M1 signals, or any specific splitting of internal signals could be easily done. The design could fit into a PLS159A, but there would be less room for variation for specific users exact needs. Figure 3 shows the pinlist for the handler. Figure 4 gives the corresponding design file.

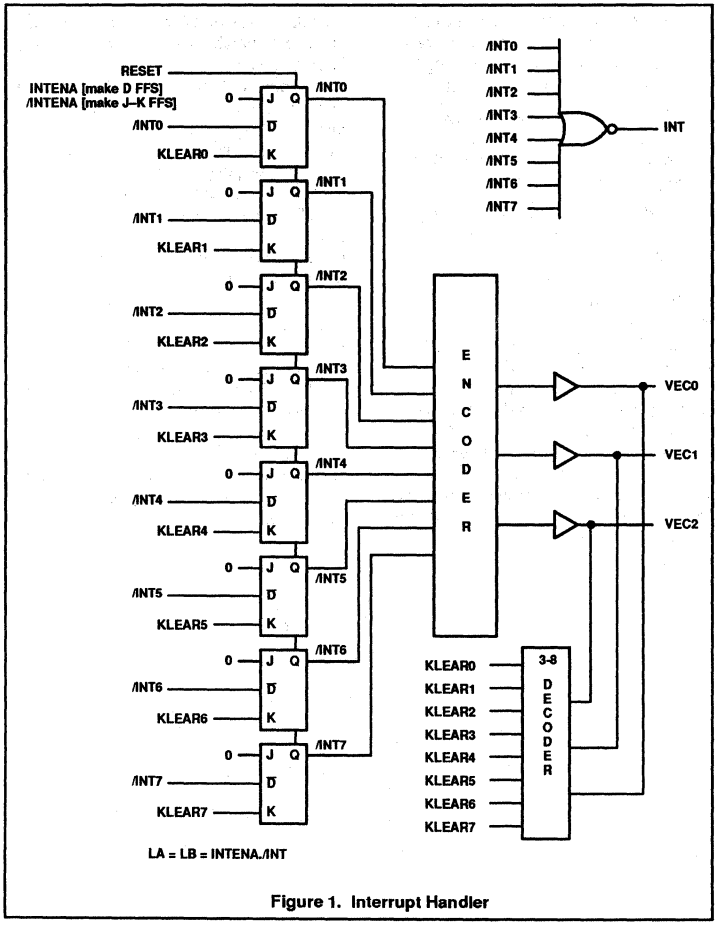


Figure 1. Interrupt Handler

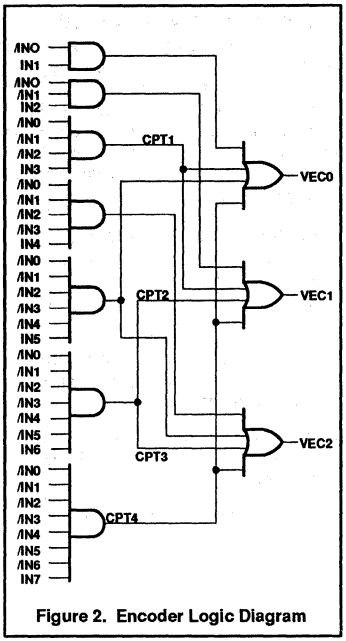


Figure 2. Encoder Logic Diagram

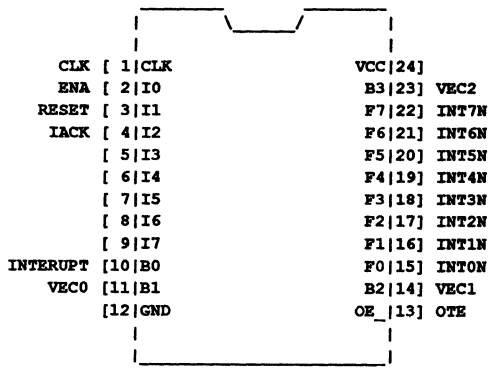


Figure 3. Interrupt Handler Pin List

```
@PINLIST
OTE          I;          INT4N        B;
IACK         I;          INT3N        B;
RESET        I;          INT2N        B;
ENA          I;          INT1N        B;
CLK          I;          INT0N        B;
VEC2         B;          VEC1         B;
INT7N        B;          VEC0         B;
INT6N        B;          INTERUPT     O;
INT5N        B;

@LOGIC EQUATIONS

"Encoder Equations"
CPT1 = INT0*INT1*INT2*/INT3;
CPT2 = INT0*INT1*INT2*INT3*INT4*/INT5;
CPT3 = INT0*INT1*INT2*INT3*INT4*INT5*/INT6;
CPT4 = INT0*INT1*INT2*INT3*INT4*INT5*INT6*/INT7;
VEC0 = (INT0*/INT1+CPT1+CPT2+CPT4);
VEC1 = (INT0*INT1*/INT2+CPT1+CPT3+CPT4);
VEC2 = (INT0*INT1*INT2*INT3*/INT4+CPT2+CPT3+CPT4);
VEC0.oe = ENA;
VEC1.oe = ENA;
VEC2.oe = ENA;

C = /(/INT0+/INT1+/INT2+/INT3+/INT4+/INT5+/INT6+/INT7);
INTERUPT = C;
INTERUPT.oe = ENA;

"Decoder Equations"
KLEAR0 = /VEC2*/VEC1*/VEC0*IACK; "DECODE VECTOR 0"
KLEAR1 = /VEC2*/VEC1*VEC0*IACK;  "DECODE VECTOR 1"
KLEAR2 = /VEC2*VEC1*/VEC0*IACK ; "DECODE VECTOR 2"
KLEAR3 = /VEC2*VEC1*VEC0*IACK ;  "DECODE VECTOR 3"
KLEAR4 =  VEC2*/VEC1*/VEC0*IACK; "DECODE VECTOR 4"
KLEAR5 =  VEC2*/VEC1*VEC0*IACK ; "DECODE VECTOR 5"
KLEAR6 =  VEC2*VEC1*/VEC0*IACK ; "DECODE VECTOR 6"
KLEAR7 =  VEC2*VEC1*VEC0*IACK  ; "DECODE VECTOR 7"

"Register equations"
INT0.J=in0j;                    INT4.J=in4j;
INT0.K=KLEAR0+in0k;             INT4.K=KLEAR4+in4k;
INT1.J=in1j;                    INT5.J=in5j;
INT1.K=KLEAR1+in1k;             INT5.K=KLEAR5+in5k;
INT2.J=in2j;                    INT6.J=in6j;
INT2.K=KLEAR2+in2k;             INT6.K=KLEAR6+in6k;
INT3.J=in3j;                    INT7.J=in7j;
INT3.K=KLEAR3+in3k;             INT7.K=KLEAR7+in7k;

"Register RESET equations"
INT0.rst=RESET;                 INT4.rst=RESET;
INT1.rst=RESET;                 INT5.rst=RESET;
INT2.rst=RESET;                 INT6.rst=RESET;
INT3.rst=RESET;                 INT7.rst=RESET;
```

Figure 4. Interrupt Handler Design File

PLUS405-55 – the ideal high speed interface AN034

INTRODUCTION

Philips Semiconductors PLUS405–55 is ideal for high performance microprocessor interfacing applications. Being a programmable integrated circuit, it adapts to nearly any bus or microprocessor protocol. The PLUS405–55 can make state machines, counters, and shifters running at speeds of 55 megahertz. The architecture of the PLUS405–55 combines a powerful programmable logic array with 16 JK flip-flops to form a programmable part superior to any comparable PAL part.

WHAT IS INTERFACING?

Interfacing is the translation of digital signals from one target device to another (see Figure 1). Each target device has their own unique signal behavior, and may not be directly connected to each other. Correct connection occurs with the use of a flexible interface. For instance, today's chip set integrated circuits connect a microprocessor to its memory and I/O devices. The signals presented by the microprocessor are not the specific /CAS and /RAS needed by the memory. Nor are the microprocessor signals the correct chip enables to attach to the UAR/T or graphics controller. Forming the translated signals which are appropriate for the memories, UAR/T's and other controllers is the job of the interface chips. Unfortunately, there are not off-the-shelf interface parts for all applications. That is where programmable parts excel.

With this information, we can ask: what are the qualities of an ideal interface part?

An ideal interface part must have sufficient logic inside to make correctly any logic translation needed. Because almost every interface is between two target devices which use separate clocks, conventional PAL parts are not adequate. This includes the popular 22V10 and most other registered PAL parts. To reconcile that an interface may exist between two target parts using different clocks, the interface must be able to synchronize signals from either or both target devices. So, an ideal interface must be clockable from at least two different clocks.

An ideal interface part must have enough flip-flops inside to capture data or control information from the target parts, and to re-synchronize control signals. This suggests it needs at least 16 flip-flops because the interface might receive 8 from either target device.

To support the use of the 16 flip-flops as handshaking flip-flops, a typical number of logic AND gates is about 4 gates per flip-flop. This number of gates can be less if the type of flip-flop is a JK rather than a D flip-flop.

An ideal interface must be as fast – if not faster – than the fastest target devices it must interface. This requirement is sometimes misunderstood. Most microprocessor parts have a clock input which is the fastest signal present at the outside world. Very few signals coming from the microprocessor are as fast or faster than the clock because they are usually made from flip-flop circuits inside the microprocessor.

In the past, many designers believed that interface PAL parts had to be twice as fast as the system clock. This was because either the system clock was operating at twice the crystal frequency or the PAL part had to compensate for the fact that some events occur on rising clock edges and other events happen on falling clock edges. This misconception should be reexamined. What is needed is an interface part which can respond to both edges of the basic clock rate, as the interface dictates.

Finally, the ideal interface must be electrically compatible with both devices that must communicate. This is almost always either CMOS or TTL and today's technologies usually support a standard TTL interface.

THE PLUS405-55 . . . THE IDEAL INTERFACE!

To meet the needs of the ideal interface, Philips Semiconductors designed the PLUS405–55 (see Figure 2). The 405 is fast enough to respond to microprocessor clocks in excess of 50MHz, contains 16 JK flip-flops and a programmable logic array to control the flip-flops. It accepts two clock sources and permits the internal flip-flops to be grouped in one of two standard ways. The PLUS405–55 handles simple data synchronization or complex bus handshaking between two target devices, within a single part. It is widely supported by Philips Semiconductors SNAP as well as third-party design tools.

Let's look inside the PLUS405 to see how it works. First, there are 15 input pins, each supplying a signal or its complement, to the main AND array of the 405. There is an additional input pin which can optionally bring in a clock. There are 8 output pins which are directly tied to specific flip-flop outputs (labeled F0 to F7). This is the fastest flip-flop configuration possible. It should be noted that the clock to Q time delay (t_{CKQ}), measured from the 405's pins, is 8 nanoseconds (max). This means that signals can get into and out of the PLUS405–55 very fast.

The choice of flip-flops was the JK flip-flop. For building counters, JKs require only one gate per bit of additional logic. For building shift registers inside, no gates are required except that all connections use gates, so it uses a small number to connect the shift register.

For making state machines, the number of gates per flip-flop is up to the application. There is no design restriction with programmable AND gates, which can be assigned as needed to any OR gates. This is superior to the ordinary PAL approach or even the 22V10 approach where each OR gate permanently connects to specific AND gates. AND gate outputs are shared as needed and there is no limit on how many OR gates may be driven from a single AND term. JK flip-flops are superior to D flip-flops for all state machine applications, because they do not need as much external logic to control their behavior.

At this point, we see that the PLUS405–55 combines two superior elements. Namely, JK flip-flops for making state machines and a programmable logic array (PLA) for forming logic expressions. However, there are still more features which the PLUS405–55 includes. There is a special pin for tri-state control of the 8 output pins or alternately it can initialization the 405. When used to initialize, the pin can applies any value to the asynchronous set or preset of each flip-flop. Initialization is to any state chosen.

The internal flip-flop connection in the 405 is also critical. Flip-flops are first grouped into two categories – 8 internal flip-flops and 8 output flip-flops. The internal flip-flops do not directly access the output pins and the output flip-flops do not feedback. To form state machines inside the 405, the internal flip-flops are best used. This doesn't mean that the output flip-flops can't be used, but rather that their use with feedback requires external connection. This is seldom necessary.

Another important flip-flop grouping is inside the 405. Four output flip-flops (F0–F3) link with their clock inputs to four internal flip-flops. The other four output flip-flops similarly link with four other internal flip-flops. This permits two separate state machines to be built. One state machine uses 4 internal flip-flops and four output flip-flops on one clock and four other internal flip-flops linked to their four output flip-flops on another clock. Alternately, the can all be linked together to a common clock, or by an external inverter, one

state machine can operate on the clock while the other one operates on the clock's complement.

Almost hidden from view in Figure 2 are two additional OR gates which do not drive any flip-flops. These two OR gates are inverted and fedback to the input of the AND logic array. The two inverted OR gates (i.e., NOR gates) are called complement arrays. They are used to save AND gates when state transitions are complicated. Luckily, the design software – SNAP and SLICE – automatically use these gates to save the designer from having to use them. The complement arrays also permit automatic homing to known states, if a power transient accidentally puts a state machine into an undefined state.

PERFORMANCE

As mentioned previously, the clock to Q time delay is 8 nanoseconds maximum. That is only one part of the performance equation. The flip-flop speed is:

$$f_{MAX} = \frac{1}{t_{SETUP} + t_{CKQ}}$$

When a flip-flop is put into a circuit with logic driving it, the logic adds delay which slows the circuit down. For a flip-flop with extra logic, that logic delay is included in the performance equation as follows:

$$f_{MAX} = \frac{1}{t_{SETUP} + t_{DELAY} + t_{CKQ}}$$

Passing signals through the PLA section of the PLUS405 (the complement array) will add additional time delay as follows:

t_{SETUP} = 10 nsec

t_{CKQ} = 8 nsec

t_{DELAY} = 8 nsec

Without the complement array, the f_{MAX} is 55.6MHz found as the inverse of 18 nanoseconds. This is because the specification includes a single signal pass through the PLA as part of the flip-flop setup time. Viewed from the outside, this makes sense because there is no way to get a signal to the flip-flop without entering the PLA. With the complement array, f_{MAX} is the inverse of 26 nanoseconds, found by adding the t_{DELAY} term to the f_{MAX} expression. This sets f_{MAX} with the complement array at 38.5MHz. An additional pass through the complement array is never needed, so the PLUS405–55 will never be slower than 38MHz. The additional complement array is included in case two distinct state machines are built, where each needs one.

The PLUS405–55 includes a PLA which has 64 AND gates in it. This permits an average of 4 AND gates per flip-flop, but this many are seldom needed, because the JK flip-flops are so efficient. Because the AND gate outputs are shared as needed, redundant terms are never used. There are additional buffers to do the asynchronous flip-flop control, and each JK flip-flop includes its own OR gates.

DESIGN SUPPORT

Designers need tools which can capture the design, compile it to a fusemap and download commercially available programmers. Philips Semiconductors supports the PLUS405–55 with SNAP design software (Figure 3). This software runs on personal computers and permits designs to be formulated with Boolean logic equations, state equations or schematics. SNAP, the full-featured product, includes advanced simulation capability found only in Field Programmable Gate Array (FPGA) or ASIC design software.

SNAP includes a simulator with back annotation of time delays to accurately model the PLUS405-55 as well as Philips Semiconductors full PLD product line. Additional support for the PLUS405–55 can be found in third-party design tools.

AN EXAMPLE OF INTERFACING WITH THE PLUS405-55

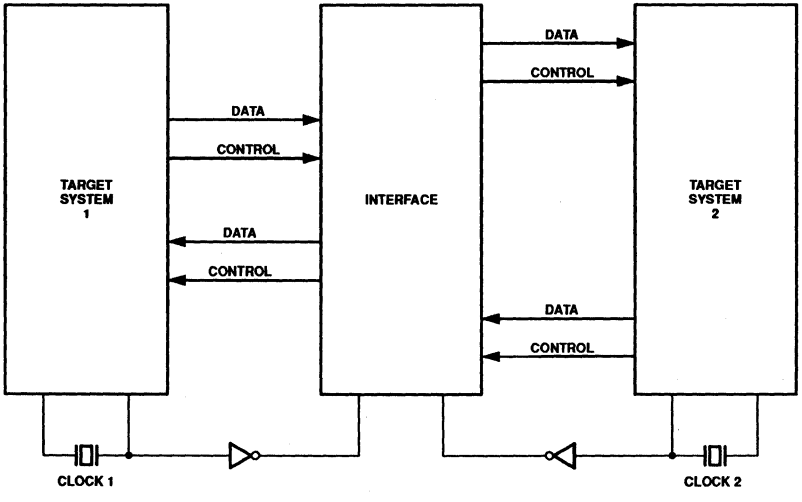


Figure 1. Digital Interfacing

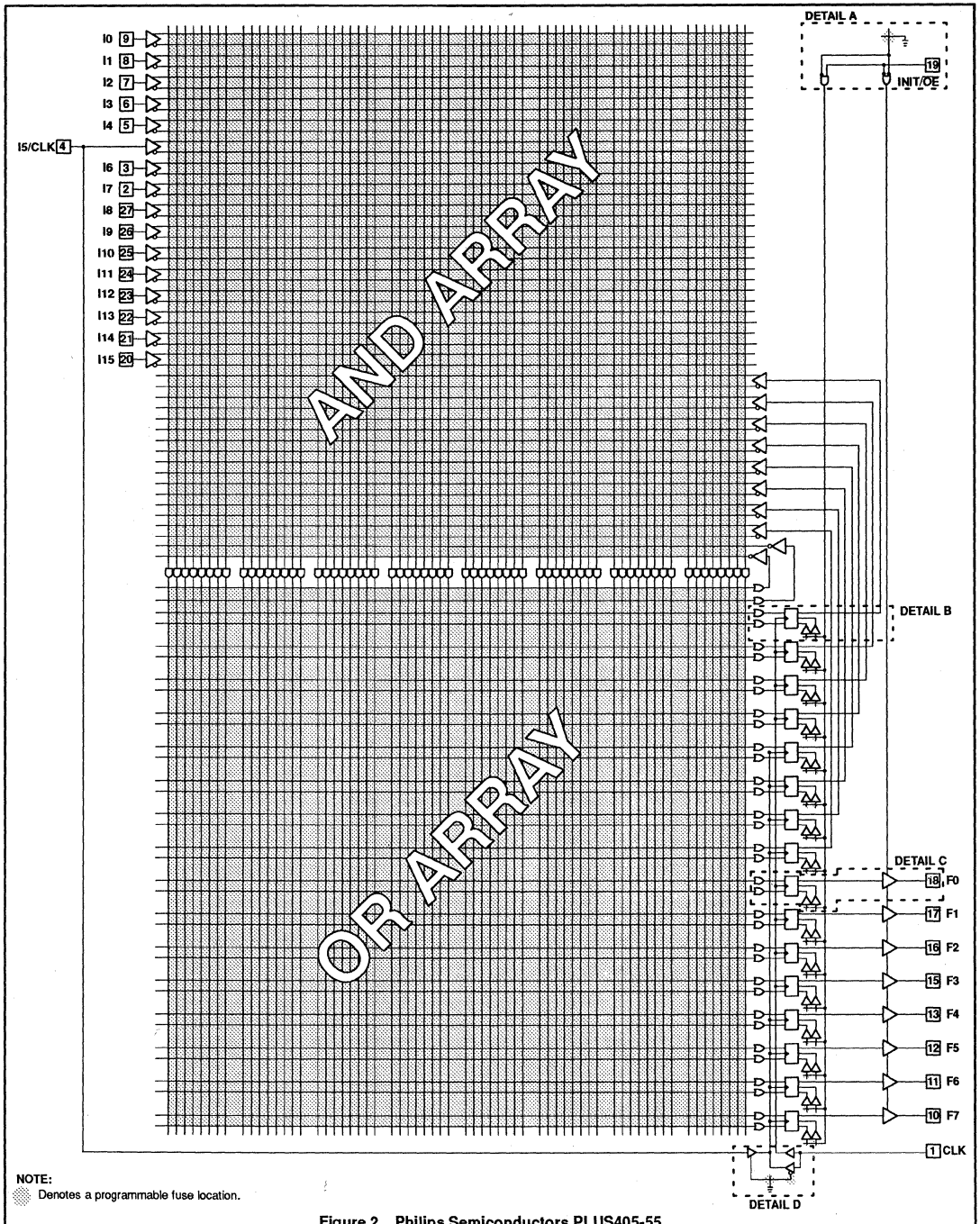


Figure 2. Philips Semiconductors PLUS405-55

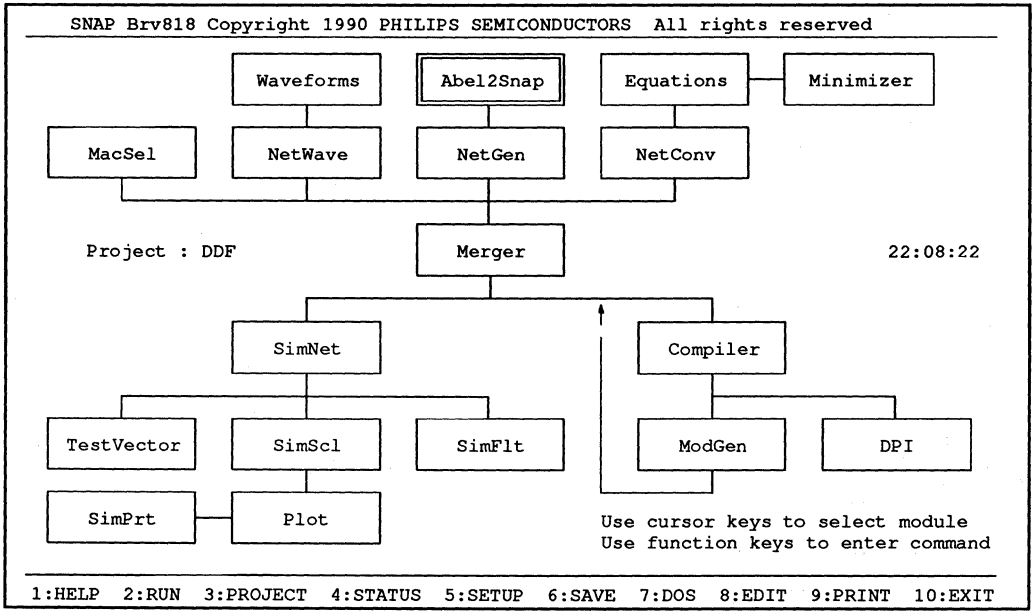


Figure 3. SNAP Shell

Single chip multiprocessor arbiter AN07

INTRODUCTION

In multiprocessor environments there is considerable savings to be made through sharing system resources. If each processor must support its own bus structure, I/O devices, and bulk storage medium, system cost could be very high. In the configuration shown in Figure 1, all processors share a common communication bus, and a number of system resources.

Since every processor must use the common system bus to communicate with its peripherals, a priority structure that resolves simultaneous processor bus requests into a single bus grant must be integrated into the system. In addition to making request-grant transactions, transient bus contention due to grant switching must be removed by inserting precise guard band times between bus grants.

Philips Semiconductors Programmable Logic Sequencer provides a convenient and cost effective means for implementing a synchronous arbiter to perform these tasks within a single chip.

ARBITER STRUCTURE

Within a multiprocessor system, two general classes of processors can be recognized: Priority A and Priority b. Priority A processors have the highest request priority and must only compete with other Priority A processors for bus control. The arbiter must issue "A" grants in a manner that prevents any high priority "A" processor from locking out another Priority A processor. To enable this, the Priority A rules implemented here use a Last Granted Lowest Priority (LGLP) ring structure. After an "A" processor has completed a bus-related task, its next arbitrated request priority will be lowest in the "A" request group. The previously second highest priority "A" processor will then become highest priority requester. The net effect of the "round robin" exchange is that every Priority A processor will have a turn at being highest priority processor. Priority A processors are typically ones that perform real-time operations or vital system tasks.

Priority b processors are lower in priority than the "A's" and may only be granted system control when no "A" requests are pending. "b" processors usually perform background tasks. Within the Priority b group, further priority ordering exists such that each "b" processor has a fixed priority position.

Plumber[1], Pearce[2], and Hojberg[3] present asynchronous techniques of arbiter implementation. These methods all have hard-wired priority rules and imprecise guard band times during grant switching. As pointed out by Hojberg, a synchronous state machine can be configured as a Mealy-type controller to provide not only precise guard band times and programmable priority rules, but also programmable input/output polarity. The state machine in Figure 2 is made from a control PROM array and an edge-triggered latch. The "A" and "b" requests and the machine's present state are used by the control PROM to determine the next "A" and "b" grants and the next state.

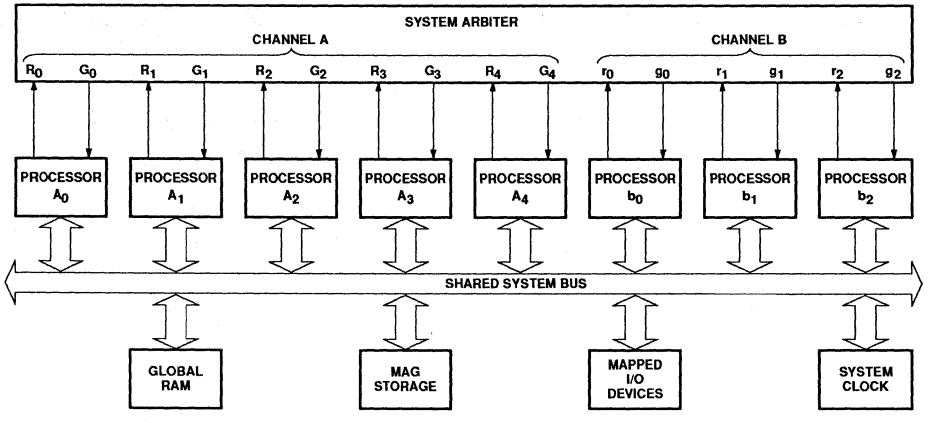
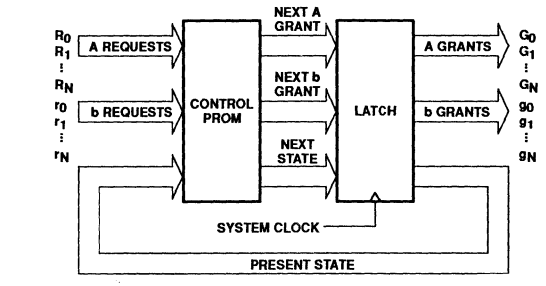


Figure 1. Multiprocessor Structure

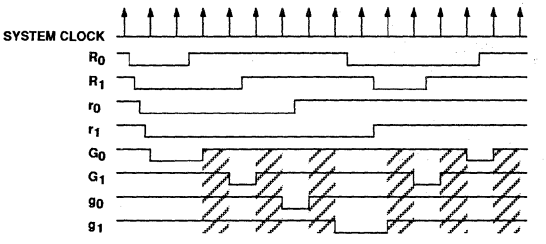
Single chip multiprocessor arbiter AN07

SYSTEM OPERATION

Two machine states can be identified by inspection: a wait state and a grant state. The state machine enters a grant state as a response to a system request on either R_N or r_N. The machine will remain in this state with a single grant line asserted as long as the request remains asserted. Upon releasing the request line, the machine will pass through a single wait state before considering other pending requests. This provides a single state guard band time. The requests received must meet the set-up requirement of the edge-triggered latch after propagating through the control PROM. If these time considerations do not fit within a given multiprocessor structure, an input latch may be added such that the R_N and r_N lines are clocked through the latch by the system clock, thereby removing asynchronous set-up time considerations. On the basis of a state machine approach, two techniques of implementation are feasible: 1) using an architecturally advanced single IC controller, the PLS, and, 2) a traditional PROM/LATCH configuration.



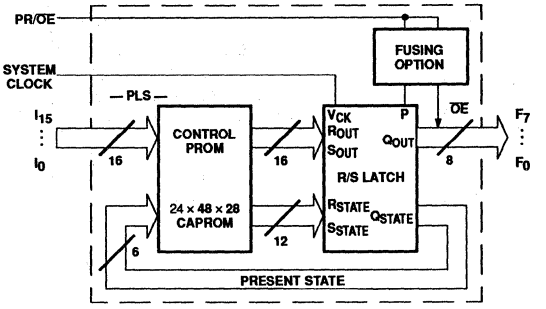
a. A and b service requests (R_N, r_N) plus the present state determine, through the control PROM, the next state and the next grant outputs (G_N, g_N).



b. Requests R_0, R_1, r_0, and r_1 are asserted low in the same clock sample period. The priority rules that determine the order in which the grants are issued and the shaded guard-band areas are programmed into the control PROM. Note that the A and b request lines and the present state input to the PROM must have a set-up time equal or greater than the latch set-up time plus the PROM access time.

Figure 2. Arbiter Constructed from a Mealy-Type State Machine

Single chip multiprocessor arbiter | AN07



a. PLS Single Chip System

		TRANSITION TERM																OPTION (P/E)											
																		OUTPUT TERM											
NO.	C	INPUT VARIABLE (Im)																PRESENT STATE (Ps)						NEXT STATE (Ns)						OUTPUT FUNCTION (Fn)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0																																					
1																																					
2																																					
3																																					
4																																					
5																																					
46																																					
47																																					

b. PLS Standard Program Table

CK	S	R	Q
↑	0	0	Q
↑	0	1	0
↑	1	0	1
↑	1	1	Undefined

NOTE:
(↑) denotes positive clock transition.
S = R = 1 is an illegal input condition.

c. Transition Table of Clocked S/R Flip-Flop

Figure 3.

PLS ARBITER IMPLEMENTATION

A five Priority "A" and three Priority "b" arbiter will be constructed such that all grant outputs will be asserted low for grants and all request inputs will be asserted low for system requests.

Brief PLS Description

The PLS block diagram shown in Figure 3(a) consists of a control PLA and 14 clocked S/R flip-flops. The control PLA is actually an AND-OR logic array that functions as a Content Addressable PROM. The PLA is organized as 48 words of 28 bits with 16 external input lines, and six internal inputs fed back from the State Register. The 28 PLA outputs drive the S/R inputs of the six-bit State Register and eight-bit Output Register. Note that the state feedback path is made inside the PLS.

I_N and present state inputs, P_S, represent 2^{22} possible input codes; 48 of these codes may be mapped in the PLA to provide a 14 bit register control word. As shown in Figure 3(b) each input code may be specified by assigning to the variables either Low "L", High "H", or Don't Care "–" logic states. If any input code falls logically outside the programmed codes, the PLA asserts a Low on all its 28 internal outputs, thereby issuing a "no change" command to the R/S flip-flops.

This is an important architectural feature because it requires that only state or output transition terms be programmed. Looping terms that change neither state nor output need not be programmed in the PLS, owing to the functional characteristics of S/R flip-flops tabulated in Figure 3(c). An example of this is shown in Figure 4.

The S/R inputs of both state and output registers are specified by using PLA outputs ("AND" functions of request inputs and present state) in the program table of Figure 3(c).

Single chip multiprocessor arbiter AN07

The corresponding next state of each bit will be set to 0 for "L", 1 for "H", and No Change for "–". The PLS's PR/OE line may be assigned either Asynchronous Preset or Output Enable functions, via a user programmable option.

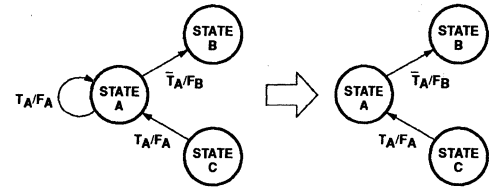
The entire function is integrated into a single 28-pin package designated as PLUS105.

State Algorithm

Figure 5(a) displays the circular state form and all possible state transitions of the LGLP priority structure. Hex states 3F, 3E, 3D, 3C, and 3B are arbiter wait states W_{0-4}. In these states, processor "A" and "b" requests are monitored. Figure 5(b) illustrates a typical grant to processor A_1 in hex state 07. As long as A_1 asserts its request line low, the next state will be 07_{16} and the next output will remain with G_1 asserted low and all the other grant outputs asserted high. Since no change in state or grant output results from this transition, no PLA resources are required.

As soon as processor A_1 returns its request line, R_1, to 1, a state transition is made to 3D, and an output transition is made to set all grant outputs to 1. Since processor A_1 was the last to be granted system resources, it will now have the lowest A level request priority (LGLP). In wait state W_2, the highest priority processor will be A_2, second A_3, third A_4, and fourth A_0. To maintain the LGLP rule, grant transitions must follow the state rule $G_N \rightarrow W_{(N+1)}$, and wait states, W_M, must set their "A" priorities so that processor A_M is highest priority. Priority decreases as one proceeds clockwise around the state ring to the lowest priority processor, $A_{(m-1)}$.

When no "A" requests are pending, "b" requests may be granted. To avoid upsetting the LGLP priority rule, a "b" grant must leave and return to the same wait state. Since the "b" priority structure is the same regardless of the wait state, only a single set of "b" transition terms are required.



NOTE:
T_N/F_N = Inputs/outputs.
The PLS requires that only two out of the three transition terms be programmed.

Figure 4. PLS State and Output Transitions

For example, a grant transition to g_2 (Hex 20-25) can be issued only if there are no "A", "b_0", or "b_1" requests pending. Given the binary wait state code 111XXX, where "X's" represent Don't Cares, a request code of 01111111 will transfer the arbiter to the grant state g_2 from any of the wait states, W_{0-4}.

It is important to realize that in making this transition, the lower 3-State bits will not be changed—they provide the wait state return address. When r_2 returns high, 1XXXXXXX, a transition back to the previously exited wait state is made by forcing a "1" in the three most significant state bits and leaving the lower 3-State bits unchanged.

All output and state bits are initially preset to "1" through the use of the optional preset function. Grant output lines are only forced low when transitions are made to grant states and are returned to "1" when jumping back to a wait state.

The complete arbiter circuit diagram is shown in Figure 6a. The SNAP equations are shown in Figure 7.

PROM/LATCH IMPLEMENTATION

The same five "A" processor and three "b" processor arbiter can be implemented with discrete PROM's and Latches using the same state diagrams for the PLS, except that now looping transition terms must be programmed. Coding of all state and output transitions requires programming of two memory fields: the "A" request PROM's (2KX6) and the "b" request PROM (64×3). The complete circuit diagram is shown in Figure 6(b).

The "A" request PROM's determine the next machine state (N_{0-5}) at all times, except when there are no "A" requests pending and there is a "b" request, or if the machine is presently in a "b" grant state. In these cases, the "b" request PROM controls the machine's next state.

The grant control lines are decoded from the next state lines and latched in two quad output latches. This PROM/LATCH organization is shown in Figure 6a.

Single chip multiprocessor arbiter AN07

Table 1. Design Alternatives for the Priority Arbiter

PARAMETER	SEQUENCER	PROM/LATCH
Parts count	1 IC	≈19IC's
PCB space	.84 in^2	7.92 in^2
Power	.65W	2.85W
Voltage	+5V	+5V

SUMMARY

As can be seen from the circuit diagrams, the PLS can offer significant advantages over discrete MSI arrays in the design of state machines. The tradeoff in both design alternatives for the Priority Arbiter is shown in Table 1. Clearly, the PLS approach uses fewer parts, with savings in PC board space and power requirements.

REFERENCES

1. W.W. Plumber: "Asynchronous Arbiters"; *IEEE Transactions on Computers*, January 1972, pp. 37–42.
2. R.C. Pearce, J.A. Field, and W.D. Little: "Asynchronous Arbiter Module"; *IEEE Transactions on Computers*, September 1975, pp. 931–933.
3. K. Soe Hojberg: "An Asynchronous Arbiter Resolves Resource Allocation Conflicts on a Random Priority Basis"; *Computer Design*, August 1977, pp. 120–123.
4. K. Soe Hojberg: "One–Step Programmable Arbiter for Multiprocessors"; *Computer Design*, April 1978, pp. 154–158.

Single chip multiprocessor arbiter AN07

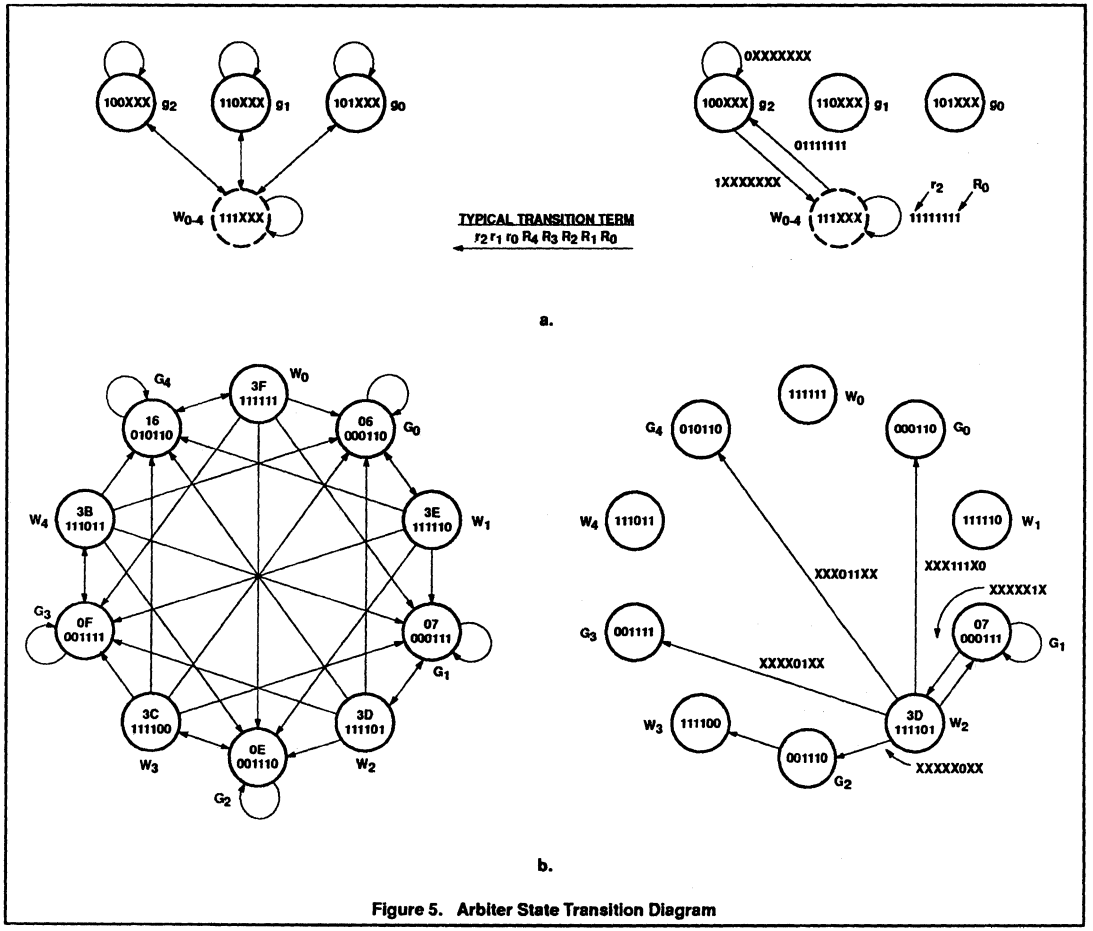
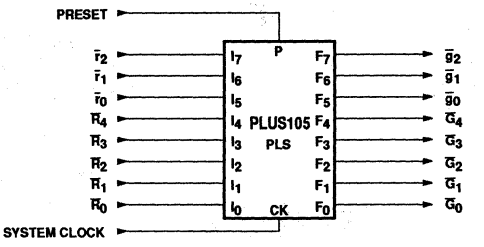
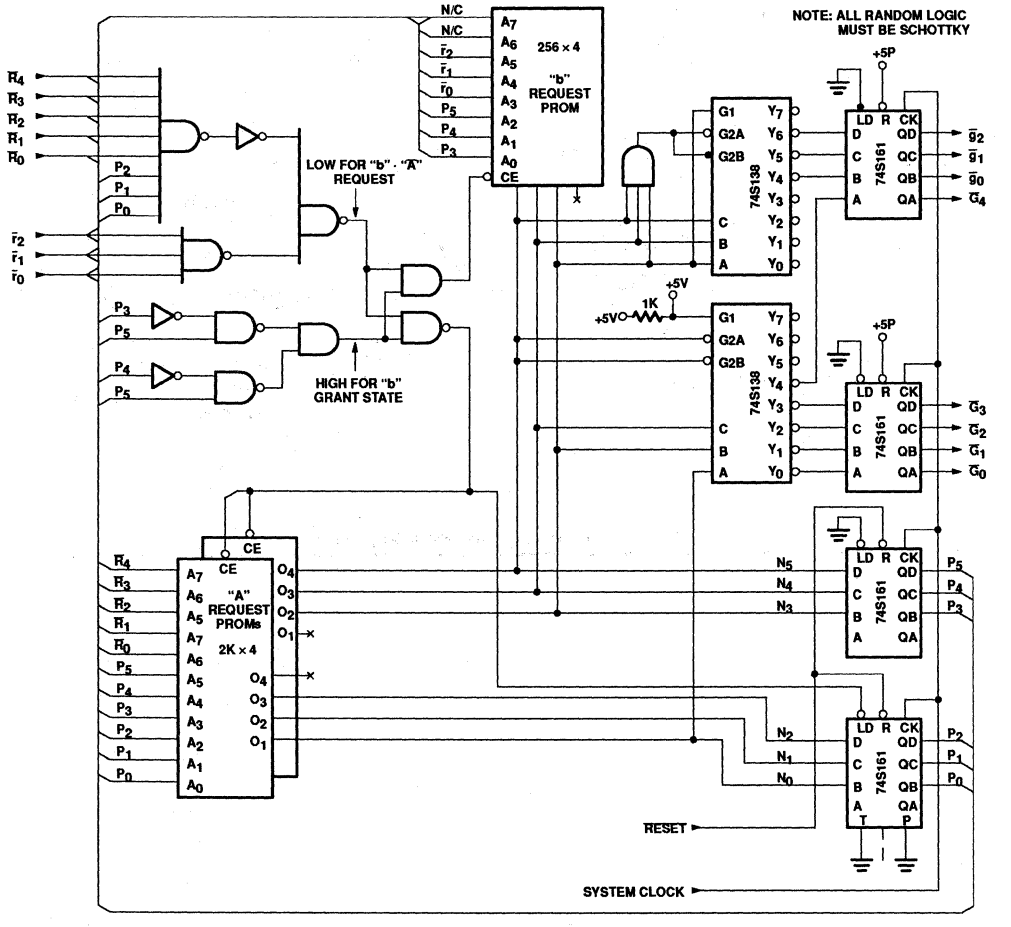


Figure 5. Arbiter State Transition Diagram



a.



b.

Figure 6. Arbiter Circuit Diagram Summary

Single chip multiprocessor arbiter | AN07

```
"-------------------------------------------"
"  Single Chip Multiprocessor Arbiter    AN7  "
"-------------------------------------------"
@PINLIST
CLOCK I;          PRESET I;
RB2 I;            OB2 O;
RB1 I;            OB1 O;
RB0 I;            OB0 O;
RA4 I;            OA4 O;
RA3 I;            OA3 O;
RA2 I;            OA2 O;
RA1 I;            OA1 O;
RA0 I;            OA0 O;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
OB0.CLK = CLOCK;
OB1.CLK = CLOCK;
OB2.CLK = CLOCK;
OA0.CLK = CLOCK;
OA1.CLK = CLOCK;
OA2.CLK = CLOCK;
OA3.CLK = CLOCK;
OA4.CLK = CLOCK;
FF0.CLK = CLOCK;
FF1.CLK = CLOCK;
FF2.CLK = CLOCK;
FF3.CLK = CLOCK;
FF4.CLK = CLOCK;
FF5.CLK = CLOCK;
"Flip-Flops are SET when PRESET is HIGH"
OB0.SET = /PRESET;
OB1.SET = /PRESET;
OB2.SET = /PRESET;
OA0.SET = /PRESET;
OA1.SET = /PRESET;
OA2.SET = /PRESET;
OA3.SET = /PRESET;
OA4.SET = /PRESET;
FF0.SET = /PRESET;
FF1.SET = /PRESET;
FF2.SET = /PRESET;
FF3.SET = /PRESET;
FF4.SET = /PRESET;
FF5.SET = /PRESET;
@INPUT VECTORS
```

```
                 _______   _____   _______
                |       \_/     \_/       |
                |                         |
CLOCK [ 1|CLK                      VCC|28]
  RB2 [ 2|I7                        I8|27]
  RB1 [ 3|I6                        I9|26]
  RB0 [ 4|I5                       I10|25]
  RA4 [ 5|I4                       I11|24]
  RA3 [ 6|I3                       I12|23]
  RA2 [ 7|I2                       I13|22]
  RA1 [ 8|I1                       I14|21]
  RA0 [ 9|I0                       I15|20]
  OB2 [10|F7                 PR/OE_|19] PRESET
  OB1 [11|F6                        F0|18] OA0
  OB0 [12|F5                        F1|17] OA1
  OA4 [13|F4                        F2|16] OA2
      [14|GND                       F3|15] OA3
         |                           |
         |___________________________|
```

a. SNAP Equations

Figure 7.

```
" PIN LABELS & BOOLEAN EXPRESSIONS USED IN TRANSITIONS "
@OUTPUT VECTORS
[OB2, OB1, OB0, OA4, OA3, OA2, OA1, OA0]srffs
QA0 = FEh ;
QA1 = FDh ;
QA2 = FBh ;
QA3 = F7h ;
QA4 = EFh ;
QB0 = DFh ;
QB1 = BFh ;
QB2 = 7Fh ;
NOGRANT = FFh ;
@STATE VECTORS
[ FF5, FF4, FF3, FF2, FF1, FF0 ]srffs
W0  = 03Fh ;
W1  = 03Eh ;
W2  = 03Dh ;
W3  = 03Ch ;
W4  = 03Bh ;
W04 = 111---b ;
GA0 = 06h ;
GA1 = 07h ;
GA2 = 0Eh ;
GA3 = 0Fh ;
GA4 = 16h ;
GB0 = 101---b ;
GB1 = 110---b ;
GB2 = 100---b ;
```

b. SNAP Equations (Continued)

Figure 7 (Continued)

```
@TRANSITIONS
WHILE [W0]
  CASE
    [/RA0] :: [GA0] WITH [QA0]
    [/RA1 * RA0] :: [GA1] WITH [QA1]
    [/RA2 * RA1 * RA0] :: [GA2] WITH [QA2]
    [/RA3 * RA2 * RA1 * RA0] :: [GA3] WITH [QA3]
    [/RA4 * RA3 * RA2 * RA1 * RA0] :: [GA4] WITH [QA4]
  ENDCASE
WHILE [W1]
  CASE
    [/RA1] :: [GA1] WITH [QA1]
    [/RA2 * RA1] :: [GA2] WITH [QA2]
    [/RA3 * RA2 * RA1] :: [GA3] WITH [QA3]
    [/RA4 * RA3 * RA2 * RA1] :: [GA4] WITH [QA4]
    [/RA0 * RA4 * RA3 * RA2 * RA1] :: [GA0] WITH [QA0]
  ENDCASE
WHILE [W2]
  CASE
    [/RA2] :: [GA2] WITH [QA2]
    [/RA3 * RA2] :: [GA3] WITH [QA3]
    [/RA4 * RA3 * RA2] :: [GA4] WITH [QA4]
    [/RA0 * RA4 * RA3 * RA2] :: [GA0] WITH [QA0]
    [/RA1 * RA0 * RA4 * RA3 * RA2] :: [GA1] WITH [QA1]
  ENDCASE
WHILE [W3]
  CASE
    [/RA3] :: [GA3] WITH [QA3]
    [/RA4 * RA3] :: [GA4] WITH [QA4]
    [/RA0 * RA4 * RA3] :: [GA0] WITH [QA0]
    [/RA1 * RA0 * RA4 * RA3] :: [GA1] WITH [QA1]
    [/RA2 * RA1 * RA0 * RA4 * RA3] :: [GA2] WITH [QA2]
  ENDCASE
WHILE [W4]
  CASE
    [/RA4] :: [GA4] WITH [QA4]
    [/RA0 * RA4] :: [GA0] WITH [QA0]
    [/RA1 * RA0 * RA4] :: [GA1] WITH [QA1]
    [/RA2 * RA1 * RA0 * RA4] :: [GA2] WITH [QA2]
    [/RA3 * RA2 * RA1 * RA1 * RA4] :: [GA3] WITH [QA3]
  ENDCASE
WHILE [W04]
  CASE
    [/RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB0] WITH (QB0)
    [/RB1 * RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB1] WITH [QB1]
    [/RA2 * RB1 * RB0 * RA4 * * RA3 * RA2 * RA1 * RA0] :: [GA2] WITH [QB2]
  ENDCASE
WHILE [GA0]
  IF (RA0] THEN [W1] WITH [NOGRANT]
WHILE [GA1]
  IF (RA1] THEN [W2] WITH [NOGRANT]
WHILE [GA2]
  IF (RA2] THEN [W3] WITH [NOGRANT]
WHILE [GA3]
  IF (RA3] THEN [W4] WITH [NOGRANT]
WHILE [GA4]
  IF (RA4] THEN [W0] WITH [NOGRANT]
WHILE [GB0]
  IF (RB0] THEN [GB1] WITH [NOGRANT]
WHILE [GB1]
  IF (RB1] THEN [GB2] WITH [NOGRANT]
WHILE [GB2]
  IF (RB2] THEN [GB0] WITH [NOGRANT]
```

b. SNAP Equations (Continued)

Figure 7 (Continued)

Minimize metastability in 50MHz state machines AN032

Engineers are excited to discover the PLUS405–55, a PLD state machine IC rated for a maximum operating frequency of 55MHz. It has a flexible architecture offering 65 product terms, and a programmable OR array driving 16 J-K flip-flops, 8 of which are buried (see Figure 1). This design allows the 64 product terms to realize 64 state transitions in a general state machine implementations. (State machines based on a counter will be implemented much more efficiently.) In order to estimate if a particular state machine will fit in the PLUS405, one need only count the state transitions and assure that there are fewer than 65! There are the remaining issues of number of states, inputs and outputs. The PLUS405 has 8 buried registers, allowing representation of 256 unique states. A dual complement array is available for the "ELSE" condition of state equations, and along with dual clocking capabilities allows two independent state machines to be synthesized on one IC.

Ease of design is further enhanced by SNAP, the PC-based PLD development tool. SNAP supports Boolean and State Equation entry of the design, simulation, and downloading of the programming information to a programmer. SNAP allows an abstract approach to design with PLDs, as the target device is not specified by the engineer until he is done fully integrating and simulating his efforts. After device selection, SNAP can back-annotate the design files with target silicon characteristics, allowing simulation of the actual device.

The engineer sets out to solve all his high speed state machine design problems armed with this new silicon and software, only to discover all this performance has its price. Studying the data sheet on the PLUS405–55 shows the following performance:

f_{MAX1}	55.6MHz minimum
Input Setup time	10ns minimum
Input Hold time	0ns minimum

The cycle time at 55MHz is roughly 18.2 nsec. The window during which data must be stable to guarantee no metastability is 10 nsec long. The difference between the setup and hold time, and the cycle time, is the allowed time interval for changes to occur. this example leaves 8.2ns for any changes.

From a system standpoint, this means the design engineer must be extremely careful in implementing his system, or he will violate the setup and hold specifications of the PLUS405–55. This can lead to metastable conditions in the state machine with several negative effects:

1. Jumps to undefined states. (May hang up system!)
2. Lengthened clock to Q times (slows down!).
3. Jumps to states out of proper sequence.

All of the above problems will yield a system that is unreliable, unpredictable and expensive in terms of servicing elusive bugs in the field.

The preceding analysis said nothing regarding asynchronicity. It is feasible to design the above system in a fully synchronous manner and have acceptable results. What about the system where known asynchronous inputs will be used in the state machine? Examples of common asynchronous signals are refresh request in DRAM controller applications and interrupts in a real-time control system. One approach to managing asynchronous inputs is to precede the state machine with a D-type flip-flop. This can serve as a synchronizing stage . . . or can it??? A simple analysis will explore the feasibility of using a simple synchronizing flip-flop.

A common Dual-D flip-flop frequently selected for this application in TTL high-performance systems is the FAST 74F74. The asynchronous data is fed into the D input of the flip-flop, and the Q output is fed into the logic input of the PLUS405 state machine. A common clock is used for both parts (see Figure 2). Based on current published data sheets, the 74F74 has a clock-to-Q time of 9.2ns maximum. The worst case setup time on the PLUS405–55 is 10ns. The minimum cycle time of the combined system is (9.2 + 10) ns, yielding a maximum clock frequency of 52MHz. Let's assume for this example a desired system clock is 50MHz.

Minimize metastability in 50MHz state machines AN032

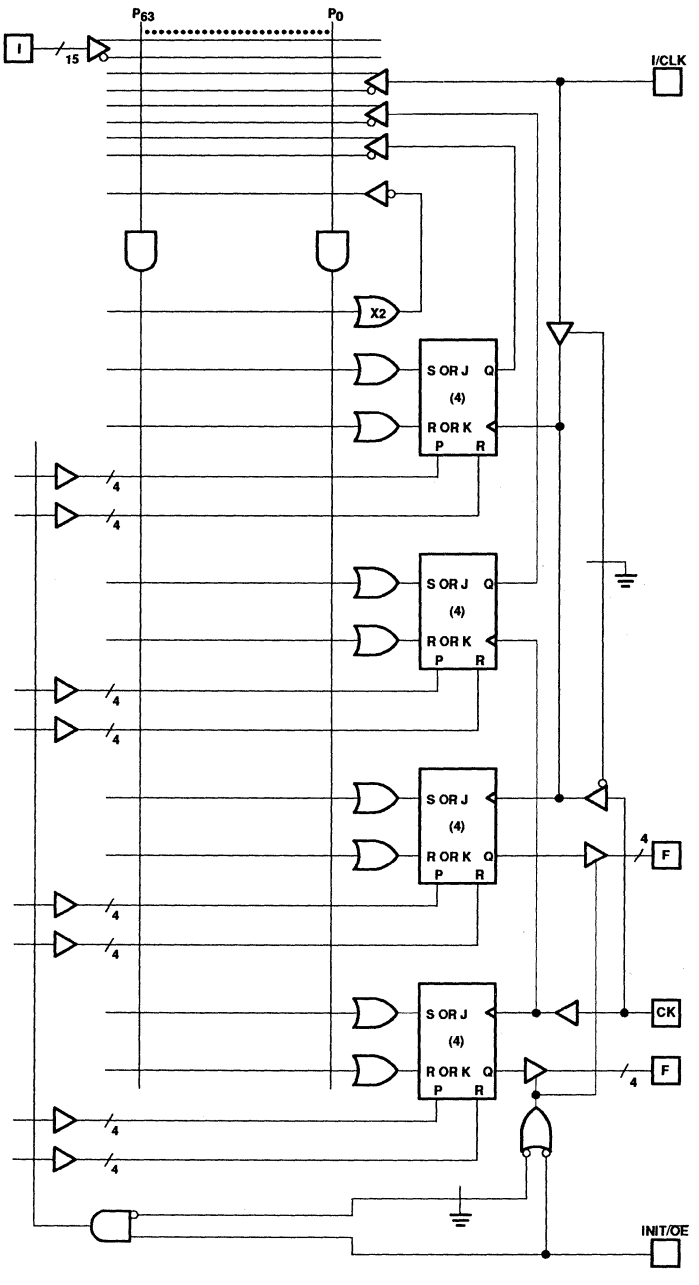


Figure 1. PLUS405–55 Functional Diagram

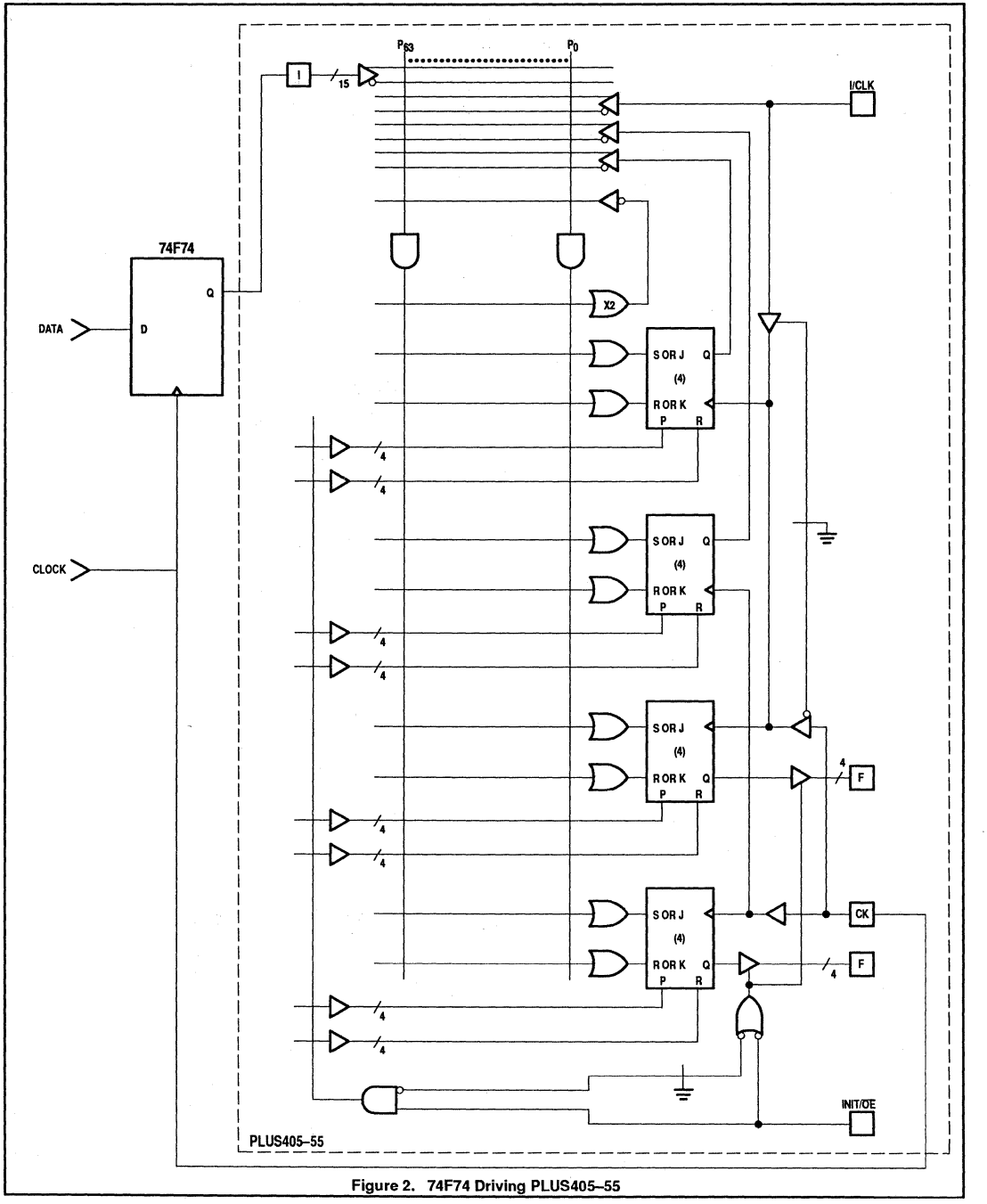


Figure 2. 74F74 Driving PLUS405–55

Minimize metastability in 50MHz state machines

AN032

The important issue to examine in the timing diagram is the time that elapses between the end of the 74F74 clock-to-Q interval and the beginning of the PLUS405's setup time (see Figure 3). This is 20ns minus 10ns minus 9.2ns, which equals .8ns! At 50MHz, this combination is just able to work reliably on a worst case basis, assuming no instances of metastability. If metastable operation is encountered, the 800 picosecond window is the only time left in the clock cycle to resolve the situation. The next issue to examine is the mean time between failures (MTBF) for this system. From the work of Mr. Chaney, an equation which models metastable behavior is:

EQUATION 1. $MTBF = \exp(\tau r/\tau) / (T0 * f * a)$

{Explanation of above symbols}

MTBF is mean time between failures, in seconds.

τr is the elapsed time before sampling the process

or

the time allotted for metastability to resolve.

τ is the "Metastability Time Constant".

T0 in seconds, the zero intercept of aperture time versus propagation delay. T0 indicates the propensity of a device to enter the metastable state.

f is the clock speed, in Hertz.

a is the transition rate of data being sampled (i.e., edges per second) in Hertz.

Assume for this discussion that the asynchronous input data is roughly 2MHz, meaning the edges that can cause metastability occur at a 4MHz rate. The system clock is assumed to be 50MHz, and the elapsed time before sampling is 10ns. (The sample time is calculated from the difference between the cycle time (1/50MHz = 20ns) and the setup time of the PLUS405–55 (10ns). The other parameters can be determined from measurements of an 'F74, or can be found in Mr. Chaney's paper. τ was found to be .4ns and T0 .2 milliseconds. Armed with a calculator and Equation #1, the MTBF for this particular scenario is calculated:

$MTBF = \exp(10/.4) / (.2e{-}3 * 50e6 * 4e6) =$
1.8 seconds

Clearly, this level of failure in any system is unacceptable. A better solution for this class of problem must be found!

Philips Semiconductors has recently introduced a new family of parts designed with metastability performance in mind. The first four members of this family are the 74F5074, 74F50109, 74F50728 and 74F50729. These are excellent general purpose flip-flops, but special attention has been paid to short setup and hold times, and fast clock-to-Q times. The output stage has also been designed with a balanced drive characteristic, leading to tight matching between rise and fall propagation delays, and matching of skews between other outputs. this makes them useful in clock driver applications also. Let's repeat the former calculation using the measured τ and T0 values for the 74F5074 used as a synchronizer (see Figure 4) ahead of the PLUS405–55.

$\tau = .135ns$
$T0 = 9.8\ E\ 6$ seconds

$MTBF = \exp(10/.135) / (9.8e6 * 50e6 * 4e6) =$
75.46 e9 seconds

NOTE: For the reader's reference, a century is 3.154 e9 seconds.

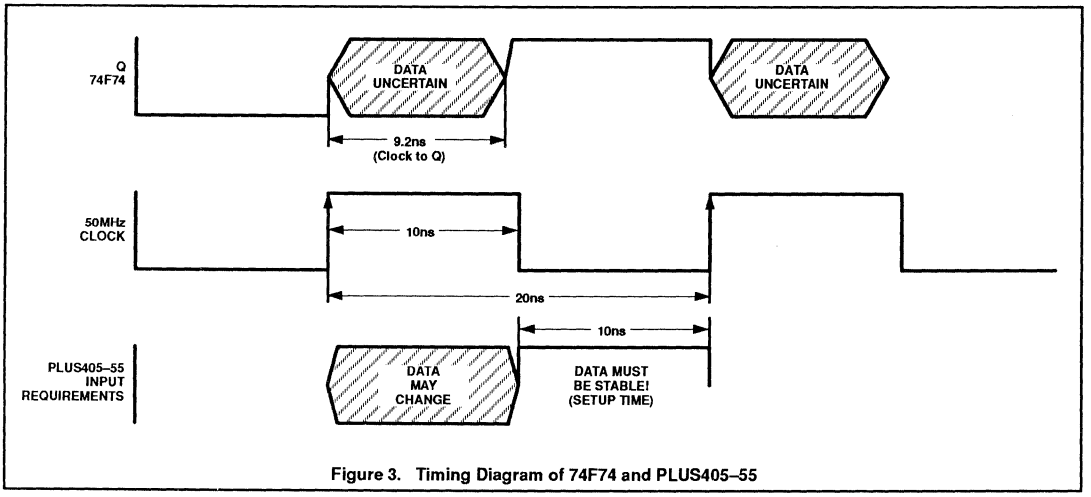


Figure 3. Timing Diagram of 74F74 and PLUS405–55

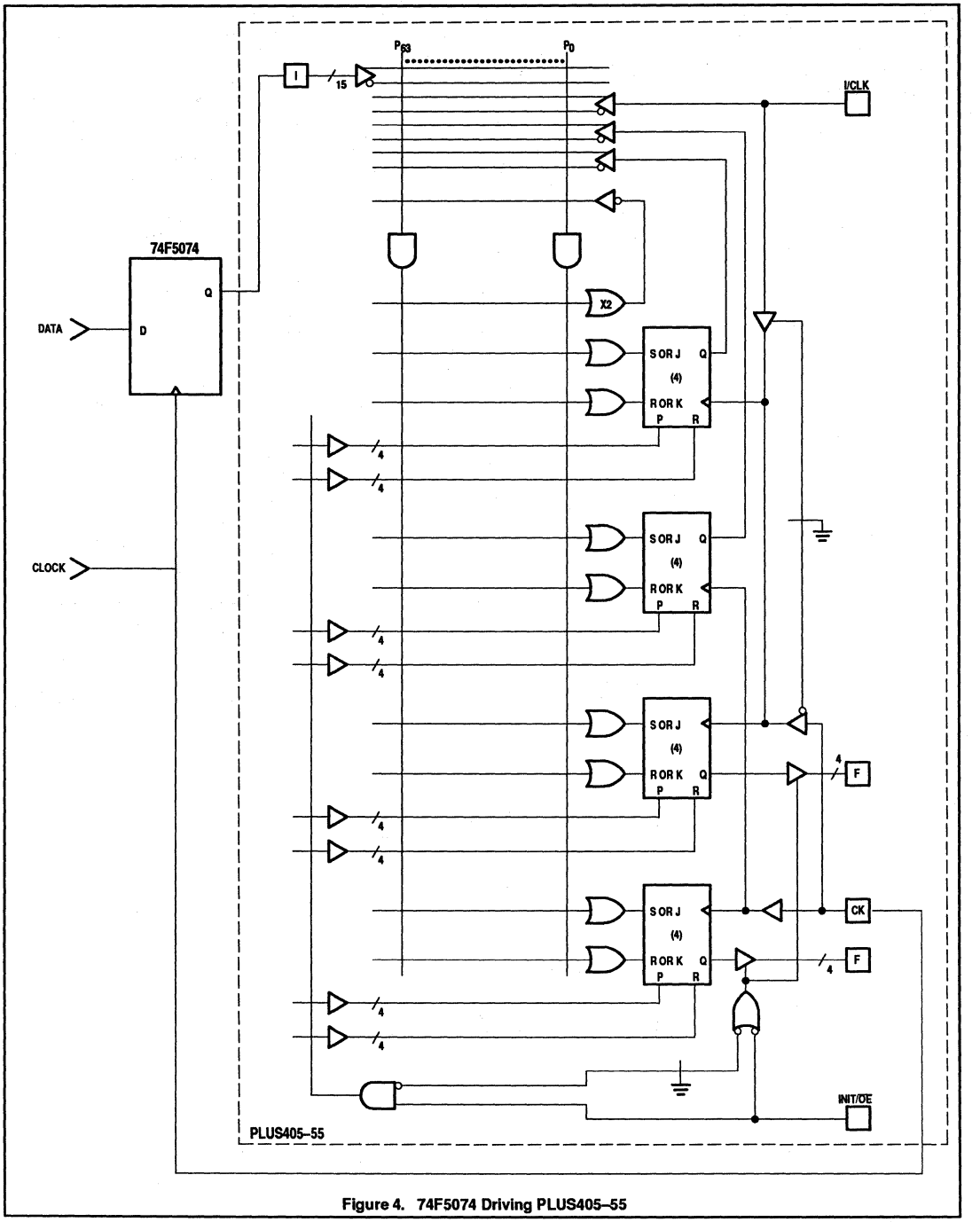


Figure 4. 74F5074 Driving PLUS405–55

Minimize metastability in 50MHz state machines AN032

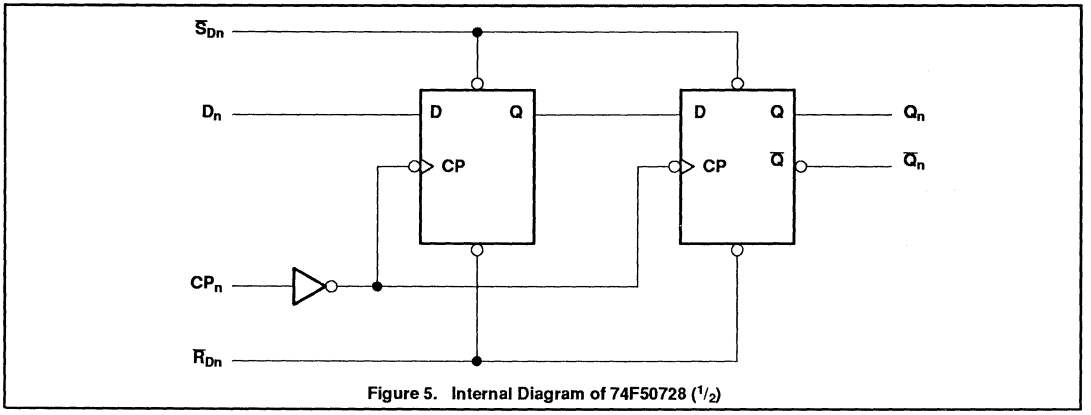


Figure 5. Internal Diagram of 74F50728 ($^1/_2$)

A system that was unreliable is now found to be quite acceptable by using the 74F5074. The major drawback to the synchronizing flip-flop solution is the added delay on the asynchronous signal before it enters the state machine. In the case of the 74F5074, this will amount to one clock cycle delay. For designs that demand the maximum in freedom from metastability, Philips Semiconductors has developed a product with cascaded D flip-flops for synchronizing applications. The 74F50728 (see Figure 5) will therefore introduce a two clock cycle delay into the system. It is pin compatible with the 74F5074 and 74F74 to allow retrofits on existing systems.

Calculation for the MTBF of a system using the 74F50728 is similar to the technique used earlier. In this case though, at least one entire clock cycle is used to resolve any metastability.

EQUATION 2. MTBF = $\exp(\tau r/\tau) / (T0 * f * a)$

{Explanation of above symbols}

All symbols are the same as EQUATION 1 with the exception of τr.

τr is the elapsed time before sampling the process or the time allotted for metastability to resolve. In the case of 74F50728, one entire clock cycle.

The flip-flops embedded in the 'F50728 are essentially the same as the flip-flops used for the 'F5074, therefore the same "Metastability Time Constant" τ, and T0, can be used in the calculation.

MTBF = exp(20/.135)/ (9.8e6 * 50e6 * 4e6) = 1.12 e43 seconds!

Now that the designer is comfortable with handling metastability, it is feasible to begin approaching the design of the system by stating a goal for MTBF and adjusting the state machine's clock to meet the desired failure level.

Let's assume out system is to have an MTBF of 5 years from metastability induced anomalies. The calculations would proceed as follows, assuming the same 2MHz data rate from our previous example:

MTBF = 5 years * (31.54 e6 seconds/year) = 157.7 e6 seconds

Setting up the equation to find the roots yields:

EQUATION 3. $T(setup)/\tau - 1/(f * \tau) + \ln(T0*a*MTBF*f) = 0$

(T(setup) is the setup time on the PLUS405)

Equation 3 is not solved using algebra, but simple numerical methods will allow easy solution, especially since we have a good initial guess for the value of f. (50 to 55MHz!) An HP 32S calculator was used to find the root of this equation by the following program:

				COMMENTS
	PRGM			start program entry
	GTO . .			go to top of memory
B01	LBL/RTN {LBL}		B	label program as B
B02	INPUT	A		a, Data rate, Edges/Sec.
B03	INPUT	F		Clock frequency, Hertz
B04	INPUT	J		τ, seconds
B05	INPUT	M		MTBF, seconds
B06	INPUT	T		T0, seconds
B07	INPUT	U		T(setup), seconds
B08	RCL	A		begin calculation of 1n argument
B09	RCL x	T		
B10	RCL x	M		
B11	RCL x	F		
B12	LN			
B13	RCL	F		
B14	RCL x	J		
B15	1/x			
B16	+/–			change sign
B17	+			add
B18	RCL	U		
B19	RCL ÷	J		divide
B20	+			
B21	LBL/RTN {RTN}			end, return from routine

To execute this program we must use the SOLVE capability on the calculator.

SOLVE {FN}	FN=		Prompt for label of function
B			
50 E 6	STO	F	load initial guess 50MHz
SOLVE {SOLVE}	SOLVE		prompt for unknown variable
F			frequency in this case!
A?	4.0 E 6		set edge rate
R/S			run
J?	135 E –12		set τ
R/S			
M?	157.7 E 6		set MTBF
R/S			
T?	9.8 E 6		set T0
R/S			
U?	10. E –9		setup time
R/S			

Minimize metastability in 50MHz state machines AN032

At this point the calculator will set off to find the root based on the initial guess and the desired conditions entered. The system clock speed determined from this technique is 52.16MHz.

Designers who are forced to deal with an uncertain system for the first time are uncomfortable with the idea that is is possible for the system to fail. Lower speed systems have been traditionally designed using worst case data sheet numbers to guarantee that the system will always work. As system clock speeds cross over 50MHz, meeting the setup and hold times becomes very difficult for TTL-based designs. The allowed time to resolve metastability gets shorter and the data stream edges become much more frequent, increasing the incidence of metastability. Faster systems demand that a design methodology based on statistics be used and the burden is now on the Engineer to manage likelihood of failure to acceptable levels. Persons defining high performance systems will need to specify goals for MTBF due to metastability. Usage of parts that have been characterized for metastability behavior will become mandatory in future systems. New parts, such as the 74F5074 and 74F50728 from Philips Semiconductors, which have published metastable traits and are pin compatible with other industry standard ICs, can make solving these problems as easy as plugging in a new part!

Dike, Charles, *"AN219, A Metastability Primer"*, Philips Semiconductors

Chaney, Thomas J., *"Measured Flip-Flop Responses to Marginal Triggering"* [*IEEE Transactions on Computing*, Vol. C-32, No. 12, December 1983, pp. 1207-1209]

Wakerly, John, *"Designers Guide to Synchronizers and Metastability, Part 1 and 2"* September 1987, *VLSI Design*

Implementing Counters in Sequencer Devices AN050

INTRODUCTION

Some state machine applications require a state machine to wait for a number of clock pulses to occur before some decision point is reached. One common example of this is a state machine that needs to analyze only certain bits in a serial data stream. The state machine may have to wait for a number of serial data bits to transpire before pulsing a load signal or proceeding into states to actually check individual data bits for specific preamble or header information.

SEQUENCER ARCHITECTURE

State machine implementations using JK flip–flop based sequencer devices are generally very efficiently implemented because product terms (AND gates) are required only to force a transition from one state to the next. Product terms are not required to hold the sequencer in a state, as they are for D–type based devices. A JK based state machine can wait forever in one state for a specific parallel combination of input signals to happen, using only one product term to perform the comparison and force a jump to a new state. In addition, Philips sequencers have a PLA architecture meaning that both the AND array and the OR array have programmable connections. A single product term may be connected to the inputs of multiple state and/or output registers. This feature allows for efficient device resource utilization since any product term may be connected to any buried or output register. The product terms are not fixed in their usage to a specific register or output.

DESIGN METHODS

PLD software packages, such as Philips SNAP, provide for different methods of design entry. The easiest and usually best format for state machines is, of course, a state equation entry method. Figure 1 shows an example using state equations. For JK based sequencers, SNAP essentially translates each 'IF' statement into a product term in the device. An OR function in the input condition field of the 'IF' statement will cause an additional product term to be used.

A series of unconditional transitions to a new state may be found in some state machine designs where it is required to wait a certain number of clock cycles before performing a function. The example in Figure 1 shows a simple state machine that runs continuously through sixteen states and outputs a pulse on output 'OUT1' while in state 'F'. This state machine is not waiting for any inputs, other than the clock to occur. It is simply a counter.

COUNTER IMPLEMENTATION

For typical state machine implementations with conditional transitions between states, state equations produce efficient state machines. However for implementing counters, state equations may not produce the most efficient implementation. JK flip-flops have a feature whereby if both J and K are active, after a clock, the output will toggle or change state. This feature may be used to implement counters very efficiently. Combining the toggle feature of the flip-flops with a PLA devices ability to connect a single product term to multiple OR array inputs, produces an implementation where only one product term is needed for each bit in the counter. A four bit counter may be constructed using only four product terms!

The function described in Figure 1 is duplicated in Figure 2, except the Figure 2 design uses a counter described with Boolean equations. Only six product terms were used compared to the sixteen used for the design in Figure 1. Four product terms were used for the counter and two to control output pin OUT1. So, when a portion of a state machine design is required to unconditionally transition from one state to the next, consider implementing a counter using Boolean equations and merging it into the state machine.
The example in Figure 2 only used Boolean equations, no state equations. So, another example is shown in Figure 3. This example, using SNAP, illustrates the proper syntax for connecting the outputs of a counter to the inputs of a state machine. This example was complied for a PLUS405 device. The state machine will wait in each state until the counter reaches a specified value. It then transitions to the next state.

Complicating the design a bit more, Figure 4 shows another SNAP example. This is a listing of a design that, in addition to using the counter outputs as inputs to the state machines, connects two outputs of the state machine back to the counter. The outputs of the state machine (actually two of the state bits) can enable or disable counting, or reset the counter. In this example the state vectors were specially assigned such that state register S1 must be LOW for the counter to count. When state register S2 is HIGH, the counter will be reset. Instead of using state registers bits, additional outputs could have been defined and connected to the counter.

Figure 5 shows a counter that counts from 0 to 12 and then resets. This example may be easily modified to produce a counter that counts to any value.

SUMMARY

The toggle feature of JK flip-flops together with a product term sharing capability, found in most Philips sequencer devices, may be used to build counters using only one product term per counter bit. If a state machine design contains many unconditional transitions, it is possible to reduce the number of product terms required to implement the design by separating the design into a counter and state machine. The counter portion should be described using Boolean equations, when state equations are preferred for the state machines. The counter outputs may be used as inputs to the state machine and some state machines outputs or state bits may be used to enable or reset the counter.

```
@PINLIST
clk  i;
init i;
outl o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

s[3..0].rst = /init; "Use INIT function pin (19) to reset counter"
outl.rst    = /init; "and to reset output pin."

@INPUT VECTORS
@OUTPUT VECTORS
[outl]jkffr
o0 = 0b;
o1 = 1b;

@STATE VECTORS
[s3,s2,s1,s0]jkffr
st0 = 0000b;
st1 = 0001b;
st2 = 0010b;
st3 = 0011b;
st4 = 0100b;
st5 = 0101b;
st6 = 0110b;
st7 = 0111b;
st8 = 1000b;
st9 = 1001b;
sta = 1010b;
stb = 1011b;
stc = 1100b;
std = 1101b;
ste = 1110b;
stf = 1111b;

@TRANSITIONS

while [st0]
 if [] then [st1]
while [st1]
 if [] then [st2]
while [st2]
 if [] then [st3]
while [st3]
 if [] then [st4]
while [st4]
 if [] then [st5]
while [st5]
 if [] then [st6]
while [st6]
 if [] then [st7]
while [st7]
 if [] then [st8]
while [st8]
 if [] then [st9]
while [st9]
 if [] then [sta]
while [sta]
 if [] then [stb]
while [stb]
 if [] then [stc]
while [stc]
 if [] then [std]
while [std]
 if [] then [ste]
while [ste]
 if [] then [stf] with [o1]
while [stf]
 if [] then [st0] with [o0]
```

Figure 1. SNAP State Equations

```
@PINLIST
clk i;
init i;
out1 o;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

"Simple four bit binary counter that"
"uses toggle feature of JK flip-flops."
"Because of p-term sharing, only 4 p-terms"
"are needed to implement this counter."

c0.j = 1;
c0.k = 1;
c1.j = c0;
c1.k = c0;
c2.j = c0 * c1;
c2.k = c0 * c1;
c3.j = c0 * c1 * c2;
c3.k = c0 * c1 * c2;

c[3..0].rst = /init; "Use INIT function pin (19) to reset counter"
out1.rst    = /init; " and to reset output pin"

"In this example the counter is free-running."
"Out1 will be high when the count is 1111B and"
"will be forced low when the counter transistions"
"from 1111 to 0000 binary or reset by pin 19."

out1.j = c3*c2*c1*/c0;
out1.k = c3*c2*c1* c0;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 2. Counter Boolean Equations

```
@PINLIST
clk i;
init i;
out1 o;
out2 o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

"Simple four bit binary counter"

c0.j = 1;
c0.k = 1;
c1.j = c0;
c1.k = c0;
c2.j = c0 * c1;
c2.k = c0 * c1;
c3.j = c0 * c1 * c2;
c3.k = c0 * c1 * c2;

c[3..0].rst   = /init; "Use INIT function pin (19) to reset counter"
s[1..0].rst   = /init; " and state registers"
out[2..1].rst = /init; " and output pins"

@INPUT VECTORS
@OUTPUT VECTORS
[out1,out2]jkffr
 o0 = 0-b;
 o1 = 1-b;
 o2 = -0b;
 o3 = -1b;

@STATE VECTORS
[s1,s0]jkffr
st0 = 00b;
st1 = 01b;
st2 = 10b;
st3 = 11b;

@TRANSITIONS

               "In this example the counter outputs are used"
               "    as inputs to the state machine"

 while [st0]
  if [c3*c2*c1*/c0] then [st1] with [o1] "move to state 1 when counter goes"
                                         "from E hex to F hex"
 while [st1]
  if [] then [st2] with [o0]             "upon next clock go to state 2 and"
                                         "reset output"
 while [st2]
  if [/c3*c2*c1*/c0] then [st3] with [o3] "wait here until count = 6 hex"
                                          "then go to state 3 and set out2"
 while [st3]
  if [] then [st0] with [o2]              "goto state 0 and reset out2"
```

Figure 3. Counter Connected to State Machine

```
@PINLIST
clk  i;    out1 o;
in1  i;    out2 o;
init i;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

"Four bit binary counter"
"controlled by state machine state register bits"

c0.j = /s1*/s2;                "Counter will be forced to 0000 upon"
c0.k = /s1;                    "clock and state bit s2 high."
c1.j = /s1 * c0 * /s2;         "Counter won't count unless state"
c1.k = /s1 * c0;               "register s1 is low. (won't count"
c2.j = /s1 * c0 * c1 * /s2;    "in state st1)"
c2.k = /s1 * c0 * c1;
c3.j = /s1 * c0 * c1 * c2 * /s2;
c3.k = /s1 * c0 * c1 * c2;

c[3..0].rst   = /init; "Use INIT function pin (19) to reset counter"
s[2..0].rst   = /init; " and state registers"
out[2..1].rst = /init; " and output pins"

@INPUT VECTORS
@OUTPUT VECTORS
[out1,out2]jkffr
 o0 = 0-b;
 o1 = 1-b;
 o2 = -0b;
 o3 = -1b;

@STATE VECTORS
[s2,s1,s0]jkffr
st0 = 000b;       "Note the special state assignments to"
st1 = -10b;       "simplify one state bit connections to"
st2 = -01b;       "the counter."
st3 = 111b;

@TRANSITIONS

               "In this example the counter outputs are used"
               "as inputs to the state machine and some of the"
               "state register bits S2 and S1 control the operation"
               "of the counter."

 while [st0]
  if [/c3*c2*/c1*c0] then [st1] with [o1] "move to state 1 when counter goes"
                                          "from 5 hex to 6 hex"
 while [st1]
  if [in1] then [st2] with [o0]    "when input 'in1 = high' go to state 2 but"
                                   "hold counter at 6 while waiting for in1"
 while [st2]
  if [c3*/c2*/c1*/c0] then [st3] with [o3] "wait here until count = 8 hex"
                                           "then go to state 3 and set out2"
 while [st3]
  if [] then [st0] with [o2]               "goto state 0 and reset out2"
                                           "and counter"
```

Figure 4. Counter Enable and Reset Functions Controlled

```
@PINLIST
clk i;
ignd i;
init i;
outl o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

"Four bit binary counter"
"modified to count from"
"0 to 11 and then reset."

c0.j = nor;
c0.k = nor;
c1.j = nor * c0;
c1.k = nor * c0;
c2.j = nor * c0 * c1;
c2.k = (nor * c0 * c1) + count12;
c3.j = nor * c0 * c1 * c2;
c3.k = (nor * c0 * c1 * c2) + count12;

count12 = c3*c2*/c1*/c0;
nor = /(count12+ignd);

"When count=11, then output NOR is LOW, disabling the product terms"
"that cause the counter to count. Another product term (count12)"
"connects to the registers of the counter that are HIGH K-inputs,"
"forcing it to all zeros upon the next clock. These connections may"
"be modified to alter the upper count limit."

c[3..0].rst = /init; "Use INIT function pin (19) to reset counter"
outl.rst    = /init; " and to reset output pin"

"In this example the counter is free-running."
"Outl will be high when the count is 1100B and"
"will be forced low when the counter transistions"
"from 1100 to 0000 binary or reset by pin 19."

outl.j = c3*/c2*c1*c0;
outl.k = c3*c2*/c1*/c0;

"For SNAP 1.90 to implement this design in a minimum number of product
 terms, two passes through the merger are necessary. First, generate a
 netlist normally - running NETCONV and MERGER. Then, highlight equations
 in the MERGER box to extract the equations from the netlist. Run the
 extracted equations through the minimizer (EQNGEN). Run through NETCONV
 (Minimized) and MERGER again to produce a minimized netlist. The design
 may then be compiled for the device."
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 5. Modulo-n Counter

CMOS sequencers

INTRODUCTION

Philips Semiconductors invented the programmable logic sequencer with the 82S105 back in 1979. Since that time, additional parts were introduced, with a wide following of users who got programmable state machines into their designs. But, many potential users are still confused about what a sequencer is. Let us clarify that by simply stating that a sequencer is a programmable logic device capable of making user configurable state machines in a single chip. The first sequencers were configured with a programmable logic array (PLA) connected to a group of flip-flops. For state machine designs, the choice of either S-R or J-K flip-flops was appropriate because of logic efficiency. D flip-flops may be used, just as well, but are less efficient. Also, the restriction of using a PLA may not be mandatory. Many have used fixed-OR structures driving D flip-flops, and found the resulting solutions satisfactory. Hence, the broad definition of a sequencer is simply a programmable logic device with flip-flops. The inclusion of additional features beyond these basics can make all the difference in the world, and will be shown to be quite useful. Among the critical additional features are: A complement array, buried versus exposed flip-flops, independent flip-flop clocking and independent asynchronous set and reset capabilities. As well, something as simple as permitting some flip-flops to be clocked on rising clock edges and others on falling edges can have far-reaching performance implications.

This booklet will look at three parts that are sequencers, but which also have the additional property of being made from CMOS, so they have some low power capabilities that similar bipolar devices do not have. These devices are the PLC18V8Z, the PLC415 and the PLC42VA12. Each of these devices will be presented with a detailed application described that is appropriate to that device. After an initial description of the parts is given, a brief discussion of power-saving techniques is given. Then the example applications are detailed with complete design files which can be run on Philips Semiconductors design software.

The PLC18V8Z

Figure 1 shows the PLC18V8Z logic diagram. As can be seen, there is a large programmable region that can interconnect input lines and feed back logic values to a region where they may be associated with the inputs to AND gates at small fixed-OR sites. This is termed Programmable Array Logic (PAL®). The outputs of the fixed-OR sites then drive into macrocells (the little boxes in Figure 1). The macrocell is detailed in Figure 2, where it is seen that the macrocell consists of one D flip-flop, three multiplexors, an exclusive-OR gate (for polarity control), configuration programmable sites and feedback paths. The output points of the macrocells have access to the chip's output pins. By configuring the macrocell multiplexors, it is possible for a signal coming into the macrocell to be routed (by MUX) to the output pins (from the combinational logic area) to take the flip-flop output to the pins, to feed back the flip-flop to the main logic array, or accept the "F" pin as in input. Because a large number of applications are "byte" oriented, eight macrocells fits in a data oriented system. The number of applications that require no more than 8 product terms per OR gate cluster, is also very large, and includes counters, shifters, pattern recognizers and handshakers. It should also be noted that a special output is dedicated on the macrocell to permit 3-State control of the output pins, from the programmable array.

Designers familiar with generic array logic (GAL™) will appreciate that the PLC18V8Z is intentionally pinned to be directly compatible with the 20-pin 16V8 device. This means that the PLC18V8Z can replace the long list of fixed-OR devices that includes the popular 16L8, 16R8, 16R6, 16R4, etc.

®PAL is a registered trademark of Advanced Micro Devices, Inc.
GAL is a trademark of Lattice Corp.

CMOS sequencers

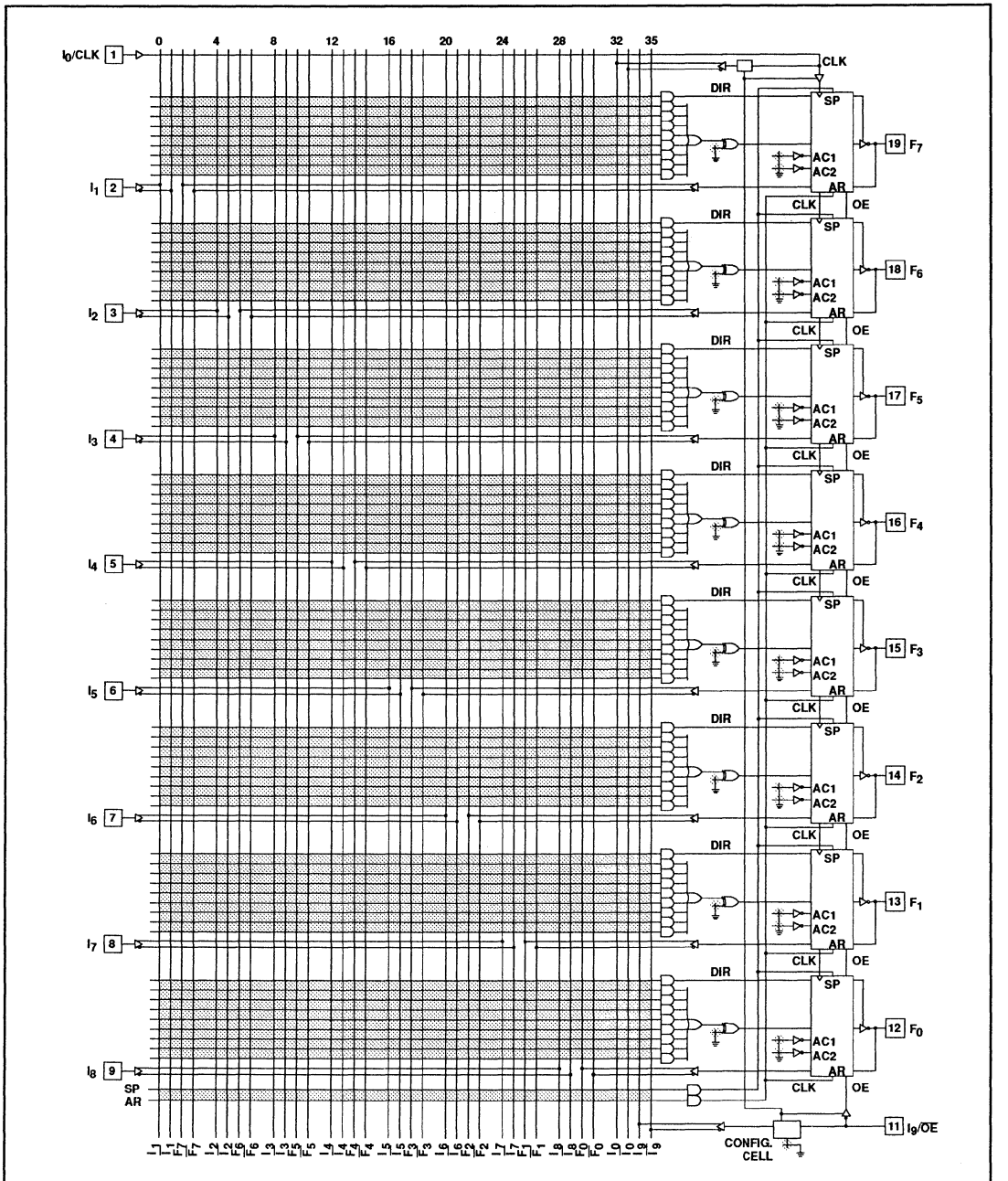


Figure 1. PLC18V8Z

CMOS sequencers

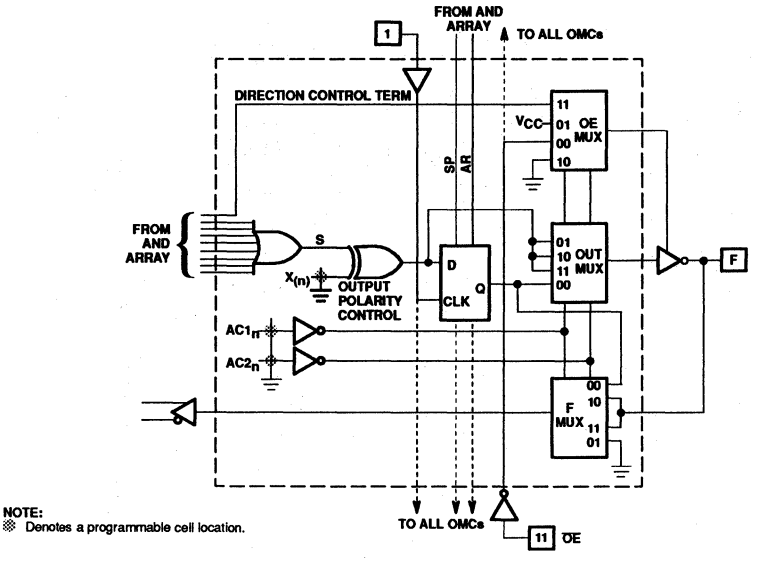


Figure 2. PLC18V8Z Macrocell

CMOS sequencers

The PLC415

The PLC415 is shown in Figure 3, in a shorthand form. The actual part incorporates groups of J-K flip-flops with a programmable logic array. The flip-flops are logically grouped by virtue of association to specific output pins and association with particular clock inputs. The PLC415 illustrates both exposed flip-flops (Q outputs directly tied to the output pins) and buried flip-flops (Q outputs fed directly back to the programmable interconnect region). Sometimes, the outputs of the buried flip-flops are referred to as the state variable register because they capture next-state information to generate the transition signals for driving other flip-flop inputs. It should be noted that the PLC415 associates one group of four buried flip-flops, to a specific group of 4 exposed flip-flops by virtue of common clock inputs. Hence, this group can be thought of as a single, synchronous programmable state machine. The 64 product terms are available to be freely interconnected to any of the flip-flop OR gate inputs. There are no connection restrictions, and complete freedom of sharing. The second state machine can also use any of the AND gates as needed.

One of the powerful features of the PLC415, which is not available on registered fixed-OR devices, is multiple clock source availability. This, coupled with independent 3-State controls, permits some unique inherent output multiplexing capabilities.

Another feature that has been seldom understood, is the "complement term." Basically, the complement term is a NOR gate located in the PLC415 PLA, to permit efficient next-state transitioning. If flip-flop transitions are accomplished with Boolean products (asserted to logical "1" at their outputs), the state machine will transition accordingly. Usually, this is thought of by saying "If the machine is in state X, and an input of Z occurs, then assume the next state of W." What happens if input value Z doesn't occur? Usually, the machine will then stay in state X. But what if it is desired to move to another state if there are no asserted input conditions present? The logic designer is confronted with generating a function that provides the correct, positive asserted transition terms. This consumes lots of product terms, and the designer quickly depletes AND gates. However, by logically combining a product term that decodes the present state, with the missing input combination, a logical product will be generated permitting a next-state transition to occur when the input condition is absent. This product can be sent to the NOR term (i.e., complement term), which generates a logical "1" when all of its inputs are at a logical "0". This NOR output can force a transition from state X to state Y. This is commonly referred to as generating a logical "else" condition. Hence, using the complement term, a state transition may be described as "if state X and input Z, then state W, else state Y." A particularly slick feature of this attribute is that by combining the current state with the queried input into a product term, other product terms can be included in the complement NOR gate, which decode different states and input conditions. The state transition payoff is that only one complement term is really ever needed per state machine, to get the "else transitions" from all possible states. Because the PLC415 can build two independent machines, it includes two complement terms. The Philips Semiconductors design software SNAP automatically use the complement term to perform "else state transitions." Because the complement term requires a signal to pass through the programming array twice before hitting a flip-flop input, there is a small speed penalty to use it. However, there is a major payoff in terms of AND gate usage. This feature has literally become a signature item in most Philips Semiconductors sequencers.

CMOS sequencers

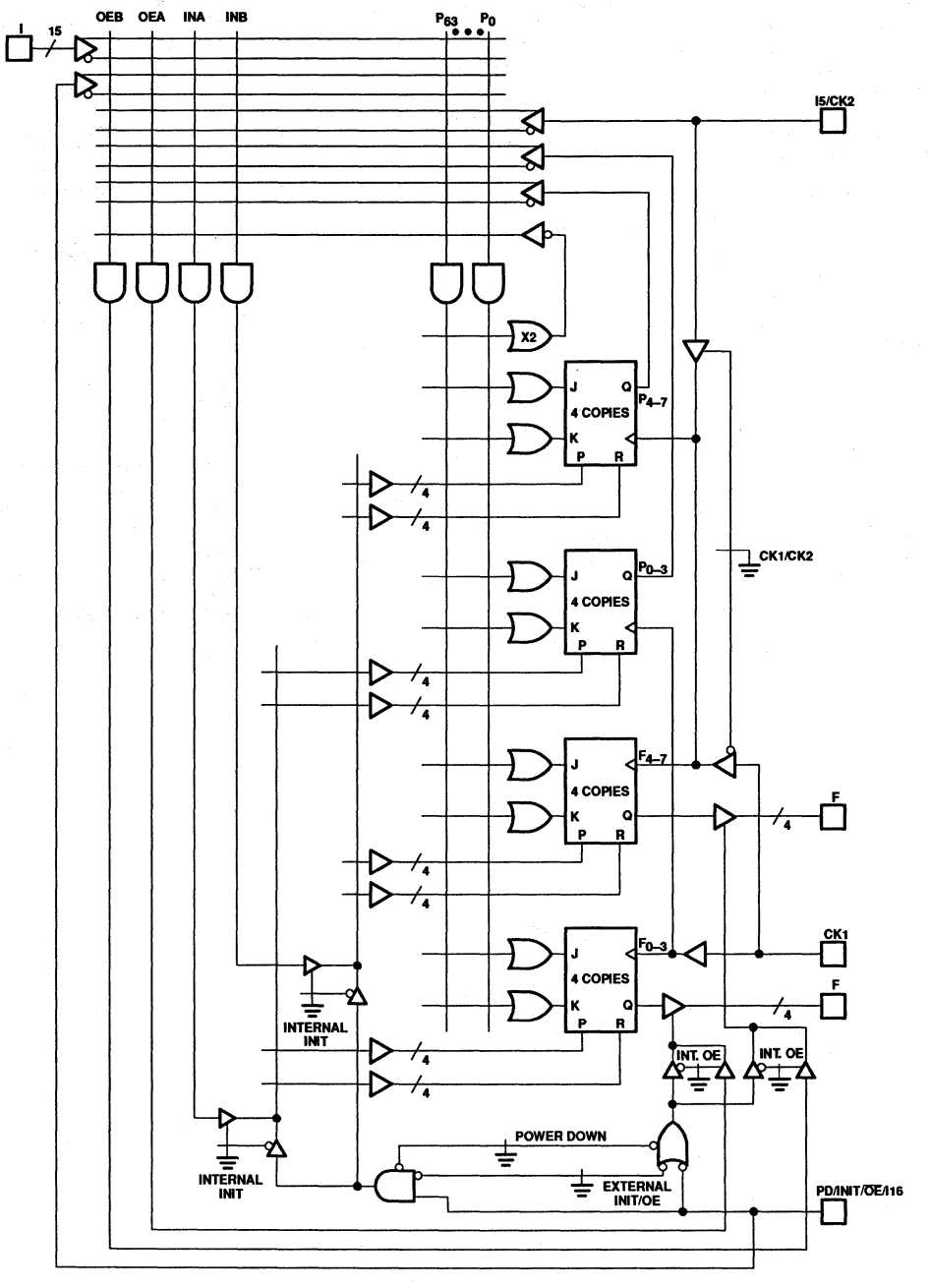


Figure 3. PLC415

CMOS sequencers

The PLC42VA12

The PLC42VA12 was recently (December 1989) heralded as one of the top programmable logic devices of the year by *Electronic Design* magazine. Its architecture solves more standard design problems for a 24-pin device than any other comparable device. The PLC42VA12 is so flexible that designers are permitted nearly unrestricted design freedom. Incorporating 10 J-K flip-flops with a large PLA, the designer is able to clock each flip-flop from a distinct source. Each flip-flop has asynchronous set and clear, and flip-flop outputs either pass to an output pin, back into the programming array, neither, or both. Using an output pin does not force the associated flip-flop function to be lost, and outputs are 3-State controlled in small groups, from either dedicated pins, or from the PLA. The PLC42VA12 also has two combinational outputs. As can be seen in Figure 4, the combinational output points are driven from polarity controlling Exclusive-OR gates. It should be noted that the PLC42VA12 has been designed for compatibility with the popular 22V10 device. Designers that have enjoyed the 22V10, but needed greater flexibility, will appreciate additional freedom in designing with the PLC42VA12.

Functional independence is a key feature of the PLC42VA12. By having separate clocks for each flip-flop, the designer may treat each flip-flop as a separate element. Most PLDs assign a single clock to large groups of flip-flops, which forces the designer to restrict the applications to standard synchronous state machines. The PLC42VA12 permits a designer to build up to 10 (granted, simple) state machines in a single chip. But basically, design freedom is maximized in a PLC42VA12. Additionally, the asynchronous reset and set inputs are carefully partitioned among the flip-flops to minimize restrictive design practices. As usual, the complement term is available for efficient utilization of the "IF-THEN-ELSE" syntax.

CMOS sequencers

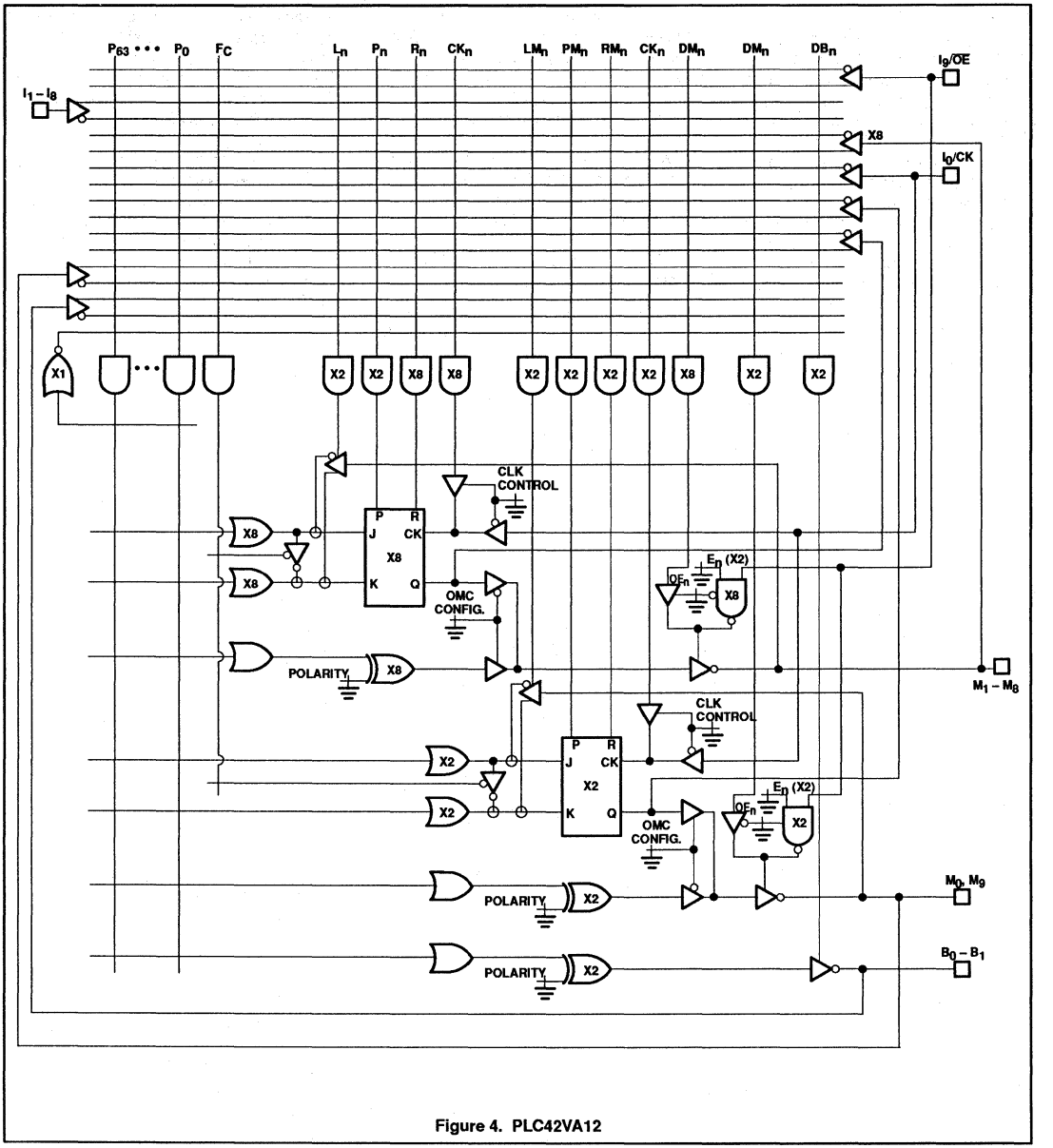


Figure 4. PLC42VA12

CMOS power in PLDs

AN0301

CMOS POWER IN PLDs

When one first thinks of CMOS, zero power is one concept that comes to mind. A CMOS device should consume not power in a DC state, and when switching the power should be proportional to the frequency. This idea seems to make the thought of a zero power CMOS PLD a natural one. Yet, when one examines the CMOS PLD marketplace, only a fraction of the devices claim the mantle of zero power. An understanding of the basic concepts can be vital.

With the exception of two architectures, all CMOS PLDs are constructed using an electrically erasable (EEPROM) or ultraviolet erasable (EPROM) cell similar to that shown in Figure 1. It should be pointed out that a true CMOS E^2 or EPROM memory cell does not exist. The core of almost all CMOS PLDs is an array of NMOS transistors. By wrapping the NMOS core with CMOS I/O cells, the illusion of a CMOS PLD is created (Figure 2). The only fallacy behind this is that NMOS devices consume power in a DC state. The array of NMOS devices are continually fed power to maintain optimum speed. This means that even in a DC state, where the device is not switching, the power level is in the 10's of mA. The main advantage here is that since only the array consumes power, the I_{CC} of the device will be much less than similar bipolar PLDs. If we could "eliminate" the power to the array, the power drops to levels expected from CMOS devices.

There are two classical electronic techniques used to eliminate the I_{CC} to the die core, and Philips Semiconductors uses both of them depending on the part. Both techniques have inherent disadvantages, but lower I_{CC} dramatically. The first method is through the use of a special pin. When correctly asserted, a series blocking transistor(s) that supplies power to the core is turned off. This requires an external signal to control when I_{CC} is to be blocked and the core becomes "asleep" or inoperable.

To wake up the part, the power down condition must be released. Placing the part into and out of power down mode takes time and this impedes performance. Figure 3 illustrates the power saving approach of a dedicated control pin with the specific transaction for the PLC415. Note that PD (Power Down) must be asserted (i.e., logic "1") and released (logic "0") with a specific timing relationship to the clock. If the timing is maintained, the device will power down and power back up in the same state. Should the timing be altered, the internal state may be lost.

From a simple viewpoint, the other method, which is used in Philips Semiconductors 18V8Z, is more convenient. This technique, called "Input Transition Detection" (ITD), has been adapted from MOS memory design. A diagram of the key circuitry is shown in Figure 4. If any input makes either a high-to-low or low-to-high transition, the ITD senses it and sends a pulse to the array supply module. The supply in turn provides power to the array for a period of time long enough that the contents of the array can be latched. This data is now available as inputs into the I/O buffers for further processing. Once the data has been latched, there is no need for the array to consume power. The width of the power pulse is designed so it only need fulfill this function. This period of time, approximately 20ns for the 18V8Z, is much shorter than the cycle time of the device (35ns for the 18V8Z). This means that the array only consumes power for 57% of each cycle. Obviously when longer cycles are used, the percentage is reduced. The rest of the circuitry obeys CMOS rules. The IDT feature gives a device more typical of a traditional CMOS technology. When all inputs are at static CMOS levels, power to the array is turned off so the device consumes a current of less than 100μA. Unlike some zero power PLDs, there is no surge in current once the device becomes active. When switching, the current of the 18V8Z rises at a linear rate which is typically 1mA per MHz.

The key to the technique is the Input Transition Detection circuit. Shown in Figure 5 is a simplified ITD implementation. When any of the inputs makes a transition, the output of the Ex-OR gate, as well as the OR gate, goes high for a period determined by the delay circuit. This "power signal" is what feeds the array supply circuit which eventually powers the array. The width of the delay varies across temperature to compensate for faster speeds at cold temperatures and slower speeds at hot temperatures.

Waveforms for the ITD circuit are shown in Figure 6. When input A makes a low-to-high transition, the power signal goes high for a period of t_{AP} after a delay of t_{DP}. From these waveforms, it is easy to see one of the disadvantages of this method. When several inputs are applied to the device in a system environment, there will be a finite amount of skew between the inputs. Since all inputs are tied to the ITD circuitry, pulses will be generated for each input. An example of this is shown in Figure 7. The power signal is initially triggered by input A. Input B, which is skewed from input A by the time t_{SKEW} generates its own power pulse, which is ORed to the pulse generated by input A. This makes the power pulse last longer in a system environment, which means the device will consume more power than originally anticipated.

A disadvantage of the ITD circuit is a speed penalty. Two factors in the design reduce the speed. Since the array is not continuously powered, some delay is incurred by the ITD circuit to provide array power. The other speed penalty is paid in the data latches. These latches are necessary to store the array contents.

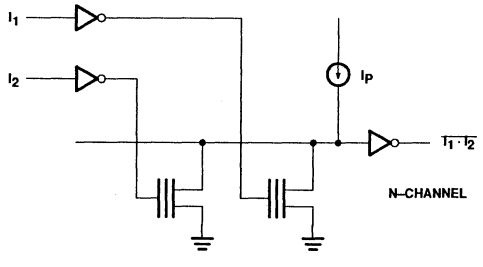


Figure 1. CMOS Programming – The EPLD

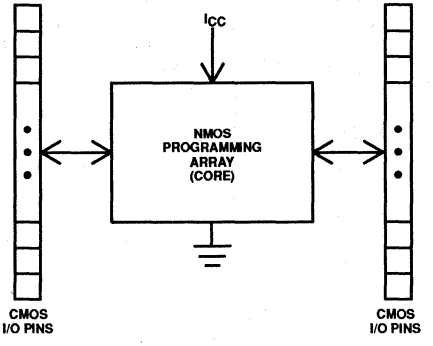


Figure 2. CMOS PLD Layout

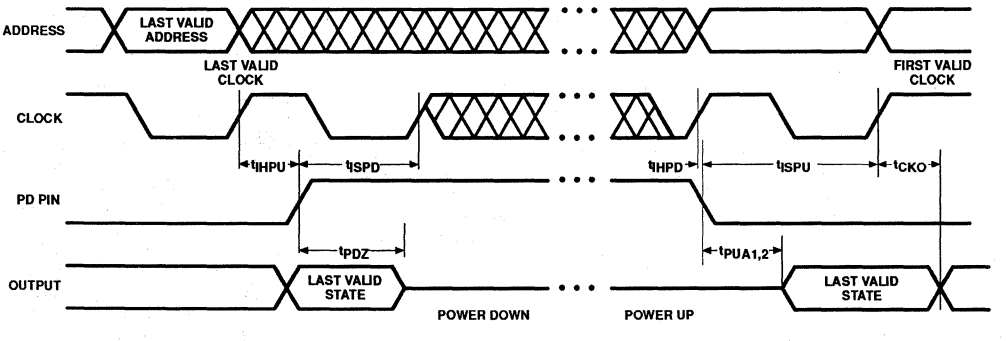


Figure 3. Power Down Enable and Disable

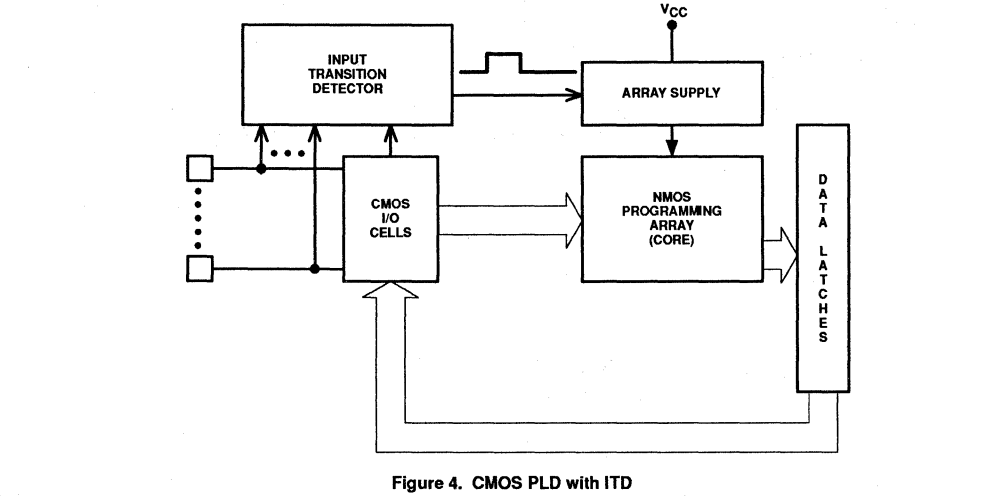


Figure 4. CMOS PLD with ITD

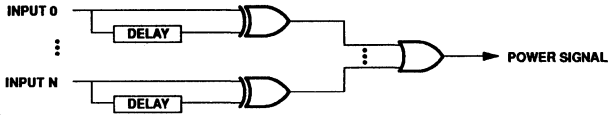


Figure 5. ITD Circuit

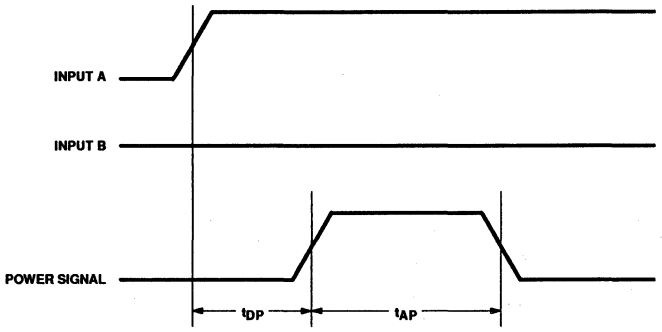


Figure 6. ITD Waveforms

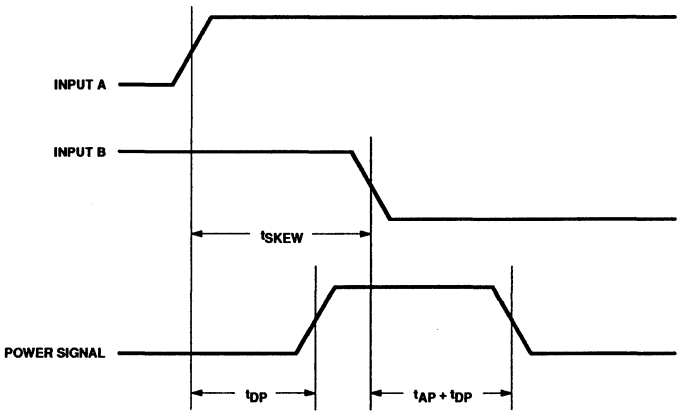


Figure 7. ITD with Skewed Inputs

Microcontroller power management | AN0302

PLC18V8Z APPLICATIONS

The 80C51 microcontroller and its CMOS derivatives have two power reducing modes, Idle and Power Down. The Power Down mode reduces the device's current to less than 50µA by only keeping the on-chip RAM and SFRs data intact. In order to resume operation while in the Power Down mode, it is necessary to apply a reset to the microcontroller.

The PLC18V8Z is in a low power mode whenever its inputs are not switching, drawing less than 100µA. An input transition causes the PLC18V8Z to power up its internal array for a short time, latch a valid output and then return to low power mode. Because of this transparent power reduction feature and its programmability, the PLC18V8Z is an excellent device to use in low power applications with an 80C51 microcontroller.

Two examples of using the PLC18V8Z with a SC87C751 microcontroller are presented. Both applications use the PLC18V8Z to detect events while the SC87C751 is in a Power Down mode and then reset (wake up) the microcontroller. The first example, shown in Figure 1, uses the PLC18V8Z as an 8-bit priority encoder. SNAP pin layout and listing of the circuitry fused inside the PLC18V8Z is shown in Figure 2.

Whenever one of the inputs $I_7 - I_0$ goes LOW, a binary representation of its position is output on pins $A_3 - A_0$. If more than one input is Active-LOW, then the input with the highest priority is represented on the output, where I_7 has the highest priority. Another output, $\overline{EO}$, is not connected to the microcontroller but is used to control the RST output of the PLC18V8Z. $\overline{EO}$ is Active-LOW anytime all inputs are high. Actually, the PLC18V8Z could easily be reprogrammed to output the inverse of this signal which could be tied to the interrupt line of the microcontroller to generate an interrupt anytime one or more inputs were low.

Pin 16 of the PLC18V8Z, labeled RST, is the output of a 3-State buffer whose input is always high. The buffer's control line is tied internally to a product term which is enabled by $\overline{EO}$ and an input from the microcontroller labeled $\overline{RSTEN}$. The RST buffer may be in only two states, either driving a high (resetting the SC87C751) or 3-State (allowing C_3 to discharge), enabling normal operation of the microcontroller. Before entering the Power Down mode, the microcontroller should force $\overline{RSTEN}$ low. Then, any low on $I_7 - I_0$ will cause $\overline{EO}$ and also RST high, resetting the microcontroller. When the microcontroller is reset, it will force its ports to input mode and since P_1 and P_3 have internal pull-up resistors, $\overline{RSTEN}$ will go high forcing RST into the 3-State mode allowing C_3 to discharge.

The second example, shown in Figure 3, with SNAP pin layout and listing in Figure 4, uses the PLC18V8Z to monitor three microcontroller input lines ($I_{NC} - I_{NA}$) and reset the microcontroller upon any change. Three internal registers inside the PLC18V8Z are used to hold the states or levels of the input lines prior to entering Power Down mode. Before entering Power Down mode, the microcontroller should clock into the PLC18V8Z the states of $I_{NC} - I_{NA}$ with the LOAD signal. Comparator logic fused into the PLC18V8Z compares the output of the registers to the three input lines. The RST output of the PLC18V8Z operates in a similar manner to the first example to reset the microcontroller whenever $\overline{RSTEN}$ is low and the output of the comparator is false.

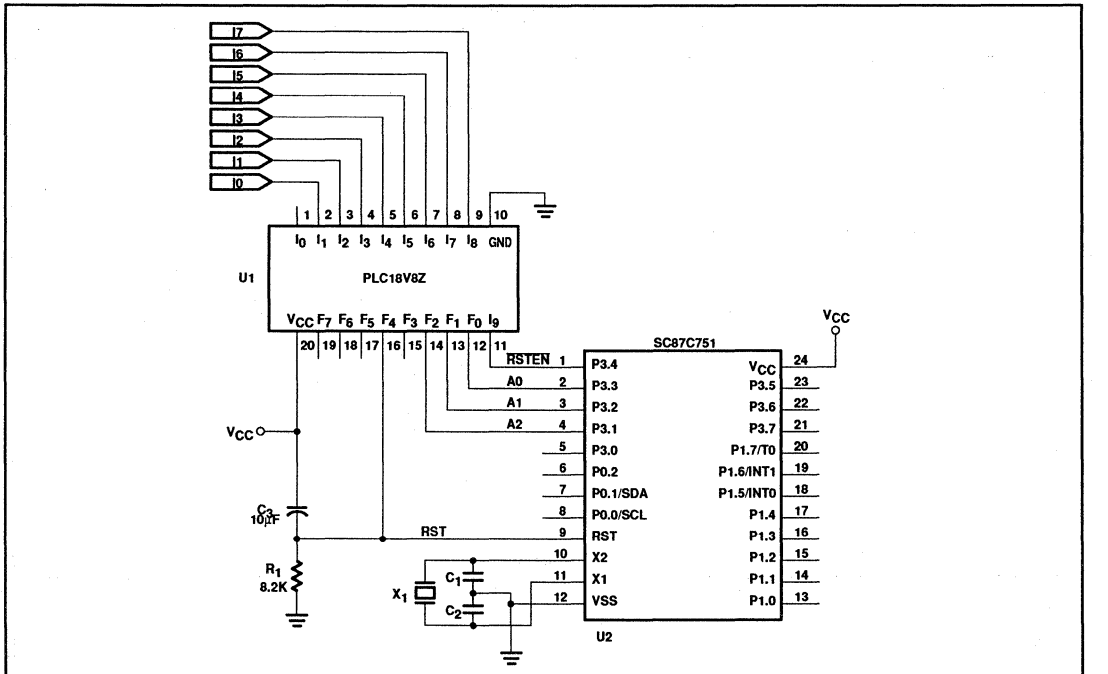


Figure 1.

```
************************************************************
*       PLC18V8Z 20-Pin DIP Package Pin Layout             *
* Date: 10/03/93                          Time: 15:27:18 *
************************************************************

                    _________\_____/_________
                   |                         |
                   |                         |
              [ 1|I0/CLK                  VCC|20]
         I0   [ 2|I1                       F7|19]
         I1   [ 3|I2                       F6|18]
         I2   [ 4|I3                       F5|17]
         I3   [ 5|I4                       F4|16] RST
         I4   [ 6|I5                       F3|15] EO
         I5   [ 7|I6                       F2|14] A2
         I6   [ 8|I7                       F1|13] A1
         I7   [ 9|I8                       F0|12] A0
              [10|GND                  I9/OE_|11] RSTEN
                   |                         |
                   |_________________________|
```

```
@PINLIST
I0 i;I1 i;I2 i;I3 i;
I4 i;I5 i;I6 i;I7 i;
RSTEN i;
A0 o;A1 o;A2 o;
EO o;RST o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

A0   =  /I1*I2*I3*I4*I5*I6*I7
      +      /I3*I4*I5*I6*I7
      +            /I5*I6*I7
      +                  /I7;
A1   =     /I2*I3*I4*I5*I6*I7
      +      /I3*I4*I5*I6*I7
      +               /I6*I7
      +                  /I7 ;
A2   =           /I4*I5*I6*I7
      +            /I5*I6*I7
      +               /I6*I7
      +                  /I7 ;
EO   =  /(I0*I1*I2*I3*I4*I5*I6*I7);
RST  =  1;
RST.OE = /RSTEN*EO;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 2.

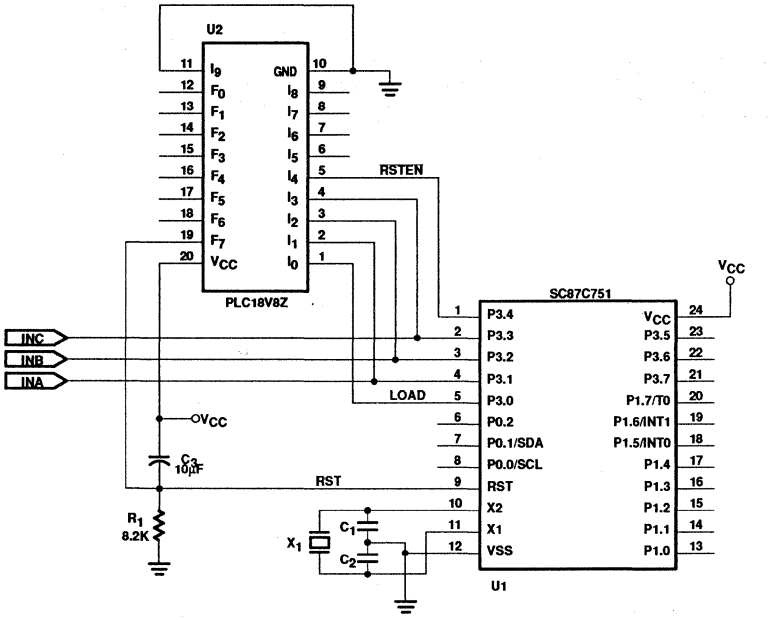

U2
PLC18V8Z
GND
RSTEN
LOAD
RST
INC
INB
INA
VCC
SC87C751
P3.4
P3.3
P3.2
P3.1
P3.0
P0.2
P0.1/SDA
P0.0/SCL
RST
X2
X1
VSS
P1.7/T0
P1.6/INT1
P1.5/INT0
P1.4
P1.3
P1.2
P1.1
P1.0
U1
C3
10µF
R1
8.2K
X1
C1
C2

Figure 3.

```
************************************************************
*       PLC18V8Z 20-Pin DIP Package Pin Layout             *
* Date: 10/03/93                          Time: 15:27:00 *
************************************************************

                    _________   _________
                   |         \_____/         |
                   |                         |
          LOAD [ 1|I0/CLK                VCC|20]
           INA [ 2|I1                      F7|19] RST
           INB [ 3|I2                      F6|18] OUTC
           INC [ 4|I3                      F5|17] OUTB
         RSTEN [ 5|I4                      F4|16] OUTA
               [ 6|I5                      F3|15] COMP
               [ 7|I6                      F2|14] NDC
               [ 8|I7                      F1|13] NDB
               [ 9|I8                      F0|12] NDA
               [10|GND                I9/OE_|11] OEN
                   |                         |
                   |_________________________|
```

```
@PINLIST

LOAD  i;INA   i;INB   i;
INC   i;RSTEN i;OEN   i;

OUTA  o;OUTB  o;OUTC  o;
COMP  o;NDA   o;NDB   o;
NDC   o;RST   o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

da.d   = ina;   "flip-flop equation"
da.clk = load;
nda = /da;      "invert buffer to pin equa-
tion"
nda.oe = /oen;

db.d   = inb;
db.clk = load;
ndb = /db;
ndb.oe = /oen;

dc.d   = inc;
dc.clk = load;
ndc = /dc;
ndc.oe = /oen;

outa = da*ina + /da*/ina;  "comparator"
outb = db*inb + /db*/inb;
outc = dc*inc + /dc*/inc;
comp = outa * outb * outc;

rst =  1;                  "3-state reset
control"
rst.oe = /rsten * /comp;   "enable when rsten
is low and no compare"

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 4.

Motor controller AN0303

PLC415 APPLICATIONS

This example places 2 independent stepper motor controllers in one PLC415. Each individually clocked controller includes a direction input as well as full and half step control. Individual set inputs force the internal state and output registers to state #1. Pin 19 is fused as a power-down input and may be used to reduce current consumption while the motors are stationary.

Suppose a stepper motor needs the sequence of data shown in Table 1 Clockwise rotation is performed by applying outputs associated with steps 1 through 8, while counter-clockwise rotation is achieved by applying outputs corresponding to 8 through 1. Each state or step in this table is actually one half step to the motor. A full step skips one state.

SNAP listings for this example are shown in Figures 3 and 4.

Table 1. Half Step Sequence

STATE	W1D	W1C	W1B	W1A
STEP 1	0	1	0	1
STEP 2	0	0	0	1
STEP 3	1	0	0	1
STEP 4	1	0	0	0
STEP 5	1	0	1	0
STEP 6	0	0	1	0
STEP 7	0	1	1	0
STEP 8	0	1	0	0

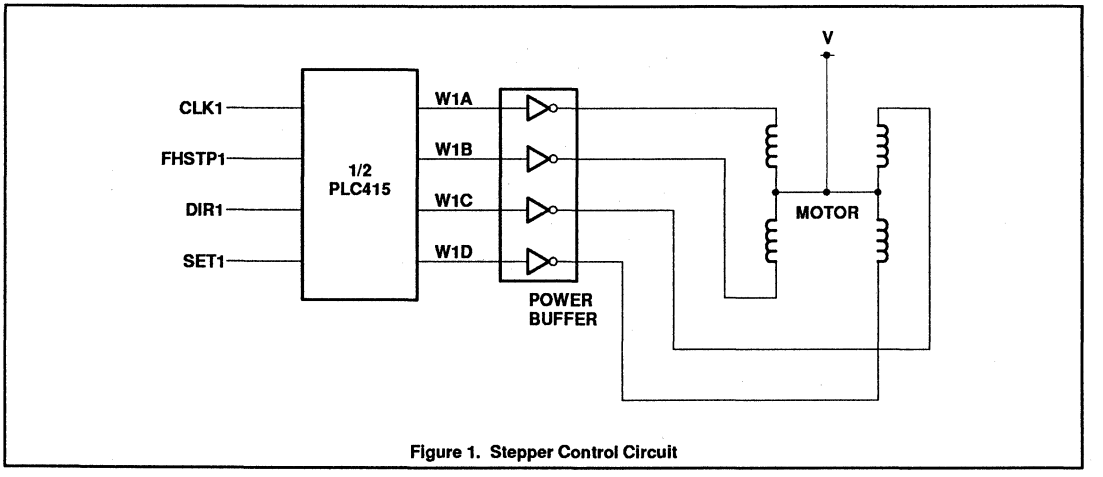
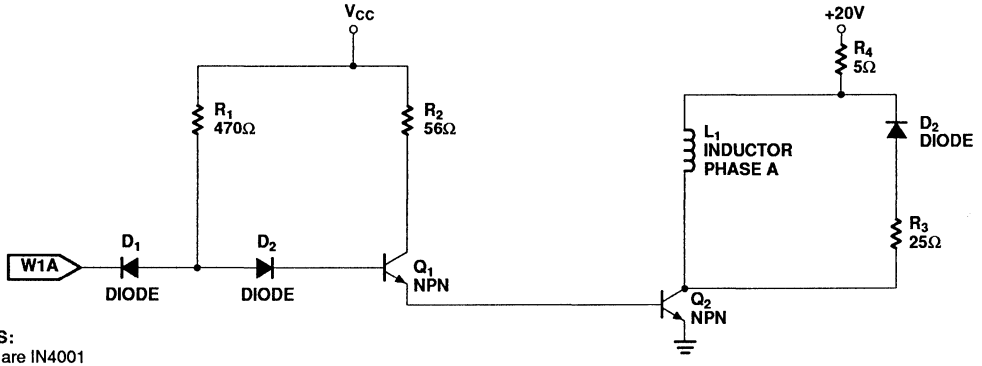


Figure 1. Stepper Control Circuit



NOTES:
D_1, D_2 are IN4001
Q_1 is a 2N2222
Q_2 is a 2N3055
D_3 is a BY127

Figure 2. Stepper Control Power Buffer
1 OF 4

```
************************************************************
*       PLC415 28-Pin PLCC Package Pin Layout              *
* Date: 10/03/93                              Time: 15:28:48 *
************************************************************

                    C D S C   S
                    L I E L   E
                    K R T K   T
                    2 1 1 1   2
                   +-+-+-+-+-+-+-+
                   | | | | |2|2|2|
                   |4|3|2|1|8|7|6|
                 +---------------------+
                 |    I I I C V I I    |
                 |    5 6 7 L C 8 9    |
                 |    /     K C        |
                 |    C                |
                 |    L                |
                 |    K                |
                 |                     |
     DIR2 [ 5|I4                  I10|25]
   FHSTP1 [ 6|I3                  I11|24]
   FHSTP2 [ 7|I2                  I12|23]
          [ 8|I1                  I13|22]
          [ 9|I0                  I14|21]
      W2A [10|F7                  I15|20]
      W2B [11|F6             IN/O/P/I|19]
                 |                     |
                 |       G             |
                 |    F F N F F F F    |
                 |    5 4 D 3 2 1 0    |
                 +---------------------+
                    |1|1|1|1|1|1|1|1|
                    |2|3|4|5|6|7|8|
                    +-+-+-+-+-+-+-+
                    W W   W W W W
                    2 2   1 1 1 1
                    C D   A B C D
```

Figure 3. SNAP Listing: Pin Diagram

```
@PINLIST
clk1   i;
clk2   i;
fhstp1 i;
dir1   i;
set1   i;
set2   i;
dir2   i;
fhstp2 i;

W1D o;
W1C o;
W1B o;
W1A o;
W2A o;
W2B o;
W2C o;
W2D o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
w1a.clk = clk1; "clock & set/rst for"
w1a.set = set1; "output flip-flops"
w1b.clk = clk1;
w1b.rst = set1;
w1c.clk = clk1;
w1c.set = set1;
w1d.clk = clk1;
w1d.rst = set1;

p1a.clk = clk1; "clock & set/rst for"
p1a.set = set1; "internal flip-flops"
p1b.clk = clk1;
p1b.rst = set1;
p1c.clk = clk1;
p1c.set = set1;
p1d.clk = clk1;
p1d.rst = set1;

w2a.clk = clk2;   "output flip-flops"
w2a.set = set2;
w2b.clk = clk2;
w2b.rst = set2;
w2c.clk = clk2;
w2c.set = set2;
w2d.clk = clk2;
w2d.rst = set2;

p2a.clk = clk2;    "internal flip-flops"
p2a.set = set2;
p2b.clk = clk2;
p2b.rst = set2;
p2c.clk = clk2;
p2c.set = set2;
p2d.clk = clk2;
p2d.rst = set2;

@INPUT VECTORS
@OUTPUT VECTORS
[w1d,w1c,w1b,w1a]
step1 = 0101b;
step2 = 0001b;
step3 = 1001b;
step4 = 1000b;
step5 = 1010b;
step6 = 0010b;
step7 = 0110b;
step8 = 0100b;
```

Figure 4. SNAP Listing: .EQN File (1 of 3)

```
[w2d,w2c,w2b,w2a]
step1a= 0101b;
step2a= 0001b;
step3a= 1001b;
step4a= 1000b;
step5a= 1010b;
step6a= 0010b;
step7a= 0110b;
step8a= 0100b;

@STATE VECTORS
[p1d,p1c,p1b,p1a]
stp1 = 0101b;    "state machine # 1"
stp2 = 0001b;
stp3 = 1001b;
stp4 = 1000b;
stp5 = 1010b;
stp6 = 0010b;
stp7 = 0110b;
stp8 = 0100b;

[p2d,p2c,p2b,p2a]
stp1a= 0101b;    "state machine # 2"
stp2a= 0001b;
stp3a= 1001b;
stp4a= 1000b;
stp5a= 1010b;
stp6a= 0010b;
stp7a= 0110b;
stp8a= 0100b;

@TRANSITIONS

   "motor controller #1"
while [stp1]
if [ fhstp1* dir1] then [stp3] with [step3]   "full step forward"
if [/fhstp1* dir1] then [stp2] with [step2]   "half step forward"
if [ fhstp1*/dir1] then [stp7] with [step7]   "full step backward"
if [/fhstp1*/dir1] then [stp8] with [step8]   "half step backward"

while [stp2]
if [ fhstp1* dir1] then [stp4] with [step4]
if [/fhstp1* dir1] then [stp3] with [step3]
if [ fhstp1*/dir1] then [stp8] with [step8]
if [/fhstp1*/dir1] then [stp1] with [step1]

while [stp3]
if [ fhstp1* dir1] then [stp5] with [step5]
if [/fhstp1* dir1] then [stp4] with [step4]
if [ fhstp1*/dir1] then [stp1] with [step1]
if [/fhstp1*/dir1] then [stp2] with [step2]

while [stp4]
if [ fhstp1* dir1] then [stp6] with [step6]
if [/fhstp1* dir1] then [stp5] with [step5]
if [ fhstp1*/dir1] then [stp2] with [step2]
if [/fhstp1*/dir1] then [stp3] with [step3]

while [stp5]
if [ fhstp1* dir1] then [stp7] with [step7]
if [/fhstp1* dir1] then [stp6] with [step6]
if [ fhstp1*/dir1] then [stp3] with [step3]
if [/fhstp1*/dir1] then [stp4] with [step4]

while [stp6]
if [ fhstp1* dir1] then [stp8] with [step8]
if [/fhstp1* dir1] then [stp7] with [step7]
if [ fhstp1*/dir1] then [stp4] with [step4]
if [/fhstp1*/dir1] then [stp5] with [step5]
```

Figure 4. SNAP Listing: .EQN File (2 of 3)

```
while [stp7]
if [ fhstp1* dir1] then [stp1] with [step1]
if [/fhstp1* dir1] then [stp8] with [step8]
if [ fhstp1*/dir1] then [stp5] with [step5]
if [/fhstp1*/dir1] then [stp6] with [step6]

while [stp8]
if [ fhstp1* dir1] then [stp2] with [step2]
if [/fhstp1* dir1] then [stp1] with [step1]
if [ fhstp1*/dir1] then [stp6] with [step6]
if [/fhstp1*/dir1] then [stp7] with [step7]

   "motor controller #2"
while [stp1a]
if [ fhstp2* dir2] then [stp3a] with [step3a]   "full step forward"
if [/fhstp2* dir2] then [stp2a] with [step2a]   "half step forward"
if [ fhstp2*/dir2] then [stp7a] with [step7a]   "full step backward"
if [/fhstp2*/dir2] then [stp8a] with [step8a]   "half step backward"

while [stp2a]
if [ fhstp2* dir2] then [stp4a] with [step4a]
if [/fhstp2* dir2] then [stp3a] with [step3a]
if [ fhstp2*/dir2] then [stp8a] with [step8a]
if [/fhstp2*/dir2] then [stp1a] with [step1a]

while [stp3a]
if [ fhstp2* dir2] then [stp5a] with [step5a]
if [/fhstp2* dir2] then [stp4a] with [step4a]
if [ fhstp2*/dir2] then [stp1a] with [step1a]
if [/fhstp2*/dir2] then [stp2a] with [step2a]

while [stp4a]
if [ fhstp2* dir2] then [stp6a] with [step6a]
if [/fhstp2* dir2] then [stp5a] with [step5a]
if [ fhstp2*/dir2] then [stp2a] with [step2a]
if [/fhstp2*/dir2] then [stp3a] with [step3a]

while [stp5a]
if [ fhstp2* dir2] then [stp7a] with [step7a]
if [/fhstp2* dir2] then [stp6a] with [step6a]
if [ fhstp2*/dir2] then [stp3a] with [step3a]
if [/fhstp2*/dir2] then [stp4a] with [step4a]

while [stp6a]
if [ fhstp2* dir2] then [stp8a] with [step8a]
if [/fhstp2* dir2] then [stp7a] with [step7a]
if [ fhstp2*/dir2] then [stp4a] with [step4a]
if [/fhstp2*/dir2] then [stp5a] with [step5a]

while [stp7a]
if [ fhstp2* dir2] then [stp1a] with [step1a]
if [/fhstp2* dir2] then [stp8a] with [step8a]
if [ fhstp2*/dir2] then [stp5a] with [step5a]
if [/fhstp2*/dir2] then [stp6a] with [step6a]

while [stp8a]
if [ fhstp2* dir2] then [stp2a] with [step2a]
if [/fhstp2* dir2] then [stp1a] with [step1a]
if [ fhstp2*/dir2] then [stp6a] with [step6a]
if [/fhstp2*/dir2] then [stp7a] with [step7a]
```

Figure 4. SNAP Listing: .EQN File (3 of 3)

DMA controller | AN0304

PLC42VA12 DMA APPLICATIONS

The PLC42VA12 contains 10 flip-flops that may flexibly be configured to build counters, shifters or any customized state machine required. With today's 32-bit microprocessors, there is a need for user-designed, system-specific DMA controllers that can generate addresses or count nibbles, bytes, half-words or words. Applications for these controllers include I/O concentration and cache subsystem updating. Typically, these devices can be preset or cleared and count (up) by 1, 2, or 4 depending on the chosen circumstances. A solution for the problem is presented in this section to illustrate solving the problem with Philips Semiconductors SNAP design software. The SNAP files are presented in Figure 1.

```
@PINLIST
CLK  I ;MODE0   I ;MODE1  I;RST I;LOAD I;CO  O;TOE  I;
OUTA O;OUTB  O;OUTC O;OUTD B;OUTE B;OUTF B;OUTG  G;OUTH B;OUTI B;OUTJ B;

@Logic Equations
               " model  mode0    function
                   0      0     count by 1
                   0      1     count by 2
                   1      0     count by 4
                   1      1       illegal
               "
QUOTA.J = 1*/load*/mode0*/model1            "load disables P-terms"
          + mode0*/load                     "force 0 for count by 2"
          + model*/load          + XOUTA;   "or count by 4"
QUOTA.K = 1*/load*/mode0*/model1  + YOUTA;   "XOUTA,YOUTA are outputs of "
                                            "tristate inputs "
QOUTB.J = outa*/load*/mode0*/model1
          + 1*/model1*mode0*/load
          + model*/mode0*/load    + XOUTB;  "force 0 count by four"
QOUTB.K = outa*/load*/mode0*/model1
          + model*/mode0*/load    + YOUTB;

DOUTC = outa*outb/load*/mode0*/model1
          + outb*/model1*mode0*/load;
QOUTC.J = DOUTC + XOUTC;
QOUTC.K = DOUTC + YOUTC;

DOUTD = outa*outb*outc*/load*/mode0*model1          "count by 1"
          + outb*outc*/model1*mode0*/load           "count by 2"
          + outc*model1*/mode0*/load;               "count by 4"
QOUTD.J = DOUTD + XOUTD;
QOUTD.K = DOUTD + YOUTD;

DOUTE = outa*outb*outc*outd*/load*/mode0*/model1
          + outb*outc*outd*/model1*mode0*/load
          + outc*outd*model1*/mode0*/load;
QOUTE.J = DOUTE + XOUTE;
QOUTE.K = DOUTE + YOUTE;

DOUTF = outa*outb*outc*outd*oute*/load*/mode0*model1
          + outb*outc*outd*oute*/model1*mode0*/load
          + outc*outd*oute*model1*/mode0*/load;
QOUTF.J = DOUTF + XOUTF;
QOUTF.K = DOUTF + YOUTF;

DOUTG = outa*outb*outc*outd*oute*outf*/load*/mode0*/model1
          + outb*outc*outd*oute*outf*/model1*mode0*/load
          + outc*outd*oute*outf*model1*/mode0*/load;
QOUTG.J = DOUTG + XOUTG;
QOUTG.K = DOUTG + YOUTG;

DOUTH = outa*outb*outc*outd*oute*outf*outg*/load*/mode0*/model1
          + outb*outc*outd*oute*outf*outg*/model1*mode0*/load;
          + outc*outd*oute*outf*outg*model1*/mode0*/load;
QOUTH.J = DOUTH + XOUTH;
QOUTH.K = DOUTH + YOUTH;
```

Figure 1. SNAP Files (1 of 3)

```
DOUTI = outa*outb*outc*outd*oute*outf*outg*outh*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*outg*outh*/mode1*mode0*/load
          + outc*outd*oute*outf*outg*outh*mode1*/mode0*/load;
QOUTI.J = DOUTI + XOUTI;
QOUTI.K = DOUTI + YOUTI;

DOUTJ = outa*outb*outc*outd*oute*outf*outg*outh*outi*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*outg*outh*outi*/mode1*mode0*/load
          + outc*coutd*oute*outf*outg*outh*outi*mode1*/mode0*/load;
QOUTJ.J = DOUTJ + XOUTJ;
QOUTJ.K = DOUTJ + YOUTJ;

CO = outa*outb*outc*outd*oute*outf*outg*outh*outi*outj*/load*/mode0*/mode1
         + outb*outc*outd*oute*outf*outg*outh*outi*outj*/mode1*mode0*/load
         + outc*outd*oute*outf*outg*outh*outi*outj*mode1*/mode0*/load;

" Reset for all flip-flops "
QOUTA.RST = RST;
QOUTB.RST = RST;
QOUTC.RST = RST;
QOUTD.RST = RST;
QOUTE.RST = RST;
QOUTF.RST = RST;
QOUTG.RST = RST;
QOUTH.RST = RST;
QOUTI.RST = RST;
QOUTJ.RST = RST;

" Flip-flops are followed by tristate outputs which drive the pin "
OUTA = /QOUTA;
OUTB = /QOUTB;
OUTC = /QOUTC;
OUTD = /QOUTD;
OUTE = /QOUTE;
OUTF = /QOUTF;
OUTG = /QOUTG;
OUTH = /QOUTH;
OUTI = /QOUTI;
OUTJ = /QOUTJ;
OUTA.OE = TOE;
OUTB.OE = TOE;
OUTC.OE = TOE;
OUTD.OE = TOE;
OUTE.OE = TOE;
OUTF.OE = TOE;
OUTG.OE = TOE;
OUTH.OE = TOE;
OUTI.OE = TOE;
OUTJ.OE = TOE;
```

Figure 1. SNAP Files (2 of 3)

```
"Pins are fed back to flip-flops using tristate inputs (FF load)"
XOUTA = /OUTA;       "feed.back to J is inverted"
YOUTA =  OUTA;       "feed-back to K IS NOT inverted"
XOUTB = /OUTB;
YOUTB =  OUTB;
XOUTC = /OUTC;
YOUTC =  OUTC;
XOUTD = /OUTD;
YOUTD =  OUTD;
XOUTE = /OUTE;
YOUTE =  OUTE;
XOUTF = /OUTF;
YOUTF =  OUTF;
XOUTG = /OUTG;
YOUTG =  OUTG;
XOUTH = /OUTH;
YOUTH =  OUTH;
XOUTI = /OUTI;
YOUTI =  OUTI;
XOUTJ = /OUTJ;
YOUTJ =  OUTJ;
XOUTA.LD = LOAD;
XOUTB.LD = LOAD;
XOUTC.LD = LOAD;
XOUTD.LD = LOAD;
XOUTE.LD = LOAD;
XOUTF.LD = LOAD;
XOUTG.LD = LOAD;
XOUTH.LD = LOAD;
XOUTI.LD = LOAD;
XOUTJ.LD = LOAD;
YOUTA.LD = LOAD;
YOUTB.LD = LOAD;
YOUTC.LD = LOAD;
YOUTD.LD = LOAD;
YOUTE.LD = LOAD;
YOUTF.LD = LOAD;
YOUTG.LD = LOAD;
YOUTH.LD = LOAD;
YOUTI.LD = LOAD;
YOUTJ.LD = LOAD;
```

Figure 1. SNAP Files (3 of 3)

I²C bus expander AN036

INTRODUCTION

This application note describes two PLD designs made with the PLC42VA12. Both designs are controller functions for an I²C-bus n-bit I/O expander. The first design is a controller function for a n-bit I²C-bus Input Expansion (I²C-bus Slave Transmitter function) and the second one for a n-bit I²C-bus Output Expansion (I²C-bus Slave Receiver function).

The I²C-bus is a 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). The designs provide remote input or output expansion for our Philips micro controller families via the two-line serial bidirectional I²C-Bus. The I²C-Bus slave address of the designs is equal to the address of the PCF8574 (remote 8-bit I/O expander). The I²C-Bus has been specified for 100kHz, but the PLC42VA12 designs can go up to 1MHz. This makes the designs suitable as test vehicle for the new fast I²C-Bus standard of 400 kHz.

The PLC42VA12 is the most powerfull PLD device in a DIL-package of Philips Semiconductors. The designs use almost all resources and most of the features of the PLC42VA12 e.g. combination of synchronous and asynchronous logic, 3-State outputs used as open collector outputs and a combination of output flip-flops and state flip-flops.

The design has been verified on a bread-board. This board contains the two PLC42VA12 controller designs, four 74HC165 devices, four 74HC595 devices and all the circuitry necessary to read 32 DIP-switches and to control 32 LED's.

This application note gives first a general description of the designs. Then it describes the characteristics of the I²C-Bus and some basic functions (tricks) used in both designs. You will find a detailed description of the designs and the I²C-Bus protocols of the controller functions in the sections entitled, I²C-Bus Slave Transmitter Function and I²C-Bus Slave Receiver Function. The appendix, gives all the used design files in the SNAP syntax. SNAP is the Philips Semiconductors PLD design software package. You will find the equation entry files EQN, the simulation control files SCL and the pinning files PIN. The last sheet gives the schematic diagram of the bread-board.

GENERAL DESCRIPTION OF THE DESIGNS

The two designs described in this report are both controller functions for an I²C-Bus n-bit I/O-expander. The designs were made on a customer request to have a solution for his problem to address via the I²C-Bus more than 200 bits of inputs and outputs. With the existing I²C-Bus devices the maximum number of input and/or output bits is 16 x 8 = 128 (8 x PCF8574 + 8 x PCF8574A). The designs work fully according to the I²C-Bus specification at 100kHz.

When an I²C-Bus master device (e.g. a micro controller) has to read data from or write data to the remote I/O-expander devices PCF8574 and PCF8574A, it first sends the I²C-Bus slave address of the device and then reads or writes only one byte. The two controller functions, the Slave-Receiver and the Slave Transmitter, don t have this problem. For these designs, the master sends the slave address only once, and then reads or writes one or multiple data-bytes. The master device, decides the number of bytes. The slave addresses used for the designs are identical to the slave addresses of the PCF8574 and PCF8574A devices.

The I²C-bus has been specified for a 100 kHz clock (SCL). With the internal maximum system clock of 8 MHz, the two PLC42VA12 designs can go upto an I²C-bus clock of 1MHz. This makes the design suitable as test vehicle for the new fast I²C-bus standard of 400 kbits/s. The speed is the only additional specification point of this new I²C-bus specification that can be met. The other new specification points as Schmitt-trigger inputs and slope control of the falling edges of the SDA and SCL signals can not be met.

The PLC42VA12 has been chosen, because of its special hardware features. These features are not available in other PLD devices available in a 24-pin DIL package e.g. the PL22V10. Some of the used features are:

- Combination of synchronous and asynchronous logic.
- Combination of D-type flip-flops with JK-type flip-flops.
- Flip-flops used as state registers. The M-pins used as inputs and/or outputs.
- 3-State buffers used as open collector outputs. The 3-State control-input of an I/O Output buffer is used as logic input. The input of that buffer is connected to the ground.

The intention was, to put both the Slave Transmitter and the Slave Receiver controller in one device. Unfortunately, the resources of the PLC42VA12 are not sufficient to implement both designs in one device. As a combination of a Slave Transmitter and a Slave Receiver with a high number of inputs and outputs is seldom requested, this should not be a problem.

The first PLC42VA12 design is an I²C-bus Slave Transmitter controller. With one or multiple 74HC165 devices, it forms an I²C-bus n-bit Input Expander. At one side the controller fully controls the I²C-Bus Slave Transmitter function, and at the other side it generates the control signals for the 74HC165. The 74HC165 devices can be cascaded to increase the number of inputs. Chapter. I²C-Bus Slave Transmitter Function gives a detailed description of the design.

The second design is an I²C-bus Slave Receiver controller. With one or multiple 74HC595 devices it forms an I²C-bus n-bit Output Expander. At one side the controller fully controls the I²C-Bus Slave Receiver function, and at the other side it generates the control signals for the 74HC595. The 74HC595 devices can be cascaded to increase the number of outputs. The section entitled I²C-Bus Slave Receiver Function describes in detail the design.

For design verification purposes, a bread-board has been made. This board contains all the devices to build a Slave Transmitter with 32 inputs and a Slave Receiver with 32 outputs. The main devices of the board are:

- A PLC42VA12 with the Slave Transmitter controller function,
- 4 PC74HC165 devices,
- 4 octal DIP-switches,
- A PLC42VA12 containing the Slave Receiver controller function,
- 4 PC72HC595 devices,
- 32 LED's.

CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is a 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit Transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (Figure 1, Bit Transfer). The maximum clock frequency is 100 kHz.

Start and Stop Conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P). Figure 2, Definition of Start and Stop Conditions, gives the timing diagram.

System Configuration

A device generating a message is a transmitter, a device receiving a message is the receiver. The device that controls the message is the master and the devices which are controlled by the master are slaves. Figure 3, System Configuration, gives a block diagram of the system configuration.

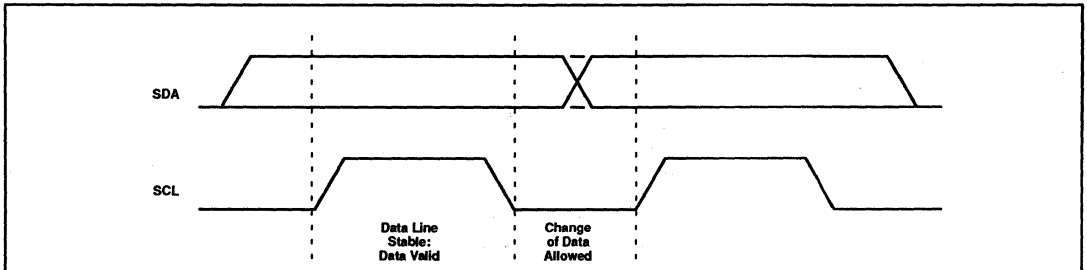


Figure 1. Bit Transfer

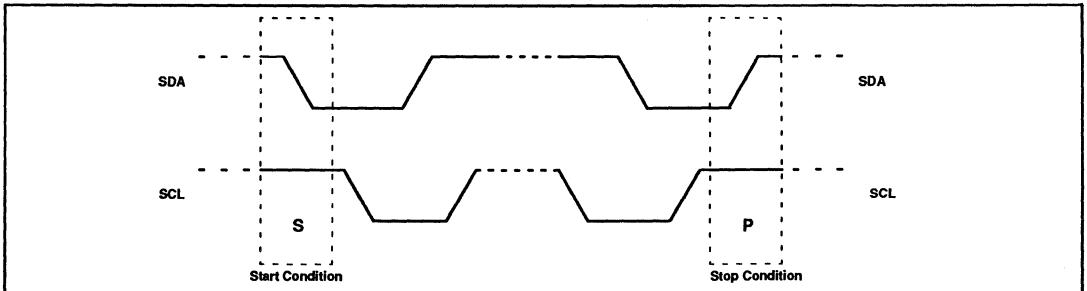


Figure 2. Definition of Start and Stop Conditions

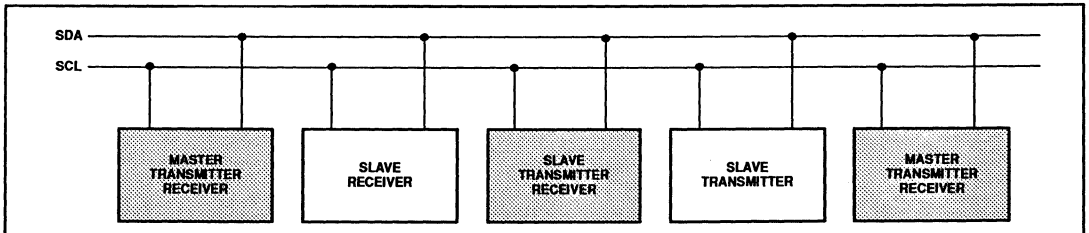


Figure 3. System Configuration

Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Setup and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition (see Figure 4, Acknowledgement on the I²C-bus).

Formats

Data transfers follow the format shown in Figure 5 Data formats of the I²C-bus. After the START condition, the master sends the slave address. This address is 7 bits long, the eighth bit is a data direction bit (R/WN). A zero indicates a transmission (WRITE) and a one indicates a request for data (READ). A master always terminates a data transfer by a STOP condition. However, if a master still wishes to communicate on the bus, it can generate an other START condition and address an other slave without first generating a STOP condition. Various combinations of read and write formats are then possible within such a transfer.

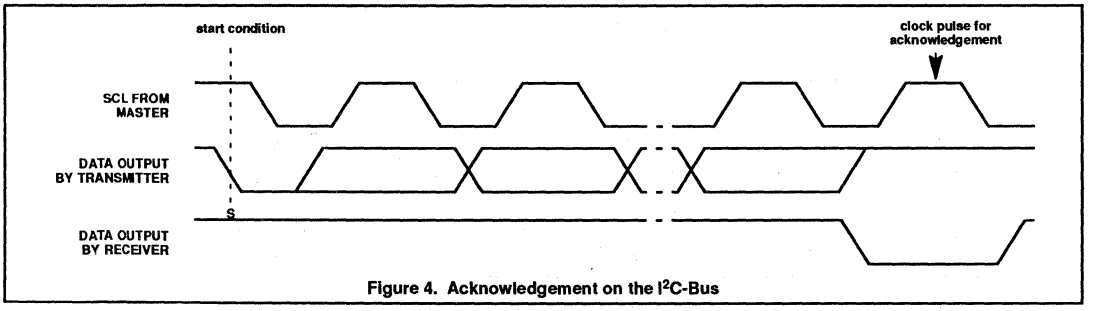
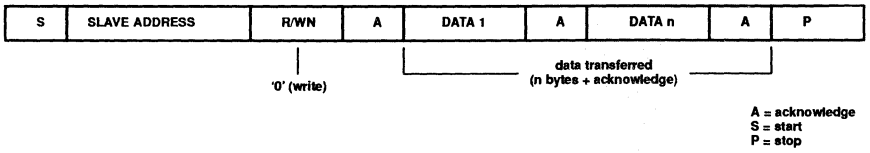


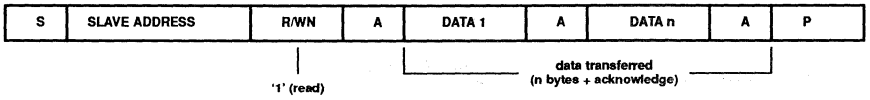
Figure 4. Acknowledgement on the I²C-Bus

Possible data transfer formats are:

a. Master transmitter to slave receiver. Direction is not changed.

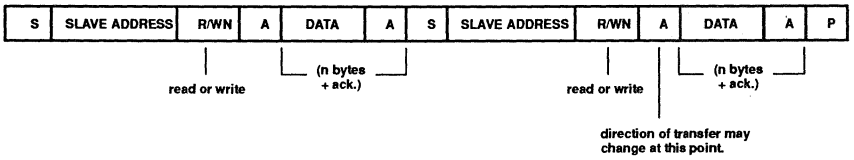


b. Master reads slave immediately after first byte.



At the moment of the first acknowledge, the master transmitter becomes a master receiver and the slave receiver becomes a slave transmitter. This acknowledge is still generated by the slave.

The STOP condition is generated by the master.



During a change of direction within a transfer, the START condition and the slave address are both repeated, but the R/W bit reversed. Start, stop, slave addresses and R/W bits are generated by the master.

Figure 5. Data formats of the I²C-Bus

COMMON BASIC FUNCTIONS

This section gives a number of common basic functions used in the designs. The report gives for each function the basic diagram, the SNAP description and the timing diagram if applicable. The following basic circuits are described:

- Oscillator
- SCL Edge detection
- Start/Stop detection

Oscillator

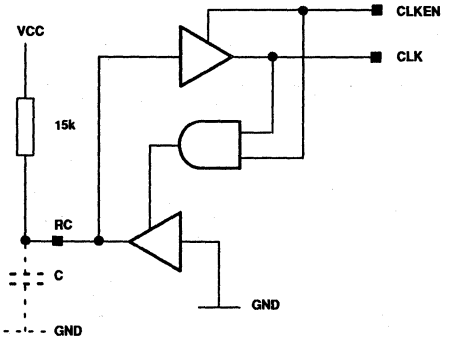
The design has two clock options, the internal oscillator and an external clock. For both options, the clock input CLK is used. A HIGH CLKEN input selects the internal oscillator and a LOW input the external clock. Without capacitor, we get the maximum frequency of the internal clock of 8 MHz. This frequency can be lowered by using a small capacitor C. Figure 6 Oscillator shows the diagram and the EQN file description of the oscillator.

SCL Edge Detection

The frequency of the system clock is much higher than the I²C-bus clock (SCL). This means, that most of the time the state machine is waiting for the edges of the SCL clock. This section describes the circuit that detects the HIGH and the LOW going edge of the SCL clock. The state machines synchronizes on the output pulses SCLH and SCLL.The detection network uses only two flip flops and two AND gates. Figure 7 SCL Timing Diagram SCL Edge Detection. gives the timing diagram of the edge detector and Figure 8 SCL Edge Detection. the diagram and the description of the EQN file.

Start/Stop Detection

A HIGH-to-LOW transition of the data line, while the clock is HIGH has been defined as the start condition (S) of the I²C-bus. A LOW-to-HIGH transition of the data line while the clock is HIGH has been defined as the stop condition (P). The easiest way to detect this start and stop condition is using asynchronous logic. The PLC42VA12 is very suited for this kind of solutions. Figure 2, Definition of Start and Stop Conditions, gives the timing diagram of these conditions.




```
@PINLIST
RC              B;
CLK             B;
CLKEN           I;

@LOGIC EQUATIONS
CLK.OW          =CLKEN
RC.OE           =CLK*CLKEN;
RC              =GND;
CLK             =RC;
```

Figure 6. Oscillator

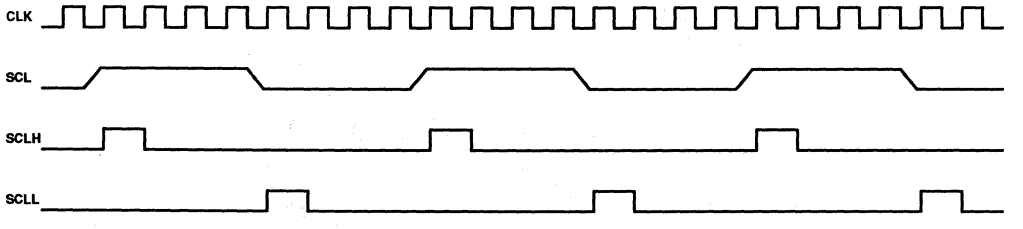
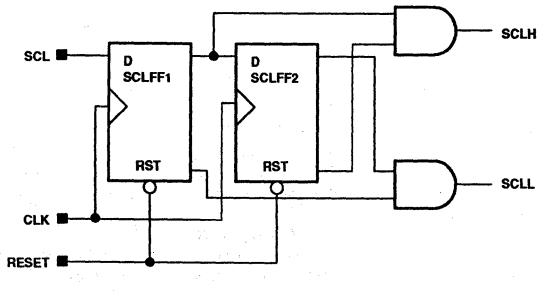
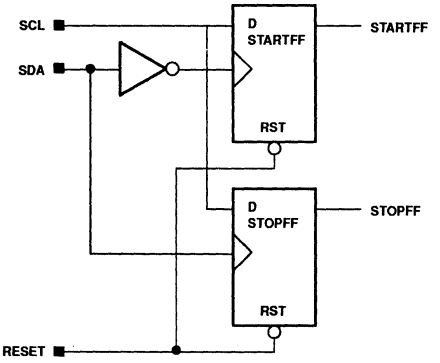


Figure 7. SCL Timing Diagram SCL Edge Detection




```
@PINLIST
SCL             I;
CLK             B;
RESET           I;

@LOGIC EQUATIONS
SCLFF1.D        =SCL;
SCLFF1.RST      =RESET;
SCLFF2.D        =SCLFF1;
SCLFF2.RST      =RESET;
SCLH            =SCLFF1*/SCLFF2;
SCLL            =/SCLFF1*SCLFF2;
```

Figure 8. SCL Edge Detection




```
@PINLIST
SCL             I;
SDA             B;
RESET           I;

@LOGIC EQUATIONS
STARTFF.CLK     =/SDA;
STARTFF.D       =SCL;
STARTFF.RST     =RESET;
STOPFF.CLK      =SDA;
STOPFF.D        =SCL;
STOPFF.RES      =RESET;
```

Figure 9. Start/Stop Detection

I²C-BUS SLAVE TRANSMITTER FUNCTION

The Slave Transmitter design provides remote input expansion for our Philips microcontroller families via the two-line serial bidirectional I²C-Bus. The I²C-Bus slave address is equal to the address of the PCF8574 (remote 8-bit I/O expander). The design will only acknowledge the read-mode.

The design handles the full slave read mode of the I²C-Bus and will generate the control signals for the 74HC165, an 8-bit parallel-in / serial-out shiftregister. This device is used to read the parallel input data and convert this into serial data. This data is written to the I²C-Bus. The total number of 74HC165 devices is almost unlimited.

With the three address selection inputs, the slave transmitter can be combined with multiple PCF8574 devices.

The design has a build-in clock oscillator. The section entitled Oscillator describes this circuit. If an external clock is required, the internal oscillator can be inhibited with the CLKEN-input.

Figure 17, Slave Transmitter EQN File, Figure 18, Slave Transmitter SCL File, and Figure 19, Slave Transmitter PIN File, give the design files. The sections SDA Control Slave Transmitter, I²C-Bus protocol Slave Transmitter and Interface with 74HC165 give a detailed description of parts of the design.

SDA Control Slave Transmitter

The SDA data line of the I²C-bus is a bidirectional line with a passive pull-up. This asks for a bidirectional open collector I/O line. As the PLC42VA12 has only 3-State I/Os, we need one of the advantages of the PLC42VA12 to get an open collector output. IN a PLC42VA12 each OE-input can be used as a logic input. With a LOW level (ground) at the input, the output has a LOW level when OE is true and is floating when OE is false. These are the characteristics of an open collector output.

Only one AND-gate controls the OE input of a bidirectional I/O. The design asks for multi-level logic. Figure 10 SDA Control Slave Transmitter shows how this input can be controlled by multi-level logic. The inputs ACKNOW, DATIN and SDAIN are auxiliary outputs of the PLC42VA12, used as inputs of the SDA control.

I²C-Bus protocol Slave Transmitter

The section entitled Formats describes the general data formats of the I²C-Bus protocol. Figure 11 I²C-Bus Protocol Slave Transmitter gives the protocol for this design. After a start condition, the master sends the slave address of the device. This 7 bits address consists of a fixed part and a programmable part. The first four bits are fixed (0100) and the three least significant bits are programmable. Three hardware address pins determine the final slave address. Up to 8 devices (or PCF8574) may be addressed by the master. After the slave address and a HIGH R/WN bit, the slave generates an acknowledge. At the next LOW SCL, the slave may start sending the first data byte. This byte will be acknowledged by the master. Also the next bytes will be acknowledged by the master. As after the acknowledge pulse the slave controls the SDA-line, the master can not generate a stop condition. The only way for a master to terminate a transmission, is not to acknowledge the last byte n. Then, the slave transmitter will release the SDA-line and the master can generate a stop condition.

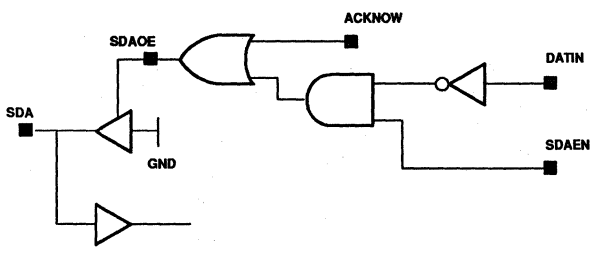
Interface with 74HC165

The 74HC/HCT165 is an 8-bit parallel load or serial-in shift register with complementary serial outputs (Q7 and Q7N) available from the last stage. When the parallel load (PLN) input is LOW, parallel data from the D0 to D7 inputs are loaded into the register asynchronously. When PLN is HIGH, data enters the register serially at the DS input and shifts one place to the right with each positive-going clock transition. This feature allows parallel-to-serial converter expansion by tying the Q7 output to the DS input of the succeeding stage.

The CLOUT output of the PLC42VA12 controls the clock of the 74HC165 and the PLOADN output controls the PLN input. The Q7 output of the 74HC165 is the data input DATIN of the PLC42VA12.

With this setup, the most significant bit of the data is the first bit that will be sent from the slave to the master. Figure 11 I²C-Bus Protocol Slave Transmitter gives the timing diagram of this interface.

When the slave address and the read bit have been detected, the controller generates the parallel load pulse PLOADN. After sending the firs bit (most significant bit of the transmission) it generates the first shift pulse CLOCKOUT. At the end of the first byte, the master generates an acknowledge. The second byte starts with a shift pulse CLOCKOUT. At each next LOW SCL level, this pulse is repeated. If at the end of the byte the master sends an acknowledge, then the next byte will be sent. A not acknowledge stops the procedure.




```
@PINLIST
SDA             B;
ACKNOW          I;
DATIN           I;
SDAEN           O;
SDAOE           O;

@LOGIC EQUATIONS
SDA             =GND;
SDA.OE          =SDAOE;
SDAOE           =ACKNOW+
                (/DATIN*SDAEN)
```

Figure 10. SDA Control Slave Transmitter

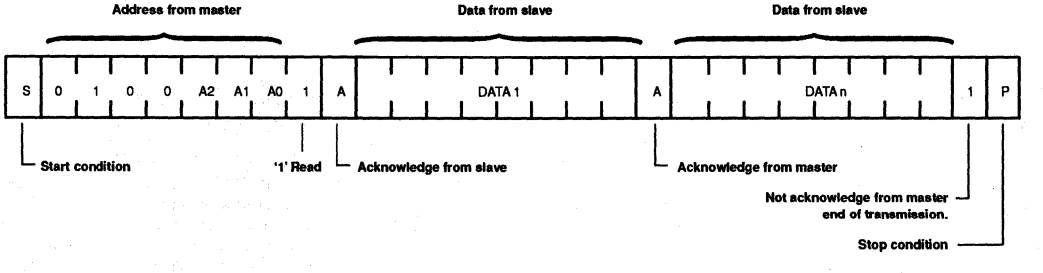


Figure 11. I²C-bus Protocol Slave Transmitter

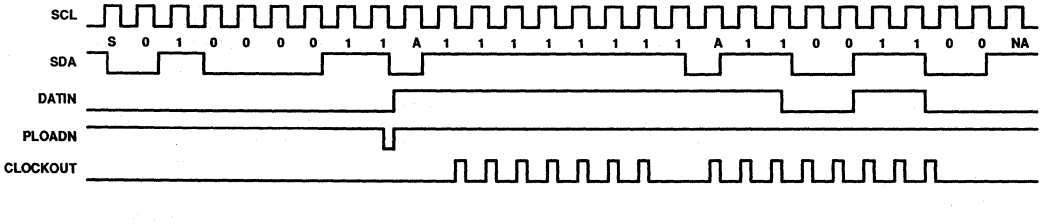


Figure 12. Interface with 74HC165

I²C-BUS SLAVE RECEIVER FUNCTION

The slave receiver design provides remote output expansion for our Philips micro controller families via the two-line serial bidirectional I²C-bus. The I²C-bus slave address is equal to the address of the PCF8574 (remote 8-bit I/O expander). The design will only acknowledge the write-mode.

The design handles the full slave write mode of the I²C-Bus and will generate the control signals for the 74HC595, an 8-bit serial-in / parallel-out shiftregister. This device is used to write the serial from the I²C-bus input to, and convert this into parallel data. The total number of 74HC595 devices is almost unlimited.

With the three address selection inputs, the slave receiver can be combined with multiple PCF8574 devices.

The design has a build-in clock oscillator. The section entitled Oscillator describes this circuit. If an external clock is required, the internal clock can be inhibited with the CLKEN-input. Figure 20, Slave Receiver EQN File, Figure 21, Slave Receiver SCL File, and Figure 22, Slave Receiver PIN File give the design files. The sections SDA Control Slave Receiver, Set-Reset Flip-Flop, I²C-Bus protocol Slave Receiver and Interface with 74HC595 give a detailed description of parts of the design.

SDA Control Slave Receiver

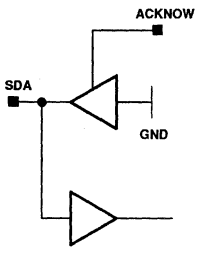
In the Slave Receiver protocol of the I²C-Bus, mainly the master controls the SDA-line. The Slave Receiver uses the SDA-line only for generating an acknowledge pulse. This is done after receiving its slave address with a write condition and all the following data bytes. Figure 13, SDA Control Slave Receiver gives the diagram and the description of the EQN-file.

Set-Reset Flip-Flop

The PLC42VA12 has 10 internal flip-flops. As the design needs an additional D-latch, this one has to be built out of gates. Figure 14, Set-Reset Flip-Flop Slave Receiver, gives the diagram and the equation file description of this function. In this example the signal STO7 is defined as an input, but in the final design this is a auxiliary output of the device.

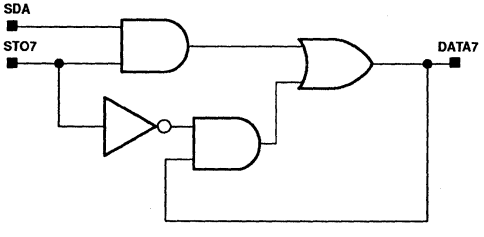
I²C-Bus protocol Slave Receiver

The section entitled Formats describes the general format of the I²C-Bus protocol. Figure 14, Set-Reset Flip-Flop Slave Receiver, gives the protocol for this design. After a start condition, the master sends the slave address of the device. This 7 bits address consists of a fixed part and a programmable part. The first four bits are fixed (0100) and the three least significant bits are programmable. Three hardware address pins determine the final slave address. Up to 8 devices (or PCF8574) may be addressed by the master. After the slave address and a LOW R/WN bit, the slave generates an acknowledge. At the next LOW SCL, the master starts sending the first data byte. This byte will be acknowledged by the slave. Also the next bytes will be acknowledged by the slave. The master terminates a transmission, by sending a stop condition or a restart condition.




```
@PINLIST
SDA             B;
ACKNOW          I;

@LOGIC EQUATIONS
SDA             =GND;
SDA.OE          =ACKNOW
```

Figure 13. SDA Control Slave Receiver




```
@PINLIST
SDA             B;
STO7            I;
DATA7           O;

@LOGIC EQUATIONS
DATA7 = ((STO7*SDA)+
        (DATA7*/STO7))
```

Figure 14. Set-Reset Flip-Flop Slave Receiver

Interface with 74HC595

The 74HC/HCT595 is an 8-stage serial shift register with a storage register and 3-State outputs. The shift register and storage register have separate clocks.

Data is shifted on the positive-going transitions of the SHCP input. The data in each register transfers to the storage register on a positive going transition of the STCP input. The shift register has a serial input (DS) and a serial standard output (Q7) for cascading. All 8 shift registers have an asynchronous reset (active LOW). The storage register has 8 parallel 3-State bus driver outputs. Data in the storage registers at the output whenever the output enable input (OEN) is LOW.

Four outputs of the PLC42VA12 control the inputs of the 74HC595. The RESOUT output the MRN input, CLKSTO the STCP input, CLKSHFT the SHCP and the DATOUT the DS input.

The I²C-Bus sends first the most significant bit of the transmission. Figure 16 I²C-Bus Interface with 74HC595 gives the timing diagram of this interface.

To set all outputs of the 74HC595 to a defined level, after power-on, the controller generates first a reset pulse at the RESOUT output and then a clock pulse CLKSTO for the storage register.

After a the slave address and the write bit (LOW) have been detected, during the next HIGH period of the SCL line there are three options. At the SDA line there can be the most significant bit of new data, the master generates a restart condition or the master generates a stop condition. This is also the case after each acknowledge.

This implies, that this first data bit must be stored. At the next LOW period of the SCL line we know whether we had data or restart/stop condition. If it was data, then we have to put this data at the DATOUT output and generate a clock pulse at the CLKSHFT output. At the next 7 HIGH periods of the SCL-line, data is valid and the controller generates a clock at the CLKSHFT output. At the end of the transmission, the master generates a stop condition or a restart. Then the stored data will be transferred to the storage register by a clock pulse at the CLKSTO output.

BREAD-BOARD I²C-BUS I/O EXPANDER

For design verification purposes, a bread-board has been designed. The board contains all the devices to build an I²C-Bus Slave Transmitter with 32 inputs and an I²C-Bus Slave Receiver with 32 outputs. The inputs can be set HIGH or LOW by 4 octal DIP-switches. The outputs are examined by 32 LED's. Figure 23, Schematic Diagram Bread-Board, gives the complete diagram of the bread-board.

The board has been designed for design verification only.

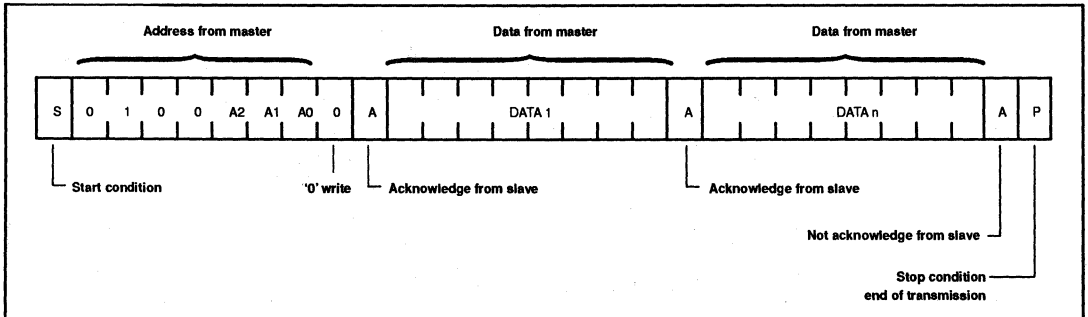


Figure 15. I²C-bus Protocol Slave Receiver

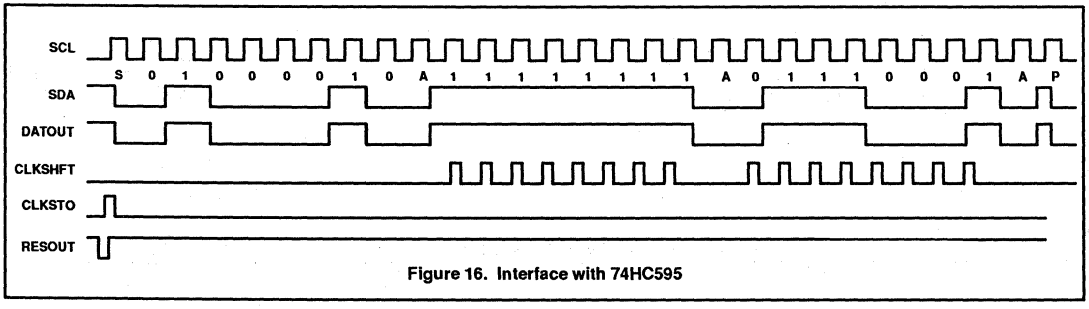


Figure 16. Interface with 74HC595

```
"
**************************************************
*                                                *
*               Equation Entry File              *
*                                                *
* Project      : IIC                             *
* Function     : IIC-bus Slave Transmitter       *
*                                                *
* File Name    : IICTRANS.EQN                    *
* Design file  : IICTRANS.SCL                    *
* Pin File     : IICTRANS.PIN                    *
*                                                *
* Date         : March 1993                      *
* Designer     : Aloys Schatorj                  *
* Company      : Philips Semiconductors          *
* Department   : PCALE                           *
* Place        : Eindhoven                       *
* Country      : The Netherlands                 *
*                                                *
**************************************************
"
@PINLIST
CLK        B;      "System clock"
RC         B;      "RC input internal clock"
CLKEN      I;      "Clock selection input"
SCL        I;      "IIC-BUS clock"
SDA        B;      "IIC-BUS data"
RESET      I;      "System reset"
ADD0       I;      "Address selection line"
ADD1       I;      "Address selection line"
ADD2       I;      "Address selection line"
DATIN      I;      "Data from input shift-register"
CLOCKOUT   O;      "Clock to input shift-register"
PLOADN     O;      "Parallel load to input shift-register"
SDAOE      O;      "Enable line SDA I/O"
SDAEN      O;      "Enable condition SDA caused by data"
ACKNOW     O;      "Acknowledge data"
STATEST    O;      "Reset start and stop FF"

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
CLK.OE       =  CLKEN    ;
RC.OE        =  CLK*CLKEN;
RC           =  GND      ;
CLK          =  RC       ;
STARTFF.CLK  = /SDA      ;
STARTFF.D    =  SCL      ;
STARTFF.RST  =  STATEST  ;
SCLFF1.D     =  SCL      ;
SCLFF1.RST   =  RESET    ;
SCLFF2.D     =  SCLFF1   ;
SCLFF2.RST   =  RESET    ;
SCLH         = SCLFF1*/SCLFF2 ;
SCLL         = /SCLFF1*SCLFF2 ;
Q0.RST       =  RESET    ;
Q1.RST       =  RESET    ;
Q2.RST       =  RESET    ;
Q3.RST       =  RESET    ;
Q4.RST       =  RESET    ;
Q5.RST       =  RESET    ;
SDA          =  GND      ;
SDA.OE       =  SDAOE    ;
PLOADN       = /PLOAD    ;
SDAOE        =  ACKNOW + (/DATIN*SDAEN)    ;
```

Figure 17. Equation Entry File Slave Transmitter (1 of 3)

```
@INPUT VECTORS

@OUTPUT VECTORS
          [ACKNOW, CLOCKOUT, PLOAD, SDAEN]
ACK      =   1        -         -        -   B;
CLKOUT   =   -        1         -        1   B;
SDAENA   =   -        -         -        1   B;
PLOUT    =   -        -         1        1   B;

@STATE VECTORS
         [Q5, Q4, Q3, Q2, Q1, Q0] JKFFR
INITL      =  00 H;
INIT       =  3F H;
WAIT       =  3E H;
WAIT1      =  3D H;
ADDBIT6    =  01 H;
ADDBIT5    =  02 H;
ADDBIT4    =  03 H;
ADDBIT3    =  04 H;
ADDBIT2    =  05 H;
ADDBIT1    =  06 H;
ADDBIT0    =  07 H;
RWBIT      =  08 H;
READMOD    =  09 H;
ACKBITR    =  0A H;
READ7L     =  0B H;
READ7C     =  10 H;
READ7      =  11 H;
READ6C     =  12 H;
READ6      =  13 H;
READ5C     =  14 H;
READ5      =  15 H;
READ4C     =  16 H;
READ4      =  17 H;
READ3C     =  18 H;
READ3      =  19 H;
READ2C     =  1A H;
READ2      =  1B H;
READ1C     =  1C H;
READ1      =  1D H;
READ0C     =  1E H;
READ0      =  1F H;
ACKPLS     =  20 H;
ACKPLSW    =  21 H;

@TRANSITIONS
WHILE    [INITL]
         IF     []              THEN   [INIT]
WHILE    [INIT]
         IF     []              THEN   [WAIT]
WHILE    [WAIT]                                    WITH   [STATEST]
         IF     [STARTFF]       THEN   [WAIT1]
WHILE    [WAIT1]                                   WITH   [STATEST]
         IF     []              THEN   [ADDBIT6]
WHILE    [ADDBIT6]
         IF     [SCLH*SDA]      THEN   [WAIT]
         IF     [SCLH*/SDA]     THEN   [ADDBIT5]
WHILE    [ADDBIT5]
         IF     [SCLH*/SDA]     THEN   [WAIT]
         IF     [SCLH*SDA]      THEN   [ADDBIT4]
WHILE    [ADDBIT4]
         IF     [SCLH*SDA]      THEN   [WAIT]
         IF     [SCLH*/SDA]     THEN   [ADDBIT3]
WHILE    [ADDBIT3]
         IF     [SCLH*SDA]      THEN   [WAIT]
         IF     [SCLH*/SDA]     THEN   [ADDBIT2]
WHILE    [ADDBIT2]
         IF     [SCLH*/((SDA*ADD2)+(/SDA*/ADD2))] THEN   [WAIT]
         IF     [SCLH*((SDA*ADD2)+(/SDA*/ADD2))]  THEN   [ADDBIT1]
```

Figure 17. Equation Entry File Slave Transmitter (2 of 3)

```
WHILE   [ADDBIT1]
        IF      [SCLH*/((SDA*ADD1)+(/SDA*/ADD1))] THEN   [WAIT]
        IF      [SCLH*((SDA*ADD1)+(/SDA*/ADD1))]  THEN   [ADDBIT0]
WHILE   [ADDBIT0]
        IF      [SCLH*/((SDA*ADD0)+(/SDA*/ADD0))] THEN   [WAIT]
        IF      [SCLH*((SDA*ADD0)+(/SDA*/ADD0))]  THEN   [RWBIT]
WHILE   [RWBIT]
        IF      [SCLH*SDA]    THEN   [READMOD]
        IF      [SCLH*/SDA]   THEN   [WAIT]
WHILE   [READMOD]
        IF      [SCLL]        THEN   [ACKBITR]
WHILE   [ACKBITR]                                    WITH   [ACK]
        IF      [SCLL]        THEN   [READ7L]
WHILE   [READ7L]                                     WITH   [PLOUT]
        IF      []            THEN   [READ7]
WHILE   [READ7C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ7]
WHILE   [READ7]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ6C]
WHILE   [READ6C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ6]
WHILE   [READ6]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ5C]
WHILE   [READ5C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ5]
WHILE   [READ5]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ4C]
WHILE   [READ4C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ4]
WHILE   [READ4]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ3C]
WHILE   [READ3C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ3]
WHILE   [READ3]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ2C]
WHILE   [READ2C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ2]
WHILE   [READ2]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ1C]
WHILE   [READ1C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ1]
WHILE   [READ1]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [READ0C]
WHILE   [READ0C]                                     WITH   [CLKOUT]
        IF      []            THEN   [READ0]
WHILE   [READ0]                                      WITH   [SDAENA]
        IF      [SCLL]        THEN   [ACKPLS]
WHILE   [ACKPLS]
        IF      [SCLH*/SDA]   THEN   [ACKPLSW]
        IF      [SCLH*SDA]    THEN   [WAIT]
WHILE   [ACKPLSW]
        IF      [SCLL]        THEN   [READ7C]
```

Figure 17. Equation entry file slave transmitter (3 of 3)

```
************************************************
*                                              *
*       Simulation Control Language File       *
*                                              *
* Project      : IIC                           *
* Function     : IIC-bus Slave Transmitter     *
*                                              *
* File Name    : IICTRANS.SCL                  *
* Design file  : IICTRANS.EQN                  *
* Pin File     : IICTRANS.PIN                  *
*                                              *
* Date         : March 1993                    *
* Designer     : Aloys Schatorj                *
* Company      : Philips Semiconductors        *
* Department   : PCALE                         *
* Place        : Eindhoven                     *
* Country      : The Netherlands               *
*                                              *
************************************************
*
 P CLK, RESET, SCL, SDA, DATIN, SDAEN, CLOCKOUT, PLOADN, ACKNOW,
#  STARTFF, STATEST, Q0, Q1, Q2, Q3, Q4,
#  RC, CLKEN, ADD0, ADD1, ADD2, VCC
*  SCLH, SCLL,
 PCO
*** Initialisation ***
 BUSI SDA
 BUSI CLK
 BUSO RC
 S  0 (50, 100, ETC)CLK
 S  0 (75)RESET
 S  0 (500, 1000, ETC)SCL
 ST 1 (DATIN)
 ST 1 (VCC)
 ST 1 (SDA)
 ST 0 (CLKEN)
 ST 001  (ADD2, ADD1, ADD0)
 SU TIME = 1225

*** Generate start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** receive device address with read (01000011). ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+4000
 ST 1 (SDA)
 SU TIME = *+2000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000

*** Transmit 8 bits data word 11111111 ***
 ST 1 (DATIN)
 SU TIME = *+8000

*** Generate acknowledge from master ***
 BUSI SDA
 ST 0 (SDA)
 SU TIME = *+1000
```

Figure 18. .SCL File Slave Transmitter (1 of 2)

```
*** Transmit 8 bits data word 11001100 ***
 BUSO SDA
 ST 1 (DATIN)
 SU TIME = *+2000
 ST 0 (DATIN)
 SU TIME = *+2000
 ST 1 (DATIN)
 SU TIME = *+2000
 ST 0 (DATIN)
 SU TIME = *+2000

*** Generate no acknowledge from master (end of transmition) ***
 BUSI SDA
 ST 1 (SDA)
 SU TIME = *+1000

*** Wait for new start condition ***
 SU TIME = *+3000

*** Generate new start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** receive device address with write (01000010). ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+4000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+1000

*** Wait for new start condition ***
 SU TIME = *+3000

*** Generate new start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** Receive wrong device address (0110000). ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+2000
 ST 0 (SDA)
 SU TIME = *+6000

*** Test internal clock ***
 BUSO CLK, RC
 ST 1 (CLK)
 ST 1 (CLKEN)
 SU TIME = *+ 1000
 F
```

Figure 18. .SCL File Slave Transmitter (2 of 2)

```
************************************************
*                                              *
*       Pinning File                           *
*                                              *
* Project      : IIC                           *
* Function     : IIC-bus Slave Transmitter     *
*                                              *
* File Name    : IICTRANS.EQN                  *
* Design file  : IICTRANS.SCL                  *
* Pin File     : IICTRANS.PIN                  *
*                                              *
* Date         : March 1993                    *
* Designer     : Aloys Schatorje               *
* Company      : Philips Semiconductors        *
* Department   : PCALE                         *
* Place        : Eindhoven                     *
* Country      : The Netherlands               *
*                                              *
************************************************
"

Device      =C42VA12
Pin2        =RESET
Pin3        =ADD0
Pin4        =ADD1
Pin5        =ADD2
Pin8        =CLKEN
Pin9        =SCL
Pin10       =SDA
Pin11       =CLOCKOUT
Pin14       =CLK
Pin15       =RC
Pin18       =DATIN
Pin19       =ACKNOW
Pin20       =SDAOE
Pin21       =SDAEN
Pin22       =STATEST
Pin23       =PLOADN
```

Figure 19. .PIN File Slave Transmitter

```
"
************************************************
*                                              *
*               Equation Entry File            *
*                                              *
* Project      : IIC                           *
* Function     : IIC-bus Slave Receiver        *
*                                              *
* File Name    : IICRECEI.EQN                  *
* Design file  : IICRECEI.SCL                  *
* Pin File     : IICRECEI.PIN                  *
*                                              *
* Date         : March 1993                    *
* Designer     : Aloys Schatorj                *
* Company      : Philips Semiconductors        *
* Department   : PCALE                         *
* Place        : Eindhoven                     *
* Country      : The Netherlands               *
*                                              *
************************************************
"
@PINLIST
CLK         B;      "System clock"
RC          B;      "RC input internal clock"
CLKEN       I;      "Clock selection input"
SCL         I;      "IIC-BUS clock"
SDA         B;      "IIC-BUS data"
RESET       I;      "System reset"
ADD0        I;      "Address selection line"
ADD1        I;      "Address selection line"
ADD2        I;      "Address selection line"
CLKSHFT     O;      "Clock to output shift-register"
CLKSTO      O;      "Parallel load into output register"
DATOUT      O;      "Data to output shift register"
DATA7       O;      "Output data RSFF bit7"
STO7        O;      "Store pulse DATA7 FF"
OUT7        O;      "Enable DATA7 FF data"
RESOUT      O;      "Reset output shift register"
ACKNOW      O;      "Acknowledge data"
STATEST     O;      "Reset start and stop FF"

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
CLK.OE       =  CLKEN    ;
RC.OE        =  CLK*CLKEN;
RC           =  GND      ;
CLK          =  RC       ;
STARTFF.CLK  = /SDA      ;
STARTFF.D    =  SCL      ;
STARTFF.RST  =  STATEST  ;
STOPFF.CLK   =  SDA      ;
STOPFF.D     =  SCL      ;
STOPFF.RST   =  STATEST  ;
SCLFF1.D     =  SCL      ;
SCLFF1.RST   =  RESET    ;
SCLFF2.D     =  SCLFF1   ;
SCLFF2.RST   =  RESET    ;
SCLH         = SCLFF1*/SCLFF2 ;
SCLL         = /SCLFF1*SCLFF2 ;
DATA7        =  ((STO7*SDA)+DATA7*/STO7) ;
Q0.RST       =  RESET    ;
Q1.RST       =  RESET    ;
Q2.RST       =  RESET    ;
Q3.RST       =  RESET    ;
Q4.RST       =  RESET    ;
Q5.RST       =  RESET    ;

SDA          =  GND      ;
SDA.OE       =  ACKNOW   ;
DATOUT       =  (SDA*/OUT7) + (DATA7*OUT7) ;
RESOUT       = /RESOUTN ;
```

Figure 20. Equation Entry File Slave Receiver (1 of 3)

```
@INPUT VECTORS

@OUTPUT VECTORS
          [CLKSHFT, CLKSTO, RESOUTN, STATEST, STO7, OUT7]
CLKSHIFT =  1       -       -        -       -     -   B;
CLKSTOR  =  -       1       -        1       -     -   B;
RESO     =  -       -       1        -       1     -   B;
STATST   =  -       -       -        1       -     -   B;
STASTO7  =  -       -       -        1       1     -   B;
OUTBIT7  =  -       -       -        -       -     1   B;
OUTCLK7  =  1       -       -        -       -     1   B;

@STATE VECTORS
         [Q5, Q4, Q3, Q2, Q1, Q0] JKFFR
INITL     =  00 H;
INIT      =  3F H;
INIT1     =  3E H;
INIT2     =  3D H;
WAIT      =  3C H;
WAIT1     =  3B H;
ADDBIT6   =  01 H;
ADDBIT5   =  02 H;
ADDBIT4   =  03 H;
ADDBIT3   =  04 H;
ADDBIT2   =  05 H;
ADDBIT1   =  06 H;
ADDBIT0   =  07 H;
RWBIT     =  08 H;
WRIDMOD   =  09 H;
ACKBITTR  =  0A H;
TESTSTA   =  0B H;
TESTSTA1  =  0C H;
TESTSTA2  =  0D H;
OUTB7     =  0E H;
CLKB7     =  0F H;
WRID6     =  10 H;
WRID6C    =  11 H;
WRID5     =  12 H;
WRID5C    =  13 H;
WRID4     =  14 H;
WRID4C    =  15 H;
WRID3     =  16 H;
WRID3C    =  17 H;
WRID2     =  18 H;
WRID2C    =  19 H;
WRID1     =  1A H;
WRID1C    =  1B H;
WRID0     =  1C H;
WRID0C    =  1D H;

@TRANSITIONS
WHILE   [INITL]
        IF      []              THEN    [INIT]
WHILE   [INIT]
        IF      []              THEN    [INIT1]
WHILE   [INIT1]                                 WITH    [RESO]
        IF      []              THEN    [INIT2]
WHILE   [INIT2]                                 WITH    [CLKSTOR]
        IF      []              THEN    [WAIT]
WHILE   [WAIT]                                  WITH    [STATST]
        IF      [STARTFF]       THEN    [WAIT1]
WHILE   [WAIT1]                                 WITH    [STATST]
        IF      []              THEN    [ADDBIT6]
WHILE   [ADDBIT6]
        IF      [SCLH*SDA]      THEN    [WAIT]
        IF      [SCLH*/SDA]     THEN    [ADDBIT5]
WHILE   [ADDBIT5]
        IF      [SCLH*/SDA]     THEN    [WAIT]
        IF      [SCLH*SDA]      THEN    [ADDBIT4]
```

Figure 20. Equation Entry File Slave Receiver (2 of 3)

```
WHILE   [ADDBIT4]
        IF      [SCLH*SDA]      THEN    [WAIT]
        IF      [SCLH*/SDA]     THEN    [ADDBIT3]
WHILE   [ADDBIT3]
        IF      [SCLH*SDA]      THEN    [WAIT]
        IF      [SCLH*/SDA]     THEN    [ADDBIT2]
WHILE   [ADDBIT2]
        IF      [SCLH*/((SDA*ADD2)+(/SDA*/ADD2))]  THEN  [WAIT]
        IF      [SCLH*((SDA*ADD2)+(/SDA*/ADD2))]   THEN  [ADDBIT1]
WHILE   [ADDBIT1]
        IF      [SCLH*/((SDA*ADD1)+(/SDA*/ADD1))]  THEN  [WAIT]
        IF      [SCLH*((SDA*ADD1)+(/SDA*/ADD1))]   THEN  [ADDBIT0]
WHILE   [ADDBIT0]
        IF      [SCLH*/((SDA*ADD0)+(/SDA*/ADD0))]  THEN  [WAIT]
        IF      [SCLH*((SDA*ADD0)+(/SDA*/ADD0))]   THEN  [RWBIT]
WHILE   [RWBIT]
        IF      [SCLH*/SDA]     THEN    [WRIDMOD]
        IF      [SCLH*SDA]      THEN    [WAIT]
WHILE   [WRIDMOD]
        IF      [SCLL]          THEN    [ACKBITTR]
WHILE   [ACKBITTR]                                      WITH    [ACKNOW]
        IF      [SCLL]          THEN    [TESTSTA]
WHILE   [TESTSTA]                                       WITH    [STATST]
        IF      [SCLH]          THEN    [TESTSTA1]
WHILE   [TESTSTA1]                                      WITH    [STASTO7]
        IF      []              THEN    [TESTSTA2]
WHILE   [TESTSTA2]                                      WITH    [STATST]
        IF      [STARTFF]       THEN    [INIT2]
        IF      [STOPFF]        THEN    [INIT2]
        IF      [SCLL]          THEN    [OUTB7]
WHILE   [OUTB7]                                         WITH    [OUTBIT7]
        IF      []              THEN    [CLKB7]
WHILE   [CLKB7]                                         WITH    [OUTCLK7]
        IF      []              THEN    [WRID6]
WHILE   [WRID6]
        IF      [SCLH]          THEN    [WRID6C]
WHILE   [WRID6C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID5]
WHILE   [WRID5]
        IF      [SCLH]          THEN    [WRID5C]
WHILE   [WRID5C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID4]
WHILE   [WRID4]
        IF      [SCLH]          THEN    [WRID4C]
WHILE   [WRID4C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID3]
WHILE   [WRID3]
        IF      [SCLH]          THEN    [WRID3C]
WHILE   [WRID3C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID2]
WHILE   [WRID2]
        IF      [SCLH]          THEN    [WRID2C]
WHILE   [WRID2C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID1]
WHILE   [WRID1]
        IF      [SCLH]          THEN    [WRID1C]
WHILE   [WRID1C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRID0]
WHILE   [WRID0]
        IF      [SCLH]          THEN    [WRID0C]
WHILE   [WRID0C]                                        WITH    [CLKSHIFT]
        IF      []              THEN    [WRIDMOD]
```

Figure 20. Equation Entry File Slave Receiver (3 of 3)

```
*****************************************************
*                                                   *
*       Simulation Control Language File            *
*                                                   *
* Project      : IIC                                *
* Function     : IIC-bus Slave Receiver             *
*                                                   *
* File Name    : IICRECEI.SCL                       *
* Design file  : IICRECEI.EQN                       *
* Pin File     : IICRECEI.PIN                       *
*                                                   *
* Date         : March 1993                         *
* Designer     : Aloys Schatorj                     *
* Company      : Philips Semiconductors             *
* Department   : PCALE                              *
* Place        : Eindhoven                          *
* Country      : The Netherlands                    *
*                                                   *
*****************************************************
 P CLK, RESET, SCL, SDA, DATOUT, CLKSHFT, CLKSTO, RESOUT, ACKNOW,
#  STATEST, DATA7, OUT7, STO7, STARTFF, STOPFF, Q0, Q1, Q2, Q3, Q4, Q5,
#  RC, CLKEN, ADD0, ADD1, ADD2, VCC
*  SCLH, SCLL,
 PCO
*** Initialisation ***
 BUSI SDA
 BUSI CLK
 BUSO RC
 S  0 (50, 100, ETC)CLK
 S  0 (75)RESET
 S  0 (500, 1000, ETC)SCL
 ST 1 (VCC)
 ST 1 (SDA)
 ST 0 (CLKEN)
 ST 001  (ADD2, ADD1, ADD0)
 SU TIME = 1225

*** Generate start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** receive device address with write (01000010) ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+4000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+1000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000

*** Receive 8 bits data word 11111111 ***
 BUSI SDA
 ST 1 (SDA)
 SU TIME = *+8000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000
```

Figure 21. .SCL file slave receiver (1 of 3)

```
*** Receive 8 bits data word 11001100 ***
 BUSI SDA
 ST 1 (SDA)
 SU TIME = *+2000
 ST 0 (SDA)
 SU TIME = *+2000
 ST 1 (SDA)
 SU TIME = *+2000
 ST 0 (SDA)
 SU TIME = *+2000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000

*** Receive 8 bits data word 01110001 ***
 BUSI SDA
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+3000
 ST 0 (SDA)
 SU TIME = *+3000
 ST 1 (SDA)
 SU TIME = *+1000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000

*** Generate new start condition ***
 BUSI SDA
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** receive device address with read (01000011) ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+4000
 ST 1 (SDA)
 SU TIME = *+2000

*** Wait for new start condition ***
 SU TIME = *+3000

*** Generate new start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500

*** receive device address with write (01000010) ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+4000
 ST 1 (SDA)
 SU TIME = *+1000
 ST 0 (SDA)
 SU TIME = *+1000

*** Generate acknowledge from slave ***
 BUSO SDA
 SU TIME = *+1000
```

Figure 21. .SCL File Slave Receiver (2 of 3)

```
*** Generate stop condition ***
 ST 0 (SDA)
 SU TIME = *+500
 ST 1 (SDA)
 SU TIME = *+500
*** Wait for new start condition ***
 SU TIME = *+3000
*** Generate new start condition ***
 ST 1 (SDA)
 SU TIME = *+500
 ST 0 (SDA)
 SU TIME = *+500
*** receive wrong device address (0110000). ***
 ST 0 (SDA)
 SU TIME = *+1000
 ST 1 (SDA)
 SU TIME = *+2000
 ST 0 (SDA)
 SU TIME = *+6000
*** Test internal clock ***
 BUSO CLK, RC
 ST 1 (CLK)
 ST 1 (CLKEN)
 SU TIME = *+ 1000
 F
```

Figure 21. .SCL File Slave Receiver (3 of 3)

```
**************************************************
*                                                *
*       Pinning File                             *
*                                                *
* Project      : IIC                             *
* Function     : IIC-bus Slave Receiver          *
*                                                *
* File Name    : IICRECEI.SCL                    *
* Design file  : IICRECEI.EQN                    *
* Pin File     : IICRECEI.PIN                    *
*                                                *
* Date         : March 1993                      *
* Designer     : Aloys Schatorje                 *
* Company      : Philips Semiconductors          *
* Department   : PCALE                           *
* Place        : Eindhoven                       *
* Country      : The Netherlands                 *
*                                                *
**************************************************
"

Device      =C42VA12
Pin2        =RESET
Pin3        =ADD0
Pin4        =ADD1
Pin5        =ADD2
Pin8        =CLKEN
Pin9        =SCL
Pin10       =SDA
Pin11       =CLKSHFT
Pin14       =CLK
Pin15       =RC
Pin16       =DATOUT
Pin17       =CLKSTO
Pin18       =RESOUT
Pin19       =ACKNOW
Pin20       =OUT7
Pin21       =DATA7
Pin22       =STATEST
Pin23       =STO7
```

Figure 22. .PIN File Slave Receiver

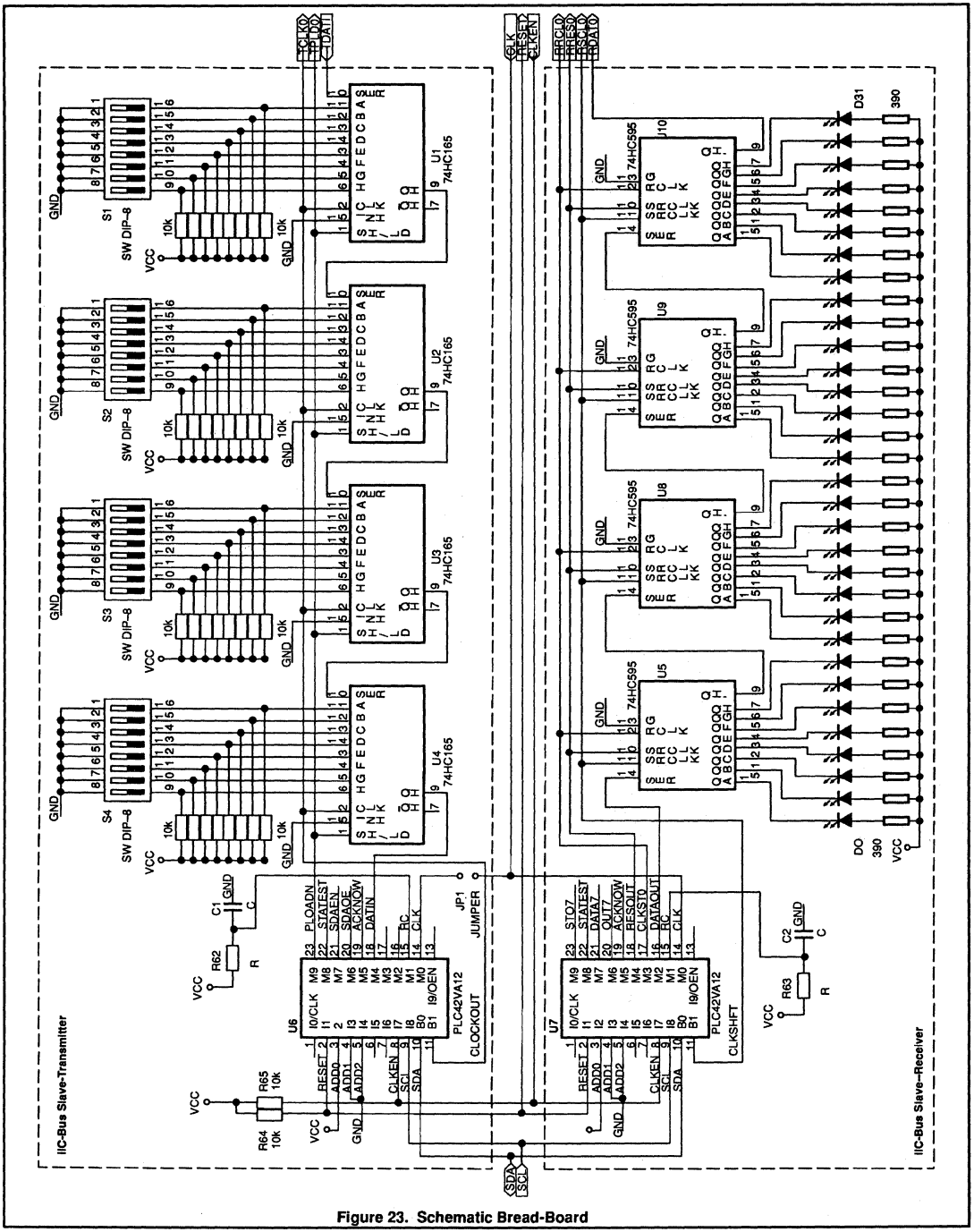


Figure 23. Schematic Bread-Board

ISDN peripheral control

AN037

INTRODUCTION

There are currently numerous integrated circuits available for data communication. They are mostly of a high complexity and their functionality covers nearly all requirements for the development of new telecommunication equipment. However, in many cases, the highly specialized communication ICs do not cover the intended product function exactly, and a few, sometimes simple functions remain to be realized seperately. Those extra parts of the electronics design can often be satisfied by the use of programmable logic circuits,a fact, that shall be demonstrated by this application note.

In order to complete the prototype of a new system board for data transmission, a small controller unit needed to be designed. The functions of this controller is to monitor an incoming serial stream of data bits, indicating certain commands at fixed positions within this bitstream, and to control an output data line in response to these commands.

SPECIFICATION OF THE CONTROLLER FUNCTION

The first function in this controller is that of monitoring an incoming stream of data bits. A general overview of the construction and timing relations of the serial bitstream and associated input control signals is shown in Figure 1. The understanding of this structure is most important for the specification of the controller design. The serial stream of data bits (via SDI – Serial Data In) is synchronous to the clock signal DCL and one data bit has the length of two clock cycles. The bit stream itself is subdivided into single data words of 32 Bits, whereas the beginning and the length of each data word is indicated by the frame signal FSC.

Given this structure, the first task of the controller is to synchronize itself on the data frame and to isolate the data bits number 27...30. These are the so called command indication bits (C/I – Command Indication in Figure 1) serving for the link and network control between the communicating stations.

The internal function of the desired controller is to evaluate the C/I-Bits and to send the appropriate response on the output data stream. The complete sequence of commands during a transmission session is shown in Figure 2, illustrating the order of incoming commands and corresponding output bit pattern. Accordingly to this graphical specification the controller has to detect the two commands PU (Power Up) and DR (Deactivate Request), and on the output side, it has to drive the line SDO (Serial Data Out) in dependence on these commands.

Basically SDO has a constant level 'Low' after the activation of a communication session. But if the input command PU is recognized, the controller shall send a respond command ARN (Active Request None Loop) within the following data frames and at the same bit position as the incoming C/I-Bits. The command ARN on the line SDO is to send till the command DR is detected at the input side. The command DR marks the end of a communication session and one Frame after its detection the line SDO is to switch to the deactive signal level ('High') as it is shown in Figure 2.

Beside the elementary fuctionality, some additional constraints need to be observed. First, it should be considered, that the complete design needs to have an asynchronous part. During the period of no communication, no frame and no clock signal is attached at the inputs and the initial change of the line Start demands an immediate acknowledgment before the clock becomes externally activated. So, the synchronous mode of operation is to extend with an asynchronous design part. The second constraint was the demand that the incoming data are to read with the HL-edge of the second clock cycle (see Figure 1), while a signal change of the output line is to initiate synchronous with the LH-edge of the system clock. This requirement assures the correct recognition of the incoming data also for long distance transmissions with slow rising signal edges, but therefore the design needs to work with two active clock edges.

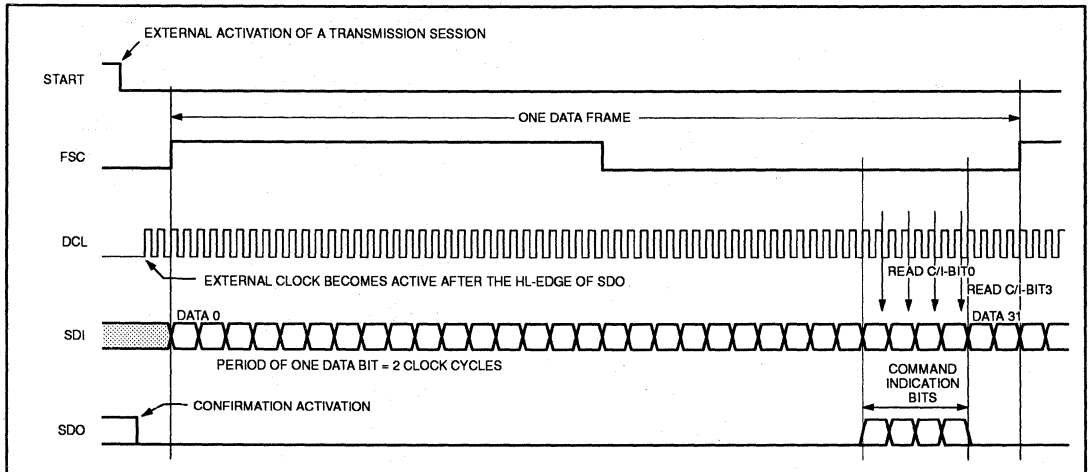


Figure 1. General Structure of the Data Frame and Basic Timing Relations

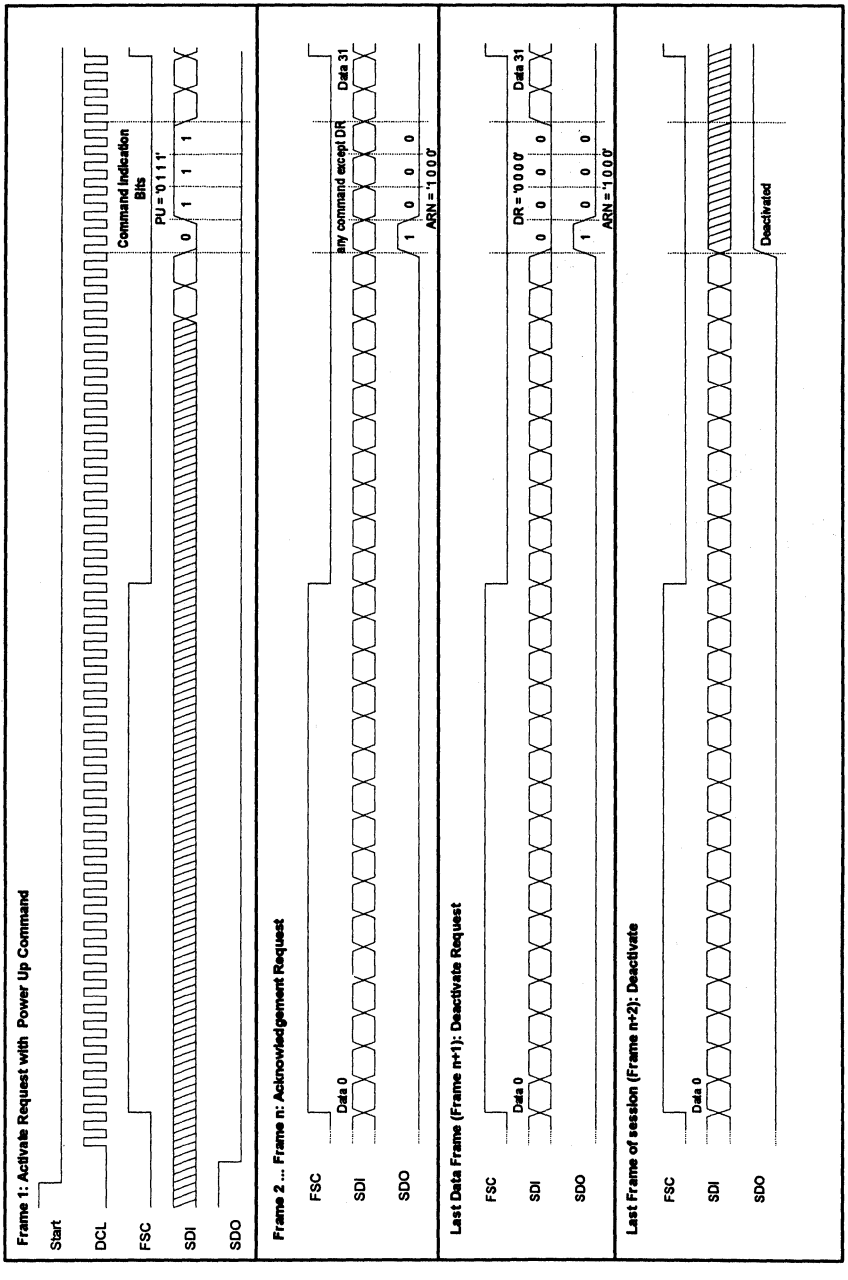


Figure 2. Command Sequence for a Complete Transmission Session

FUNCTIONAL DESIGN DESCRIPTION

Figure 3 shows a possible approach for the realization of the specified controller function. This block structure contains a cyclic 6-Bit-Counter clocked with DCL and synchronized on the frame signal FSC. Refering to the counter value it is possible to determine the position of Command-Bits within input bitstream and so the decoder block 'CI_Decod' can derive the appropriate control signals for Read- and Write-Cycles of Data. The signal CI controls the second essential module, the shift register, via its Enable-Input. The 4-Bit Shift-Register has to read the four Command Bits from the serial bitstream and the following Decoder 'Com_Decod' has just to indicate the two relevant commands, 'PU' and 'DR'. Finally, the real controller is contained within the block 'Control'. There a finite state machine evaluates the 'PU' and 'DR' signals after a new command was read and in correspondence to the actual section of a transmission session the appropriate bit pattern for the output will be generated.

Using this global design description, all constraints can be satisfied easily while realizing the complete controller function. First, the shift-register and the Controller-module can be clocked with the inverted and noninverted DCL-pulse and so different clock edges are taken for the Read- and Write- Cycle of Data Bits. Furthermore, the asynchronous initialization of the controller is accomplished by the Reset-Inputs of the internal Flip-Flop's and the combinatorial output decoder. Finally the clear structure of the design guarantees the complete testability of its circuit implementation.

While a functional specification of each block in this initial design description can easily be created, the final design implementation leads to serious problems. Due to its structure the design requires a sequencer component with one or more combinatorial outputs, and the presence of sequential blocks with two different clocks needs a circuit that fit this condition too. Additionally, the complete design requires a minimum of thirteen internal Flip-Flop's. If these three constraints are taken together an appropriate component can hardly be found. Simple PLDs cannot contain so much multilevel logic. Complex PLDs turned out to be too expensive for this application. So, a different design approach was indicated for this design. The schematic-like block structure was given up for a more abstract, but also compact design description.

DESIGN IMPLEMENTATION

Since the original design couldn't be directly implemented in a simple PLD, a complete revision of its structure had to be carried out. The resulting description file is to see in Figure 4, now given as an abstract HDL-file. The essential advantage of this design description consists of the facts that abstract descriptions are favorable for all kinds of automatic optimization, and that they can easily be adapted to several hardware architectures.

Several changes were made within the design description. First, the counter and the control unit are now merged into one state machine. In doing so, an initial concept was given up. Instead of counting through the whole data frame, the sequencer waits just for the rising and the falling clock edges of the frame signal. So only the second part of a data frame needs to be evaluated and the state machine counts only the steps up to the beginning of the C/I-Bits. Furthermore, some FF's of the shift register are now used twice. While reading the C/I-Bits from the incoming bitstream they have their original function, but during the rest of time they serve as flags. So one FF stores the information about a detected DR-command, while another FF helps to evaluate a frame signal edge timing. Here the feature of two different clock pulses is taken in order to achieve a save mode of operation. All in all the design function is now given in a much more compact description and the abstract description style allows its easy mapping onto different device architectures. So finally a PLC42VA12 was found to be a suitable circuit for the design's realization.

In spite of its general fitting, a successful implementation of the design requires design optimization. An optimal state assignment for the included state machine description as well as a final boolean minimization are absolutely necessary for the design compilation and Figure 5 shows the optimized version, which can now be compiled directly. Figure 7 gives a corresponding Pinning for the PLC42VA12 and Figure 6 shows a simulation output resulted from the implemented circuit model. This part of the simulation represents the beginning of a communication session from the initialisation via the Start-signal up to the first acknowledgement (ARN) on output SDO.

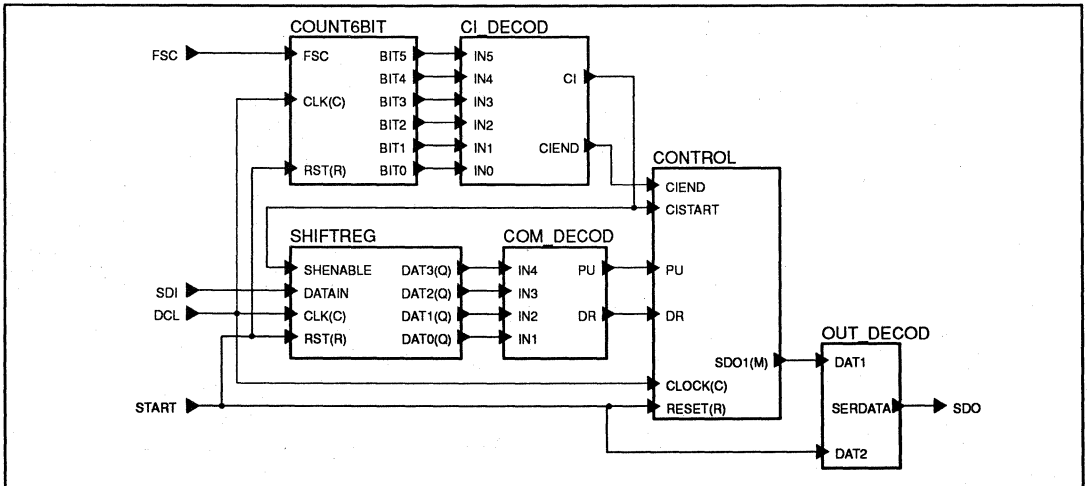


Figure 3. Block Structure of the Controller Design

```
@PINLIST

Start   I ;
FSC     I ;
DCL     I ;
SDI     I ;

SDO     O ;

@LOGIC EQUATIONS

" Shift-Enable Signal defines the Time Slots to read the Command Bits  "

ShEnable =  Q4 * Q3 * Q2 * /Q1 * /Q0
          + Q4 * /Q3 * Q2 * /Q1 * Q0
          + Q4 * /Q3 * Q2 * Q1 * /Q0
          + Q4 * /Q3 * /Q2 * Q1 * Q0 ;

" DR and PU mark the corresponding Commands decoded from the Shift Register "

DR      = /DataBit3 * /DataBit2 * /DataBit1 * /DataBit0 ;

PU      = /DataBit3 * DataBit2 * DataBit1 * DataBit0 ;

" Shift Register - SHIFT operation only when Enable, else HOLD  "

DataBit0.CLK = / DCL ;
DataBit0.RST = / Start ;
DataBit0.J = ShEnable * SDI ;
DataBit0.K =  ShEnable * / SDI
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

DataBit1.CLK = / DCL ;
DataBit1.RST = / Start ;
DataBit1.J = ShEnable * DataBit0 ;
DataBit1.K =  ShEnable * / DataBit0
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

" DataBit2 serves also for the Detection of a new Frame Phase   "

DataBit2.CLK = / DCL ;
DataBit2.RST = / Start ;
DataBit2.J =  ShEnable * DataBit1
            + /Q4 * /Q3 * /Q2 * /Q1 * /Q0 * FSC ;
DataBit2.K =  ShEnable * / DataBit1
            + /Q4 * /Q3 * /Q2 * /Q1 * Q0 * /FSC
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

" DataBit3 serves also as Flag for a detected DR-Command        "

DataBit3.CLK = / DCL ;
DataBit3.RST = / Start ;
DataBit3.J =  ShEnable * DataBit2
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 * DR ;
DataBit3.K = ShEnable * / DataBit2
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 * /DR ;

" The Flag PU stores the switches at the first occurence of the Command PU  "

PU_Flag.CLK  = DCL ;
PU_Flag.RST  = / Start ;
PU_Flag.J    = Q4 * /Q3 * /Q2 * Q1 * Q0 * PU ;
PU_Flag.K    = 0 ;

Q4.CLK  = DCL ;
Q4.RST  = / Start ;

Q3.CLK  = DCL ;
Q3.RST  = / Start ;

Q2.CLK  = DCL ;
Q2.RST  = / Start ;

Q1.CLK  = DCL ;
Q1.RST  = / Start ;

Q0.CLK  = DCL ;
Q0.RST  = / Start ;
```

Figure 4. Complete HDL-Description for the Controller (1 of 3)

```
" Finally the Output Signal      "

SDO =  Start
     + Q4 * Q3 * Q2 * /Q1 * PU_Flag
     + Q4 * /Q3 * /Q2 * /Q1 * /Q0 ;

@INPUT VECTORS

[ DataBit3, DataBit2 ]

FSC_Flag        = -1 B;
NFSC_Flag       = -0 B;
DR_Flag         = 1- B;
NDR_Flag        = 0- B;

@OUTPUT VECTORS

@STATE VECTORS

[ Q4, Q3, Q2, Q1, Q0 ] JKFFR

" state assignment with One Bit Changes for a Minimum of Logic  "

Wait_on_FSC     = 00000 B;
Wait_on_NFSC    = 00001 B;
" Step1           = 00011 B; to much "
Step2           = 00010 B;
Step3           = 00110 B;
Step4           = 00111 B;
Step5           = 00101 B;
Step6           = 00100 B;
Step7           = 01100 B;
Step8           = 01101 B;
Step9           = 01111 B;
Step10          = 01110 B;
Step11          = 01010 B;
Step12          = 01011 B;
Step13          = 01001 B;
Step14          = 01000 B;
Step15          = 11000 B;
Step16          = 11001 B;
Step17          = 11011 B;
Step18          = 11010 B;
Step19          = 11110 B;
Step20          = 11111 B;
DatBit1_0       = 11101 B;
DatBit1_1       = 11100 B;
DatBit2_0       = 10100 B;
DatBit2_1       = 10101 B;
DatBit3_0       = 10111 B;
DatBit3_1       = 10110 B;
DatBit4_0       = 10010 B;
DatBit4_1       = 10011 B;
End_Cycle       = 10001 B;
End             = 10000 B;

@TRANSITIONS

WHILE [ Wait_on_FSC ]
        IF [ FSC_Flag ] THEN [ Wait_on_NFSC ]
                                          " else remain in this state "

WHILE [ Wait_on_NFSC ]
        IF [ NFSC_Flag ] THEN [ Step2 ]

" WHILE [ Step1 ]              "
"        IF [] THEN [ Step2 ] "

WHILE [ Step2 ]
        IF [] THEN [ Step3 ]

WHILE [ Step3 ]
        IF [] THEN [ Step4 ]

WHILE [ Step4 ]
        IF [] THEN [ Step5 ]
```

Figure 4. Complete HDL-Description for the Controller (2 of 3)

```
WHILE [ Step5 ]
        IF [] THEN [ Step6 ]

WHILE [ Step6 ]
        IF [] THEN [ Step7 ]

WHILE [ Step7 ]
        IF [] THEN [ Step8 ]

WHILE [ Step8 ]
        IF [] THEN [ Step9 ]

WHILE [ Step9 ]
        IF [] THEN [ Step10 ]

WHILE [ Step10 ]
        IF [] THEN [ Step11 ]

WHILE [ Step11 ]
        IF [] THEN [ Step12 ]

WHILE [ Step12 ]
        IF [] THEN [ Step13 ]

WHILE [ Step13 ]
        IF [] THEN [ Step14 ]

WHILE [ Step14 ]
        IF [] THEN [ Step15 ]

WHILE [ Step15 ]
        IF [] THEN [ Step16 ]

WHILE [ Step16 ]
        IF [] THEN [ Step17 ]

WHILE [ Step17 ]
        IF [] THEN [ Step18 ]

WHILE [ Step18 ]
        IF [] THEN [ Step19 ]

WHILE [ Step19 ]
        IF [] THEN [ Step20 ]

WHILE [ Step20 ]
        IF [ DR_Flag ] THEN [ End ]
        IF [ NDR_Flag ] THEN [ DatBit1_0 ]

WHILE [ DatBit1_0 ]
        IF [] THEN [ DatBit1_1 ]

WHILE [ DatBit1_1 ]
        IF [] THEN [ DatBit2_0 ]

WHILE [ DatBit2_0 ]
        IF [] THEN [ DatBit2_1 ]

WHILE [ DatBit2_1 ]
        IF [] THEN [ DatBit3_0 ]

WHILE [ DatBit3_0 ]
        IF [] THEN [ DatBit3_1 ]

WHILE [ DatBit3_1 ]
        IF [] THEN [ DatBit4_0 ]

WHILE [ DatBit4_0 ]
        IF [] THEN [ DatBit4_1 ]

WHILE [ DatBit4_1 ]
        IF [] THEN [ End_Cycle ]

WHILE [ End_Cycle ]
        IF [] THEN [ Wait_on_FSC ]

WHILE [ End ]
        IF [] THEN [ End ]
```

Figure 4. Complete HDL-Description for the Controller (3 of 3)

```
@PINLIST

Start    I ;
FSC      I ;
DCL      I ;
SDI      I ;

SDO      O ;

@LOGIC EQUATIONS

" Shift-Enable Signal defines the Time Slots to read the Command Bits  "

 ShEnable =  Q4 * Q3 * Q2 * /Q1 * /Q0
           + Q4 * /Q3 * Q2 * /Q1 * Q0
           + Q4 * /Q3 * Q2 * Q1 * /Q0
           + Q4 * /Q3 * /Q2 * Q1 * Q0 ;

" DR and PU mark the corresponding Commands decoded from the Shift Register "

DR       = /DataBit3 * /DataBit2 * /DataBit1 * /DataBit0 ;

PU       = /DataBit3 * DataBit2 * DataBit1 * DataBit0 ;

" Shift Register - SHIFT operation only when Enable, else HOLD  "

DataBit0.CLK = / DCL ;
DataBit0.RST = / Start ;
DataBit0.J = ShEnable * SDI ;
DataBit0.K =  ShEnable * / SDI
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

DataBit1.CLK = / DCL ;
DataBit1.RST = / Start ;
DataBit1.J =  Q4 * /Q3 * Q2 * /Q1 * Q0 * DataBit0
            + Q4 * /Q3 * Q2 * Q1 * /Q0 * DataBit0
            + Q4 * /Q3 * /Q2 * Q1 * Q0 * DataBit0 ;
DataBit1.K =  Q4 * /Q3 * Q2 * /Q1 * Q0 * /DataBit0
            + Q4 * /Q3 * Q2 * Q1 * /Q0 * /DataBit0
            + Q4 * /Q3 * /Q2 * Q1 * Q0 * /DataBit0
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

" DataBit2 serves also for the Detection of a new Frame Phase   "

DataBit2.CLK = / DCL ;
DataBit2.RST = / Start ;
DataBit2.J =  Q4 * /Q3 * Q2 * Q1 * /Q0 * DataBit1
            + Q4 * /Q3 * /Q2 * Q1 * Q0 * DataBit1
            + /Q4 * /Q3 * /Q2 * /Q1 * /Q0 * FSC ;
DataBit2.K =  Q4 * /Q3 * Q2 * Q1 * /Q0 * /DataBit1
            + Q4 * /Q3 * /Q2 * Q1 * Q0 * /DataBit1
            + /Q4 * /Q3 * /Q2 * /Q1 * Q0 * /FSC
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

" DataBit3 serves also as Flag for a detected DR-Command        "

DataBit3.CLK = / DCL ;
DataBit3.RST = / Start ;
DataBit3.J =  Q4 * /Q3 * /Q2 * Q1 * Q0 * DataBit2
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 * DR ;
DataBit3.K =  Q4 * /Q3 * /Q2 * Q1 * Q0 * / DataBit2
            + Q4 * /Q3 * /Q2 * /Q1 * Q0 * /DR ;

" The Flag PU stores the switches at the first occurence of the Command PU  "

PU_Flag.CLK  = DCL ;
PU_Flag.RST  = / Start ;
PU_Flag.J    = Q4 * /Q3 * /Q2 * Q1 * Q0 * PU ;
PU_Flag.K    = 0 ;

Q4.CLK  = DCL ;
Q4.RST  = / Start ;
Q4.J    = /Q4 * Q3 * /Q2 * /Q1 * /Q0 ;
Q4.K    = Q4 * /Q3 * /Q2 * /Q1 * Q0 ;
```

Figure 5. HDL-Description for the Final Design Implementation (1 of 2)

```
Q3.CLK  = DCL ;
Q3.RST  = / Start ;
Q3.J    = /Q4 * /Q3 * Q2 * /Q1 * /Q0 ;
Q3.K    =  Q4 * Q3 * Q2 * Q1 * Q0 * Databit3
         + Q4 * Q3 * Q2 * /Q1 * /Q0 ;

Q2.CLK  = DCL ;
Q2.RST  = / Start ;
Q2.J    =  /Q4 * /Q3 * /Q2 * Q1 * /Q0
         + Q4 * Q3 * /Q2 * Q1 * /Q0 ;
Q2.K    =  /Q4 * Q3 * Q2 * Q1 * /Q0
         + Q4 * Q3 * Q2 * Q1 * Q0 * DataBit3
         + Q4 * /Q3 * Q2 * Q1 * /Q0 ;

Q1.CLK  = DCL ;
Q1.RST  = / Start ;
Q1.J    =  /Q4 * /Q3 * /Q2 * /Q1 * Q0 * /Databit2
         + /Q4 * Q3 * Q2 * /Q1 * Q0
         + Q4 * Q3 * /Q2 * /Q1 * Q0
         + Q4 * /Q3 * Q2 * /Q1 * Q0 ;
Q1.K    =  /Q4 * /Q3 * Q2 * Q1 * Q0
         + /Q4 * Q3 * /Q2 * Q1 * Q0
         + Q4 * Q3 * Q2 * Q1 * Q0
         + /Q4 * /Q3 * /Q2 * Q1 * Q0
         + Q4 * /Q3 * /Q2 * Q1 * Q0 ;

Q0.CLK  = DCL ;
Q0.RST  = / Start ;
Q0.J    = /Q4 * /Q3 * /Q2 * /Q1 * /Q0 * DataBit2
         + /Q4 * /Q3 * Q2 * Q1 * /Q0
         + /Q4 * Q3 * Q2 * /Q1 * /Q0
         + /Q4 * Q3 * /Q2 * Q1 * /Q0
         + Q4 * Q3 * /Q2 * /Q1 * /Q0
         + Q4 * Q3 * Q2 * Q1 * /Q0
         + Q4 * /Q3 * Q2 * /Q1 * /Q0
         + Q4 * /Q3 * /Q2 * Q1 * /Q0 ;
Q0.K    =  /Q4 * /Q3 * /Q2 * /Q1 * Q0 * /DataBit2
         + /Q4 * /Q3 * Q2 * /Q1 * Q0
         + /Q4 * Q3 * Q2 * Q1 * Q0
         + /Q4 * Q3 * /Q2 * /Q1 * Q0
         + Q4 * Q3 * /Q2 * Q1 * Q0
         + Q4 * Q3 * Q2 * Q1 * Q0 * DataBit3
         + Q4 * Q3 * Q2 * /Q1 * Q0
         + Q4 * /Q3 * Q2 * Q1 * Q0
         + Q4 * /Q3 * /Q2 * /Q1 * Q0 ;

" Finally the Output Signal     "

SDO =  Start
     + Q4 * Q3 * Q2 * /Q1 * PU_Flag
     + Q4 * /Q3 * /Q2 * /Q1 * /Q0 ;
```

Figure 5. HDL-Description for the Final Design Implementation (2 of 2)

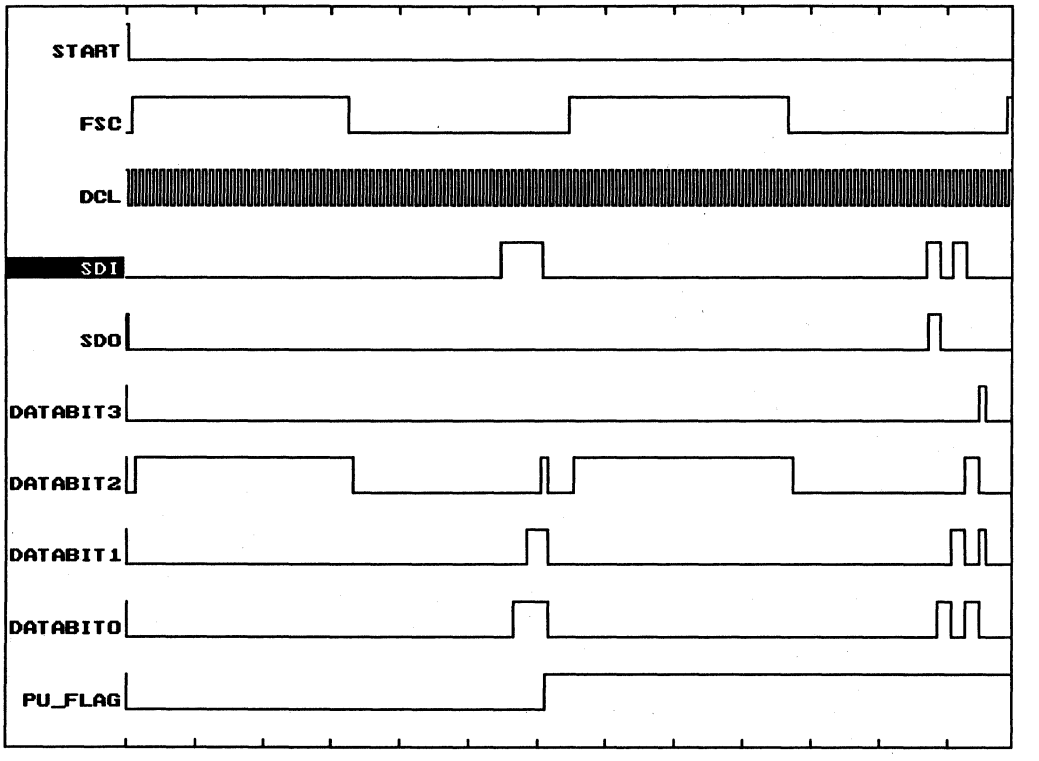


Figure 6. Simulator Output of the Modeled Circuit Implementation

```
Device      = C42VA12

Pin1        = DCL
Pin2        = START
Pin3        = FSC
Pin4        = SDI
Pin14       = SDO
```

Figure 7. Pinlist for the Controller Implementation

```
*************************************************
*          Output of Updsim Version 1.85        *
* Date: 02/04/93                 Time: 13:56:28 *
*************************************************
*                                               *
* Input File Name   : APPNOTE3.net              *
* Output File Name  : APPNOTE3.SCL              *
*                                               *
*************************************************
*
 P START, FSC, DCL, SDI, PU_Flag, DataBit3, DataBit2, DataBit1, DataBit0,
# SDO
 PCO
*
 S 1 ( 500 ) START
 S 0 ( 1000, 33000, 65000, 97000, 129000, 161000, 192000, 224000 ) FSC
 S 1 ( 500, 1000, etc) DCL
 S 0 ( 55000, 61000, 117000, 119000, 121000, 123000 ) SDI
 SU time = 300000
*
 F
```

Figure 8. SCL

SUMMARY

The example of the developed controller has shown that even relative complex designs can be realized with quite small PLDs. Especially if sequential control functions or irregular logic is to be implemented. Hardware programmable logic ICs are often the most suitable solution and sometimes the use of PLDs can simplify the development of new boards and systems significantly.

In addition, the example also illustrates the great effect, which can be achieved by certain design styles and by an appropriate optimization of designs. An initial design description has an essential influence on the final network and its implementation and so it affects the requirements for a component as well as the whole projects costs. By optimization a designer can reduce the amount of gates for a certain design too, leading to a much more efficient use of the given components.

PLD programmable retriggerable one-shot AN011

FEATURES

- Programmable pulse-width/delay
- Maximum 256 clock cycles
- Asynchronous TRIGGER input
- Active-High and Active-Low outputs
- Asynchronous RESET
- 20-pin package

THEORY OF OPERATION

The one-shot consists of a PLC42VA12 and an external clock which may be part of the system in which this one-shot is to work. As shown in Figures 1 and 3 the PLD is configured to have a latch and an eight-bit binary up counter which is presettable by input data to any number less than 256. Since the input data is inverted before it is loaded into the registers, counting from the complements of the input to FF will give the correct number of counts as counting from the input down to 00.

Pulse-width/delay inputs may be the outputs of another device or switches. When /RESET goes Low, flip-flops are set to all 1's (terms PB, PA, and PM0). At the rising edge of the next clock, data is latched into the registers (terms LB, LA, and LM0). When /TRIG goes Low, it is latched into the input latch formed by term # 0, 1, 2 and 13. The output O1 of the latch goes High and O2 goes Low which enables the 8-bit counting cycle. The O1 and /O1 will maintain their output levels until the end of the counting cycle at which time the counter reaches the count FF, resets the latch by term # 13, and sets O2 High. At the rising edge of the next clock, terms LA, LB, and LM0 cause data to be loaded again into the registers, and the device is ready for another /TRIG input. The output waveforms are illustrated in Figure 2.

If the /TRIG pulse-width is longer than the desired pulse-width of the one-shot, the device will react as mentioned above, and at the end of the count cycle new data will be loaded, another count cycle begins while the outputs remain set by the /TRIG input without changing throughout the change-over of one count cycle to another. O1a, on the other hand, will go Low for one clock period at the change-over. As long as the /TRIG is Low, O1a will continue to pulse Low for one clock period at the change-over of one count cycle to another. The output O2 will pulse High for one clock cycle at the change-over. Figure 2 illustrates output wave-forms for both cases. The output wave-forms are as illustrated in Figure 2.

The one-shot is implemented by programming the PLC42VA12 as shown by the SNAP listing in Figures 3 and 4.

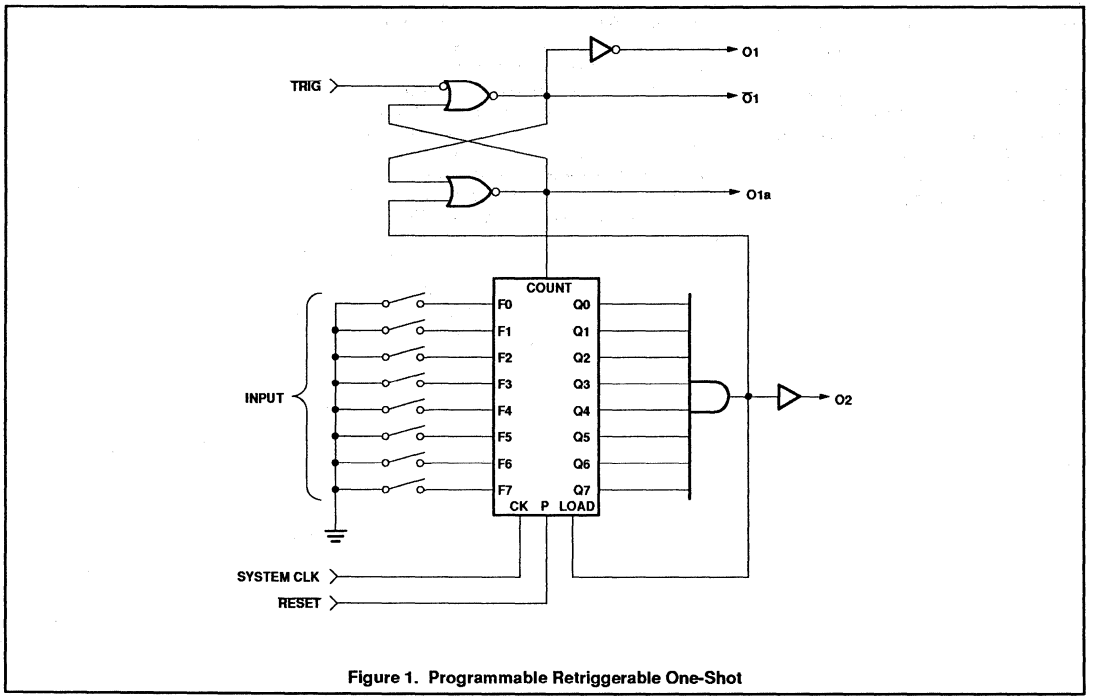
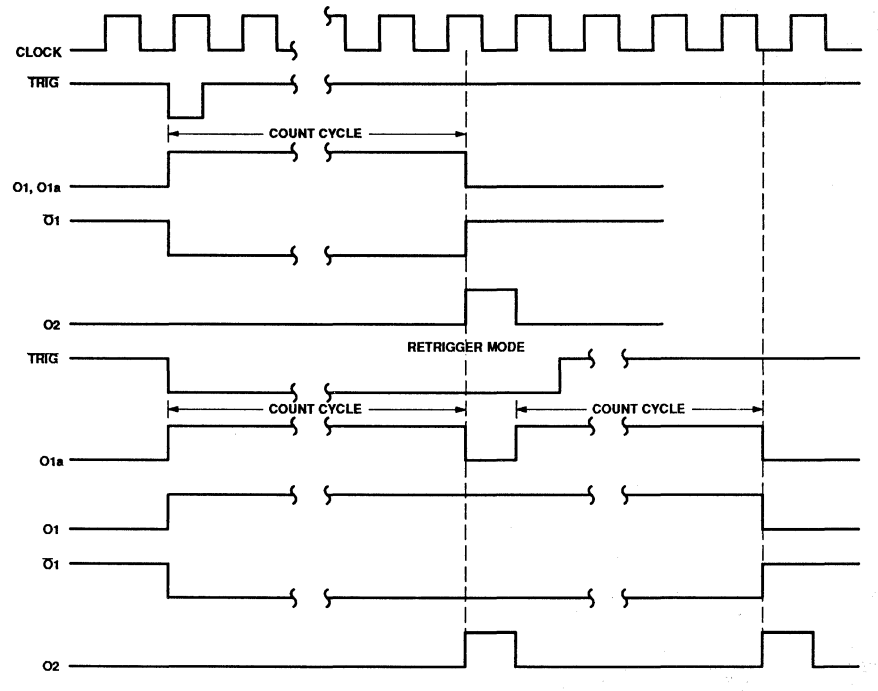
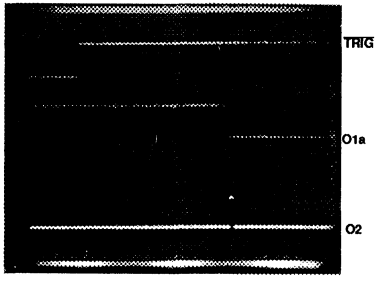


Figure 1. Programmable Retriggerable One-Shot



Programmable One-Shot



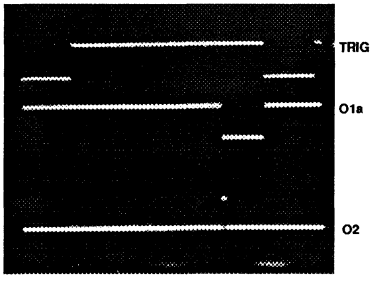


Figure 2. Timing Diagram of Programmable Retriggerable One-Shot

```
*************************************************************
*       PLC42VA12 24-Pin DIP Package Pin Layout             *
* Date: 08/10/93                          Time: 13:02:11 *
*************************************************************

                      _________\_____/_________
                     |                         |
                     |                         |
          CLK [ 1|IO/CLK                 VCC|24]
        RESET [ 2|I1                      M9|23] O2
         TRIG [ 3|I2                      M8|22] NO1
              [ 4|I3                      M7|21] INF7
              [ 5|I4                      M6|20] INF6
              [ 6|I5                      M5|19] INF5
              [ 7|I6                      M4|18] INF4
              [ 8|I7                      M3|17] INF3
              [ 9|I8                      M2|16] INF2
           O1 [10|B0                      M1|15] INF1
          O1A [11|B1                      M0|14] INF0
              [12|GND                19/OE_|13]
                 |_____________________________|
```

Figure 3. Pin Layout

```
"--------------------------------------"
" Programmable Retriggerable One-Shot "
"--------------------------------------"

" This design is for a PLC42VA12 device.
  A similar type of function may be programmed into
  any of the PLS155/7/9A type devices. These devices
  contain a flip-flop preload function which may be
  controlled by input pins and TTL voltage levels or
  by feedback into the array from the flip-flops outputs.

  This one-shot loads the data at the INF0-INF7
  input pins into the counter at the end of the clock
  cycle (O2 = HIGH). If TRIG input is LOW longer than
  the count cycle, output O1A will go LOW for one
  clock period and will go HIGH again for another count
  cycle. Outputs NO1 and O1 stay LOW and HIGH
  respectively until TRIG goes HIGH and the count
  cycle is completed without interruption.
"

@pinlist

clk         i;
reset       i;
trig        i;
inf[7..0]   i;
o1          o;
o2          o;
no1         b;
o1a         b;

@logic equations

"------------------------------"
" equations for latch circuit "
"------------------------------"

o1      =  (no1+/trig);
o1a     = /(no1+/trig);
o1a.oe = 1;
no1     = /((f7*f6*f5*f4*f3*f2*f1*f0)
        +  o1a);
no1.oe = 1;
"---------------------------"
" count comparison equation "
"---------------------------"

o2      = (f7*f6*f5*f4*f3*f2*f1*f0);
```

Figure 4. SNAP Listing (1 of 2)

```
"--------------------------------------"
" equations to load counter from pins "
"--------------------------------------"

"use register preload feature"

temp[15..0].ld  = (f7*f6*f5*f4*f3*f2*f1*f0);
temp14          = inf7;
temp15          = /inf7;
temp12          = inf6;
temp13          = /inf6;
temp10          = inf5;
temp11          = /inf5;
temp8           = inf4;
temp9           = /inf4;
temp6           = inf3;
temp7           = /inf3;
temp4           = inf2;
temp5           = /inf2;
temp2           = inf1;
temp3           = /inf1;
temp0           = inf0;
temp1           = /inf0;

"------------------"
" counter equations"
"------------------"

"The counter is constructed using the toggle feature of
 JK flip-flops. Both J and K are connected to the same
 product term so only eight product erms are required to
 implement this counter. The '+tempXX' input does not require
 a product term in the 42VA12 or PLS155/7/9A type devices
 due to the wire-or register preloading feature. This
 feature is controlled internally in the device by the
 LA and LB product terms. SNAP automatically uses these
 control terms to implement the preload.
"

f0.j    = ((no1*reset)
           +temp1);
f0.k    = ((no1*reset)
           +temp0);
f1.j    = ((no1*f0*reset)
           +temp3);
f1.k    = ((no1*f0*reset)
           +temp2);
f2.j    = ((no1*f1*f0*reset)
           +temp5);
f2.k    = ((no1*f1*f0*reset)
           +temp4);
f3.j    = ((no1*f2*f1*f0*reset)
           +temp7);
f3.k    = ((no1*f2*f1*f0*reset)
           +temp6);
f4.j    = ((no1*f3*f2*f1*f0*reset)
           +temp9);
f4.k    = ((no1*f3*f2*f1*f0*reset)
           +temp8);
f5.j    = ((no1*f4*f3*f2*f1*f0*reset)
           +teml1);
f5.k    = ((no1*f4*f3*f2*f1*f0*reset)
           +teml0);
f6.j    = ((no1*f5*f4*f3*f2*f1*f0*reset)
           +teml3);
f6.k    = ((no1*f5*f4*f3*f2*f1*f0*reset)
           +teml2);
f7.j    = ((no1*f6*f5*f4*f3*f2*f1*f0*reset)
           +teml5);
f7.k    = ((no1*f6*f5*f4*f3*f2*f1*f0*reset)
           +teml4);

f[7..0].clk = clk;
f[7..0].set = reset;
```

Figure 4. SNAP Listing (2 of 2)

Designing with programmable macro logic

INTRODUCTION TO PROGRAMMABLE MACRO LOGIC DESIGN CONCEPTS

Programmable Macro Logic (PML), an extension of the Programmable Logic Array (PLA) concept combines a programming or fuse array with an array of wide input NAND gates wherein each gate folds back upon itself and all other such NAND gates. This is called a foldback NAND structure and its basic elements have been outlined previously (Cavlan[1], Wong[2], Gheissari and Safari[3]).

The choice of an internal NAND logic cell is appropriate because the cell is functionally complete, requiring but a single cell type to generate any Boolean function. A cell within the PLHS501 may be configured to accommodate from one to 32 inputs from the outside world, and up to 72 inputs from within the chip. Because the user can select either direct or inverted input variables, and either a direct or complemented output, the NAND function can generate, with a single pass through the programming array, the basic four logic functions of AND, OR, NAND, NOR. all these basic functions, can be extremely wide, of course (see Figure 1). This convenient structure allows efficient exploitation of all widely used minimization techniques (Karnaugh Maps, Quine-McClusky, Boolean Algebra, etc.).

The obvious extensions to additional combinational functions for decoding, multiplexing and general Boolean functions is straightforward. Adding feedback to the system expands the range of realizable functions to include sequential as well as combinational functions. Figure 2 illustrates the basic arrangement of the PLHS501.

Because of the large number of inputs each NAND gate has available, logic functions that require several levels of conventional 4 or 8 input gates may be able to be reduced to 1 or 2 levels. However, it is important to realize that unlike AND-OR PLD architectures, more than 2 levels of logic may be implemented in the PLHS501 without wasting output or input pins. Up to 72 levels of logic may be implemented due to each of the 72 foldback NAND gates.

So far, the concept of a "macro" is still not evident. Two ways for the generation of a macro exist—namely, hard and soft. Borrowing from the concept in computer programming wherein a section of code (called a macro) is repeated every time its use is required, we can establish subfunctions which can be repeated each time required. The user defined or soft macro can be one which will generate a function by fused interconnect. When a fixed design function is provided, it is a hard macro. This may be an optimized structure like a flip-flop or an adder, or some other function which is generated on the foundation, by the manufacturer. Soft macros are seldom optimized or precisely consistent, but hard macros are both optimized and unalterable.

When a user function for a particular use is isolated, defined and repetition of the function is required, special software constructs are provided which will allow it to be defined at a higher performance and functional density, and an array of choices which contain optimized functions or hard macros will be offered in successor chips. In particular, the PML2552 and PML2852 include an array of flip-flops for state machine design.

Optimizing combinational functions in PML consists largely in making choices and trade-offs. For single output logic functions, the choice is obvious from the truth table. If a particular function's truth table has fewer entries that are logical zeroes than logical ones, product of sums should be chosen and the appropriate OR-AND structure generated. Otherwise, the usual sum of products should be chosen, minimizing as usual, before dropping into the two level AND-OR structure (using the NAND-NAND realization). Combining the availability of inversion at the input and output of the chip, the NAND-NAND structure can perform either the OR-AND or the AND-OR rendition of a function with equal logic levels. The designer needs only to choose the optimal rendition to suit his needs (see Table 1). Truth tables with 50% ones can use either version at the designers whim unless other uses arise.

PERFORMANCE

The PLHS501 (Figure 2) is a high speed, oxide isolated, vertically fused PML device containing 72 internal NAND functions which are combined with 24 dedicated outputs. A large collection of applications, both combinational and sequential, may be configured using this part which looks roughly like a small, user definable gate array. For the sake of clarity, worst case passing a signal from an input, making one pass through the NAND array (output terms) and exiting an output takes around 25 nanoseconds with each incremental pass through the NAND foldback array taking about 8 nanoseconds.

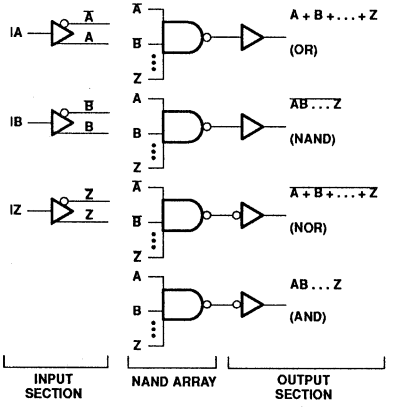


Figure 1. PML Basic Functions

Designing with programmable macro logic

Table 1. Example Demonstration

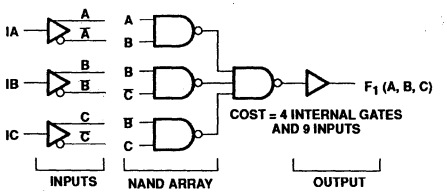
$F1\ (A, B, C) \equiv \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C} + ABC$

A	B	C	f1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

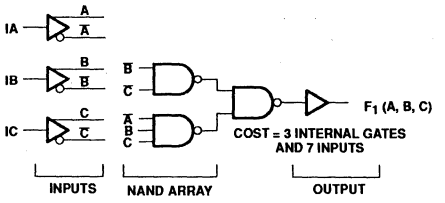
C \ AB	00	01	11	10
0	0	1	1	0
1	1	0	1	1

The optimal choice would be to generate the zero entries

If we group on the one entries we shall get: $AB + \overline{B}C + B\overline{C}$



If we group on the zero entries we get instead: $F_1 = (\overline{B} + \overline{C})(\overline{A} + B + C)$



Designing with programmable macro logic

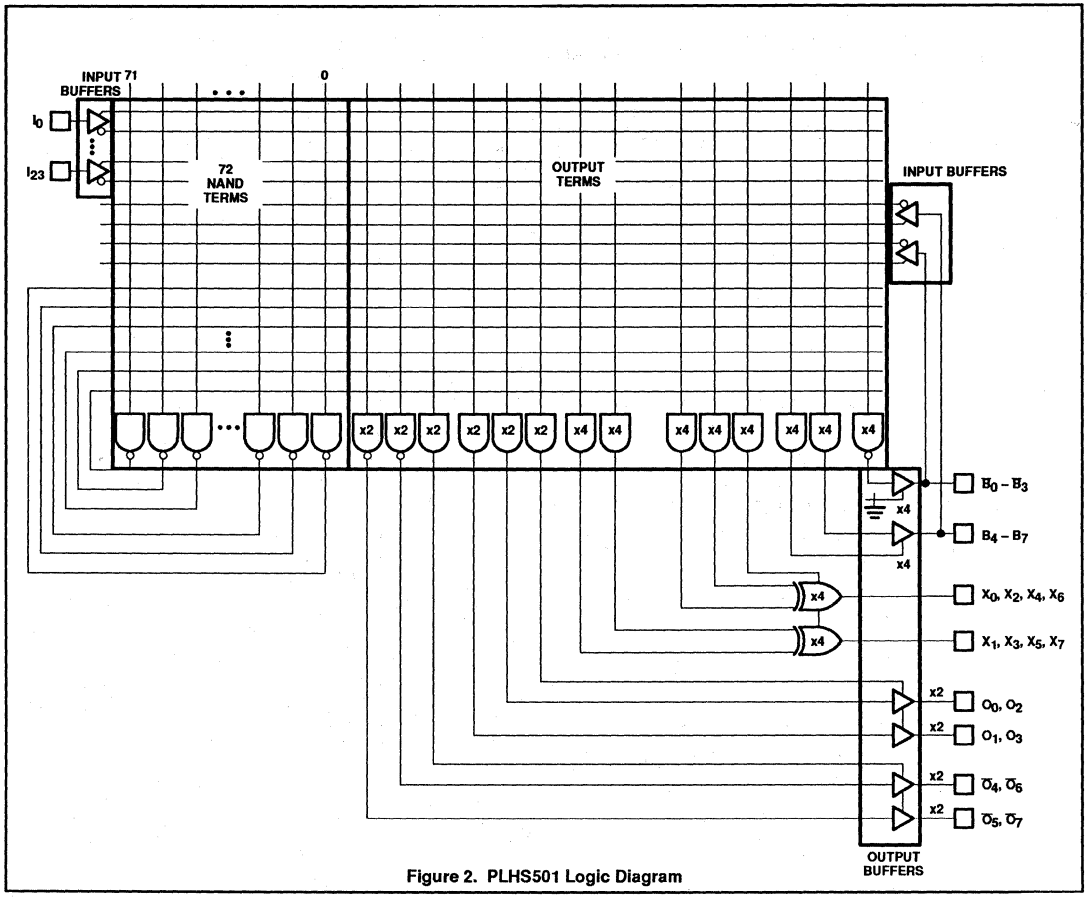


Figure 2. PLHS501 Logic Diagram

Designing with programmable macro logic

The data sheet first lists some maximum propagation delays from an input, through a NAND output term and out through various output gates. Secondly, it lists maximum propagation delays from an input, through a NAND foldback term, through a NAND output term and out through the different output gates.

It is intriguing that subtracting one from the other yields a NAND foldback gate delay of 5 to 6ns when the worst case gate delay of an internal foldback gate is listed as 8ns. This is due to the fact that a gate has less of a delay when its output is falling (t_{PHL}) than when its output is rising (t_{PLH}). When passing a signal through two NAND gates one gate will have less of a delay than the other, and since the individual rise and fall delays are not specified, this causes the apparent discrepancy between the two delays.

Figure 3, Figure 4, Figure 5 and Figure 6 show graphically the timing paths listed in the PLHS501 data sheet.

PLHS501 TIMING

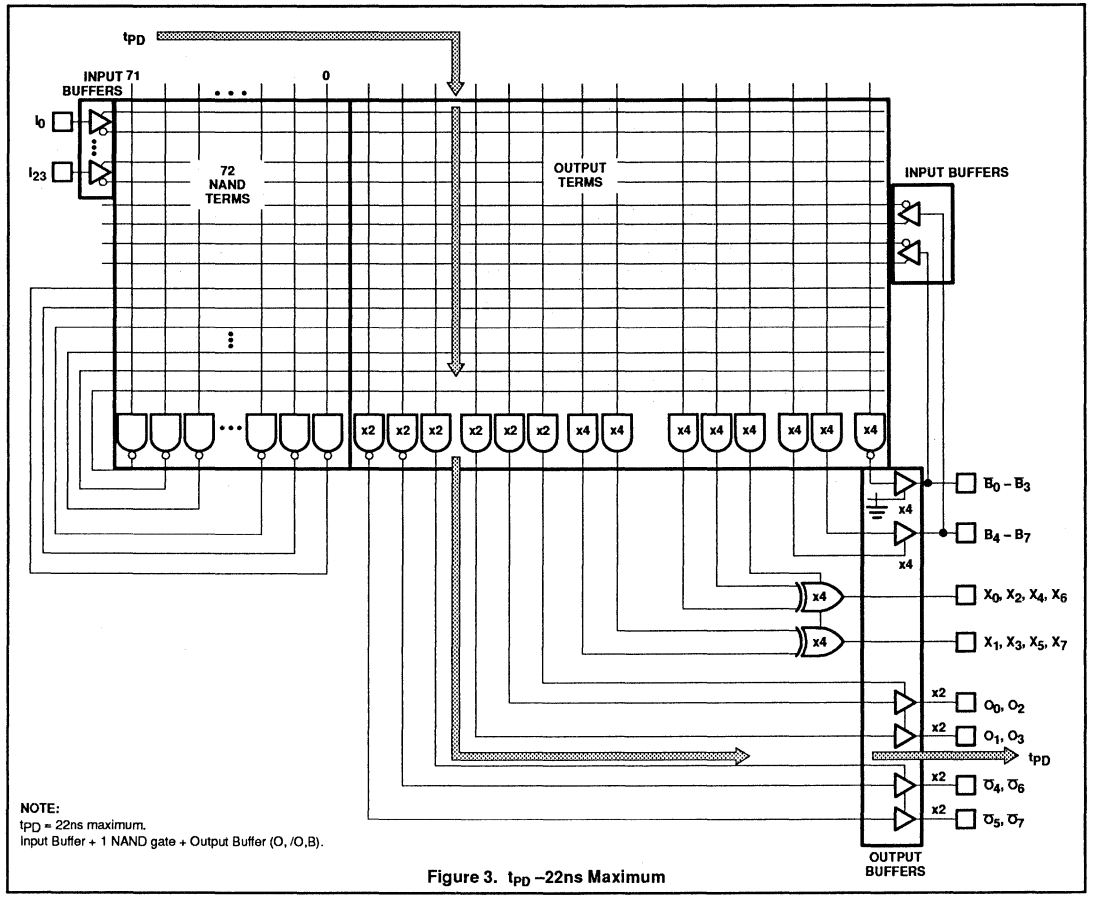


Figure 3. t_{PD} –22ns Maximum

Designing with programmable macro logic

PLHS501 TIMING (Continued)

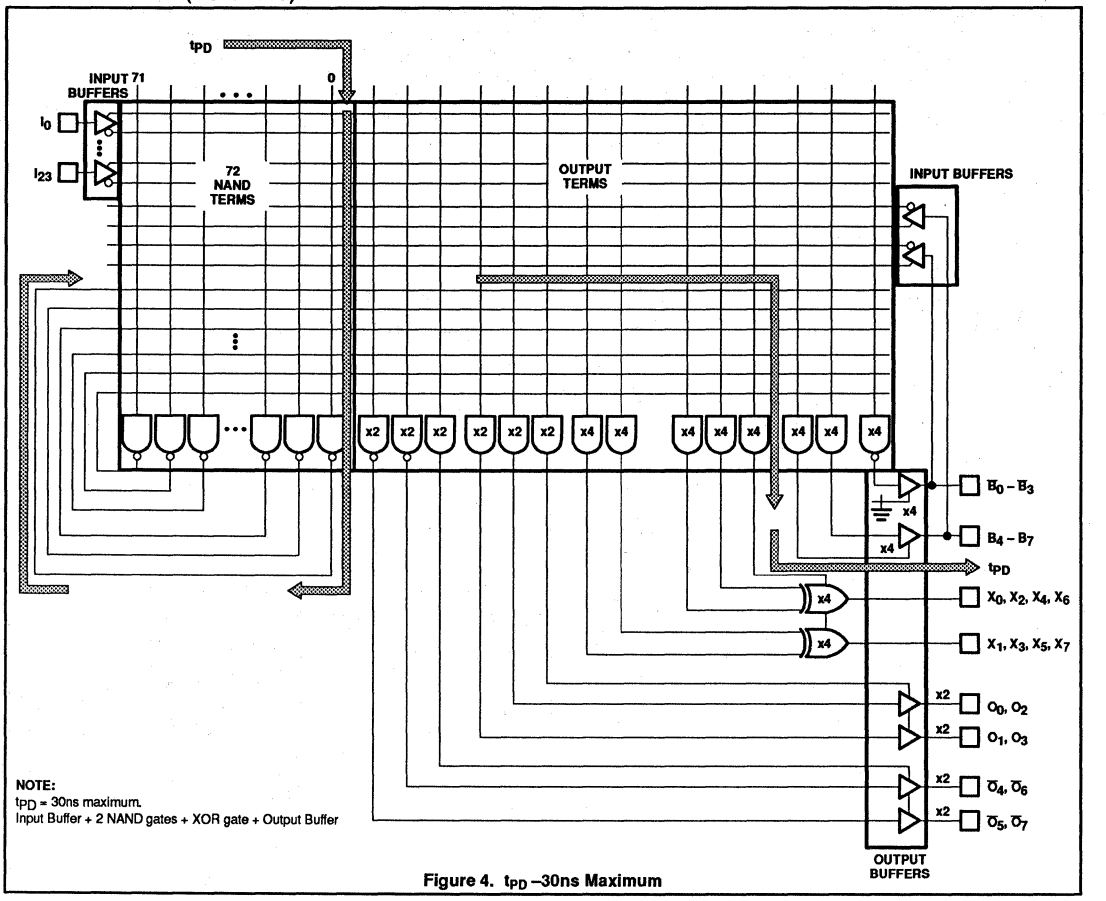


Figure 4. t_{PD} –30ns Maximum

Designing with programmable macro logic

PLHS501 TIMING (Continued)

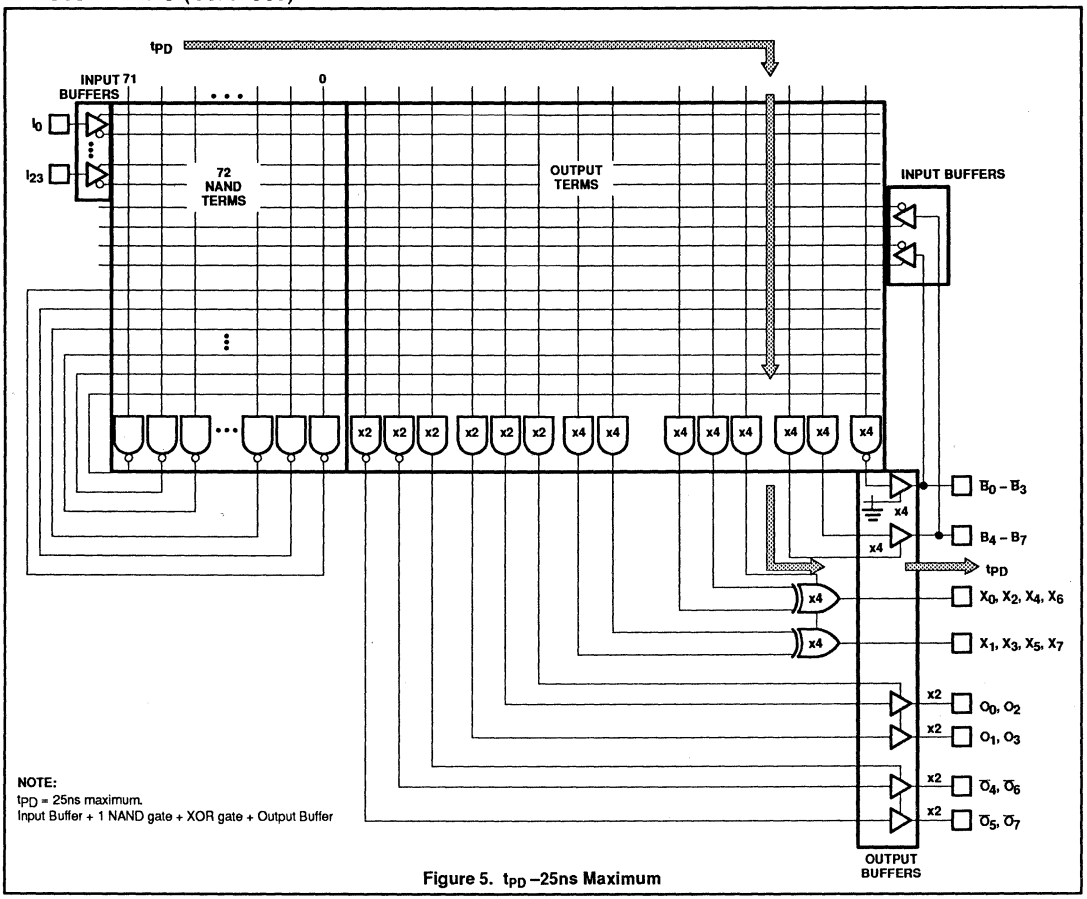


Figure 5. t_{PD} –25ns Maximum

Designing with programmable macro logic

PLHS501 TIMING (Continued)

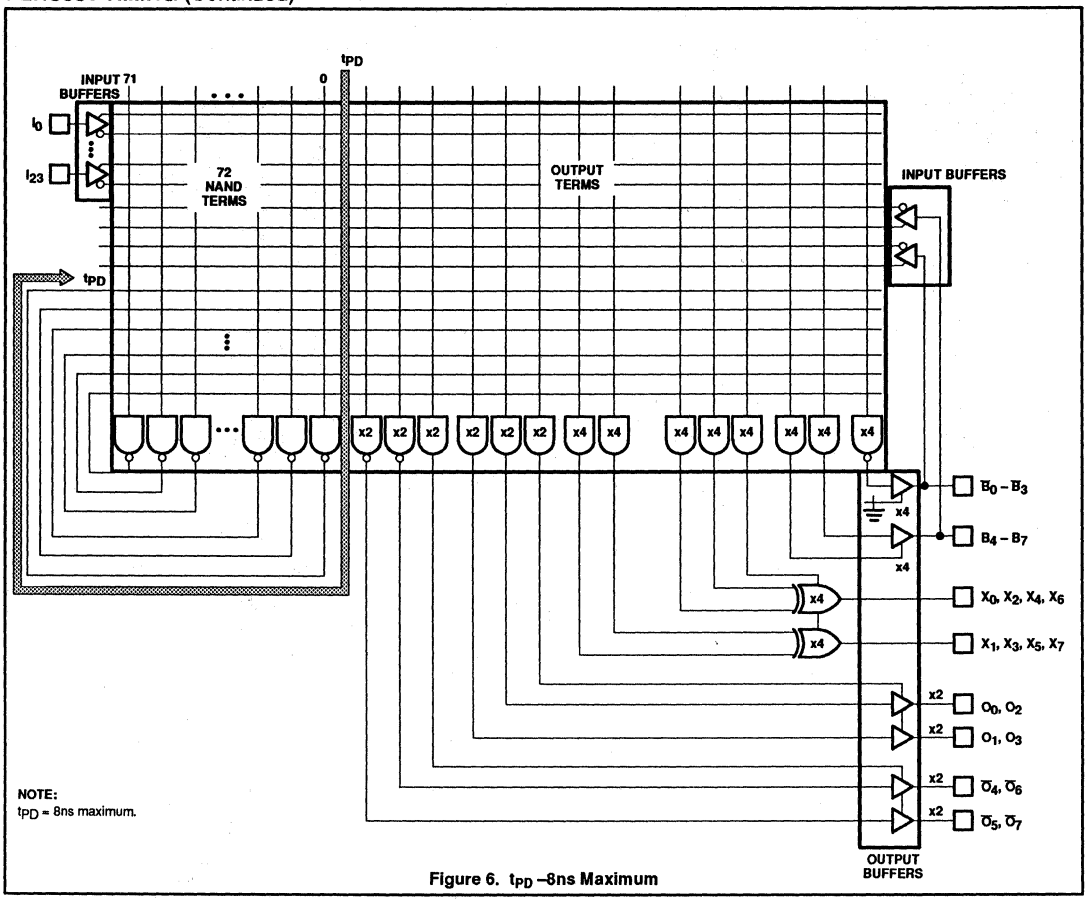


Figure 6. t_{PD} –8ns Maximum

Designing with programmable macro logic

NAND GATE FLIP-FLOPS

Various types of flip-flops and latches may be constructed using the NAND gate building blocks of the PLHS501. A typical 7474 type of edge-triggered D flip-flop requires 6 NAND gates as shown in Figure 7.

No additional gates are required to implement asynchronous set and reset functions to the flip-flop. The equations necessary for SNAP to implement the D flip-flop are shown in Figure 8. However, please note that the equations of Figure 8 define a D flip-flop configured as a divide by 2 (i.e., QN is connected to the data input) whereas Figure 7 shows a general case. Also note that flip-flops with some additional features may be constructed without using more than the six NAND gates. This is possible because of the large number of inputs associated with each NAND gate. For instance, a flip-flop may be required to have a clock gated by one or more signals. Using the PLHS501, it may be implemented by adding additional input signal names to NAND gate equations of gates #2 and #3 of Figure 7. If the data input is to the AND of several signals, extra inputs to NAND gate #4 may be used. Or if additional set or reset lines are required, they may be added simply by using more of the inputs of each NAND gate connected to the main set or reset.

Figure 10 shows two simulations of the same flip-flop. The first one is at a little less than maximum frequency, for clarity in following the waveforms, and the second is at the maximum toggling frequency. For these simulations each NAND gate has a maximum t_{PHL} or t_{PLH} of 8ns (which is the gate delay of a NAND gate in the PLHS501's foldback array). First of all, it can be seen from these simulations that for proper simulation or testing of such a device a set or reset input is mandatory. Both Q and QN outputs are unknown not matter what the inputs do, until they are put into a known state by either a set or reset input. Secondly, various timing parameters such as propagation delay, as well as setup and hold times may be determined.

Therefore, performance of the flip-flop depends a great deal on which gates in the PLHS501 are used, either NAND gates in the foldback array or output NAND gates, connected to bidirectional pins. As a test of the simulation, a D flip-flop connected as a divide by 2 was constructed using only the foldback NAND terms (see Figure 8). An output NAND terms was used to invert the QN output and drive an output buffer. The only inputs were the clock and a reset. The data input to the flop was driven internally by the QN output. According to the simulation, it was possible to drive the clock at a frequency of 25MHz and this small circuit also functioned at that frequency.

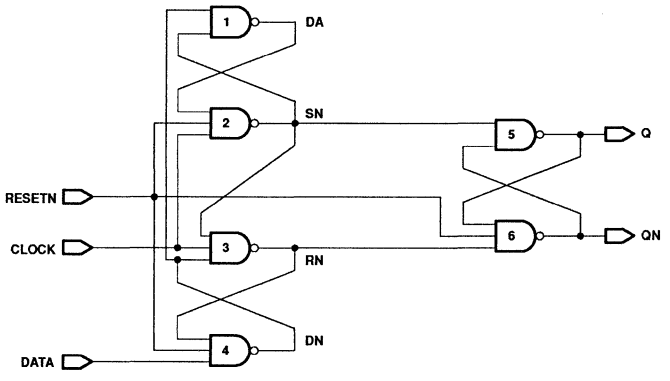


Figure 7. Edge-Triggered D Flip-Flop

Designing with programmable macro logic

```
**********************************************************
*       PLHS501 52-Pin PLCC Package Pin Layout           *
* Date: 10/13/93                        Time: 16:42:21 *
**********************************************************

                          R C
                          S L
                          T K
                      +-+-+-+-+-+-+-+-+-+-+-+-+-+
                      | | | | | | | |5|5|5|4|4|4|
                      |7|6|5|4|3|2|1|2|1|0|9|8|7|
                   +----------------------------------+
                   |    I I I I I I I I I I I I I     |
                   |    1 1 1 1 1 1 1 1 9 8 7 6 5     |
                   |    7 6 5 4 3 2 1 0               |
                   |                                  |
                [ 8|VCC                          VCC|46]
                [ 9|I18                           I4|45]
                [10|I19                           I3|44]
                [11|I20                           I2|43]
                [12|I21                           I1|42]
                [13|I22                           I0|41]
                [14|I23                          /B3|40] OUT
                [15|B4                           /B2|39]
                [16|B5                           /B1|38]
                [17|B6                           /B0|37]
                [18|B7                            X7|36]
                [19|O0                            X6|35]
                [20|GND                          GND|34]
                   |                                  |
                   |          / / / /                 |
                   |    O O O O O O O X X X X X X     |
                   |    1 2 3 4 5 6 7 0 1 2 3 4 5     |
                   +----------------------------------+
                      |2|2|2|2|2|2|2|2|2|3|3|3|3|
                      |1|2|3|4|5|6|7|8|9|0|1|2|3|
                      +-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
@PINLIST
clk i;
rst i;
out o;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

flop.clk =  clk;
flop.d   = /flop;
flop.rst =  rst;
out      =  flop;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 8. SNAP Listing

Designing with programmable macro logic

```
PLHS501
77666666666655555555554444444444333333333322222222221111111111000000000
10987654321098765432109876543210987654321098765432109876543210987654321O
IIIIIIIIIIIIIIIIIIIIIIIIBBBBBBBB
22221111111111000000000000000000
32109876543210987654321076543210
--------------------------------   Col for   /B 0
................................................................A..
HH------------------------------   Col for    P 0
...............................................................A...
-H------------------------------   Col for    P 1
..............................................................A...A
H-------------------------------   Col for    P 2
.............................................................A...A.
--------------------------------   Col for    P 3
..............................................................A...A
H-------------------------------   Col for    P 4
................................................................AA.
--------------------------------   Col for    P 5
.................................................................A.A
```

Figure 9. Partial PLHS501 Fusemap Showing Test Flip-Flop Fusing

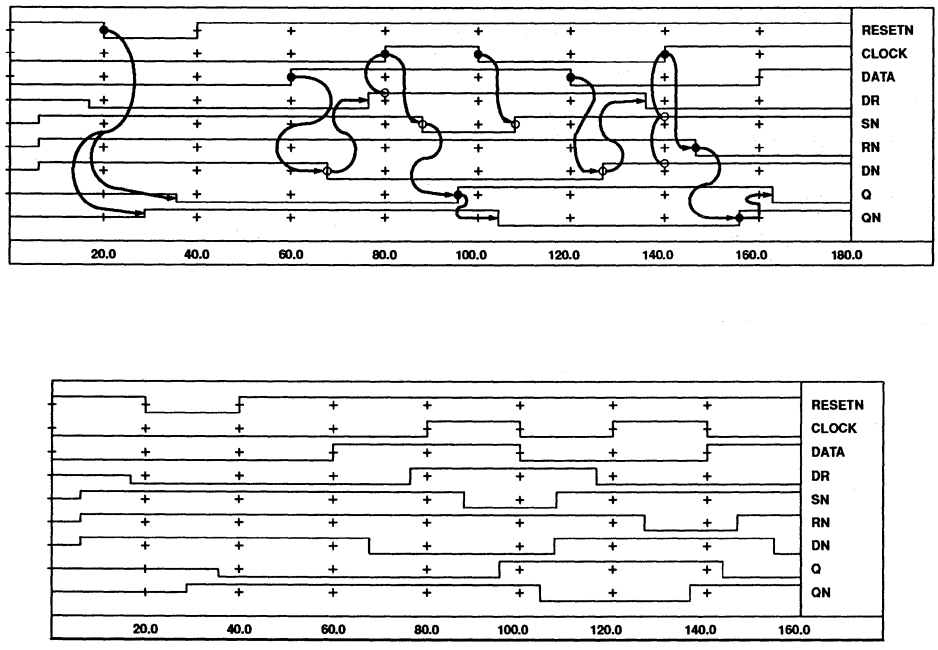


Figure 10. Waveforms of Test Flip-Flop

Designing with programmable macro logic

FUNCTIONAL FIT

In the late 1960's and early 1970's designers used SSI, MSI and small amounts of early LSI to generate logic solutions. Frustrated by the lack of wide input gates to accommodate a lot of product terms for two level solutions, they turned toward the budding ROM and PROM products. These devices relied on literally realizing a function by generating its truth table in silicon. The logic function had to have each logical one and zero realized distinctly as an entry for a particular combination of input variables, usually supplied on the address lines of the memory. Observing that many such truth tables were dense in ones or zeroes and sparse in the remainder, a cadre of initial manufacturers emerged with focus on supplying a programmable product with a few AND gates and OR gates which were versatile enough to compete against the ROM/PROM parts. The gimmick supplied these PLA manufacturers was to illustrate the functional equivalency of the PLA to the PROM by comparing the number of product terms (to be shortened to "p-terms") the PLA supplied and comparing this to the width and depth of available PROMs. P-terms became the "currency" of the PLA world and a designer only had to assess the equivalent number of Boolean product terms required by his function to determine whether a particular PLA was a suitable candidate for his design.

Almost in parallel, gate arrays became available. These provided an array of identical, fixed input gates (usually two input NANDs or NORs). These were generated in a regular fashion on substrate which has a fixed input/output pin arrangement. Also recognizing that all logic functions could be built from the appropriate two input gate, when interconnected correctly, manufacturers offered these devices to customers who required increased density.

The designer's responsibility was to generate what would ultimately be a metal interconnect pattern of his design. Special tools were required to allow an untrained system designer to do this successfully. Flop-flops, decoders, registers, adders, etc., could all be generated from the low level gate building blocks.

The currency of gate arrays became known as gate equivalent functions. That is with limited number of available gates on a substrate, the user needed to know precisely how many gates were used up, on a function by function basis, to generate each piece of his design. A D flip-flop requires about six gates, a D latch four, a 3 to 8 decoder takes about 14 gates and so forth. This allowed estimation regarding whether the function could conceivable be fit onto a particular substrate or not. Manufacturers had to offer multiple foundations to that a designer could be assured that his design would result in a working IC.

The classic method of estimating whether a logic function would fit into a PLA was to determine the number of I/O pads required and the number of product terms required to generate the logical function, then select the PLA. For a gate array, the required measure included the I/O pad arrangement but substituted the number of available gates to generate the logical function (usually by table lookup). In an attempt to reconcile the two measures, Hartman[4] has evolved a formula for his product line. A calculation using this method and developing an appropriate "exchange rate: is shown in Table 2 for the PLHS501 and PLHS502. An alternate method of generating an estimate is to consider the gate equivalent of generating, say for the PLHS501, a gate equivalent of the part in an optimistic functional configuration (72 occurrences of a 32 input NAND gate).

Figure 11 shows how this will result in over 2000 equivalent gates. Conversely, by stacking the NAND gates into D flip-flops, its least efficient function, the PLHS501 will have a gate equivalent of only about 100 gates.

The most rational method of assessing fit is to isolate functions and identify the correct configuration in terms of gates, to allow direct tally of the gates used, to generate the proposed configuration. Table 3 may assist in doing this analysis. Note that all basic gates require precisely one gate to generate the function. Also note the occurrence of functions in the table which could never be generated as standard ICs previously. The procedure is to tally the design against a total budget of 72 multiple input NAND gates.

Table 3 is illustrative only, and should by no means be taken as complete. It may be simply expanded by designing the proposed function with disregard to the usual restrictions on the number of inputs to a gate, realize the function as one, two, three, or more levels of interconnected logic and count the number of gate occurrences required. Special software has been provided to allow pyramided logic structures to be generated under the designer's control. These structures may, however, be no deeper than 72 levels for the PLHS501. Functions should be generated in accord with the guidelines mentioned before, for selecting an optimal 2 level logical solution.

It is an interesting observation that manufacturers of gate arrays and standard cell products which offer embedded PROMs, ROMs or RAMs have not successfully described these embedded functions in terms of equivalent gates, but rather resort to other means (such as divulging their relative area with respect to the area of a basic gate). There is, as yet, no standard in this arena.

Designing with programmable macro logic

Table 2. Equivalency Ratio

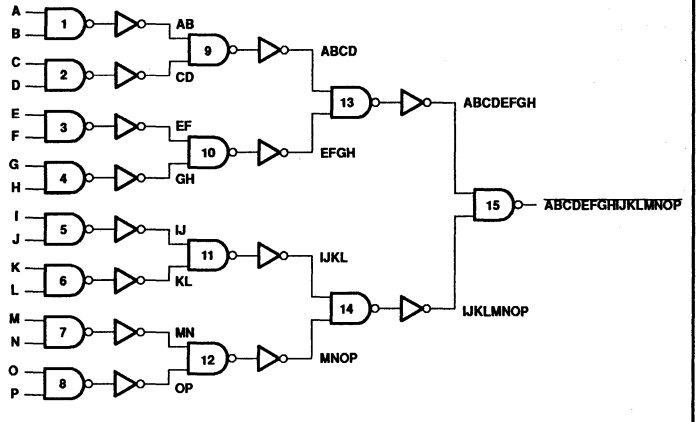
Hartman's method is based on a CMOS gate array equivalency wherein 4 transistors constitute a 2 input NAND or NOR gate, equal to one gate. Thus, his "exchange rate" is as follows:

E.R. = 4 × # inputs
+9 × # FFs
+7 × # 3-State outputs
+(15 to 30) × # OR outputs from the AND/OR array.

For the PLHS501: (using CMOS numbers which may be inappropriate)

E.R. = 4 × 32
+9 × 0
+7 × 24
+(15 to 30) × 50% of 72 feedbacks = 836 to 1376 gates

Being for two bipolar ICs, in this case, the method may be inappropriate, but may be taken as an estimating procedure.



NOTE:
Double this and add one for a 32 input NAND.

Figure 11. 16 Input NAND Formed from 2 Input Gates

Designing with programmable macro logic

Table 3. PLHS501 Gate Count Equivalents

FUNCTION	INTERNAL NAND EQUVAL-ENT	COMMENTS
Gates		
NANDs	1	For 1 to 32 input variables
ANDs	1	For 1 to 32 input variables
NORs	1	For 1 to 32 input variables
ORs	1	For 1 to 32 input variables
Decoders		
3-to-8	8	Inverted inputs available
4-to-16	16	Inverted inputs available
5-to-32	32	Inverted inputs available (24 chip outputs only)
Encoders		
8-to-3	15	Inverted inputs, 2 logic levels
16-to-4	32	Inverted inputs, 2 logic levels
32-to-5	41	Inverted inputs, 2 logic levels, factored solution.
Multiplexers		
4-to-1	5	Inverted inputs available
8-to-1	9	
16-to-1	17	
27-to-1	28	Can address only 27 external inputs - more if internal
Flip-Flops		
D-type Flip-Flop	6	With asynchronous S-R
T-type Flip-Flop	6	With asynchronous S-R
J-K-type Flip-Flop	10	With asynchronous S-R
Adders		
8-bit	45	Full carry-lookahead (four levels of logic)
Barrel Shifters		
8-bit	72	2 levels of logic
Latches		
D-latch	3	2 levels of logic with one shared gate

Programmable macro logic primer AN029

SUMMARY

The evolution of Programmable Logic Devices (PLD's) has led to the birth of a new generation of programmable devices designated as PML (Programmable Macro Logic). The immense versatility of these devices brings them closer as plausible alternatives to semicustom design approaches in low-to-medium ranges of applications. The following paper gives a description of all three PML devices, the PLHS501, PML2552, and the PML2852. In addition, some basic design tips and techniques for PML devices are presented.

THE EMERGENCE OF THE THIRD GENERATION PLD ARCHITECTURE

PML was introduced at WESCON '85 by Philips Semiconductors Corporation. The unique architecture of PML breaks away into a new era of programmable logic devices. The purpose of the PML architecture is to overcome the two level AND-OR bottleneck and provide the user with a higher level of logic integration. Current PLD's rely on two levels of logic transformation to implement combinational logic in Sum-Of-Products (SOP) form. In addition, various PLD's make use of higher level macros such as flip-flops to form sequential logic functions. These macros connect the AND-OR chain to dedicated I/O pins.

Figure 1 show the basic architecture of one of the most recent PAL® devices. It is clear that this architecture is inefficient in making full use of the available on-chip resources. This is due to the fact that an unused I/O macro will be wasted and remains futile.

For example, if an I/O pin is used as an input, the output macros are all wasted. Obviously, such an architecture cannot provide the user with an increase in the levels of logic integration. The PML device takes advantage of the fundamental architecture shown in Figure 2 to overcome these deficiencies and waste of on-chip resources. As shown in Figure 2, PML incorporates the NAND-NAND gate equivalence to break the AND-OR bottleneck.

The core of the PML is the programmable NAND-NAND network which connects the input and output macros to each other. Thus the inputs, outputs, and function macros are all connected by a single array.

The first device is the PLHS501. The seemingly simple structure of this device can implement every logic function furnished by the current PAL/PLA devices. Although the PLHS501 is principally a combinational logic device, its unique architecture makes it an ideal tool for applications involving asynchronous state machines (See Reference 2).

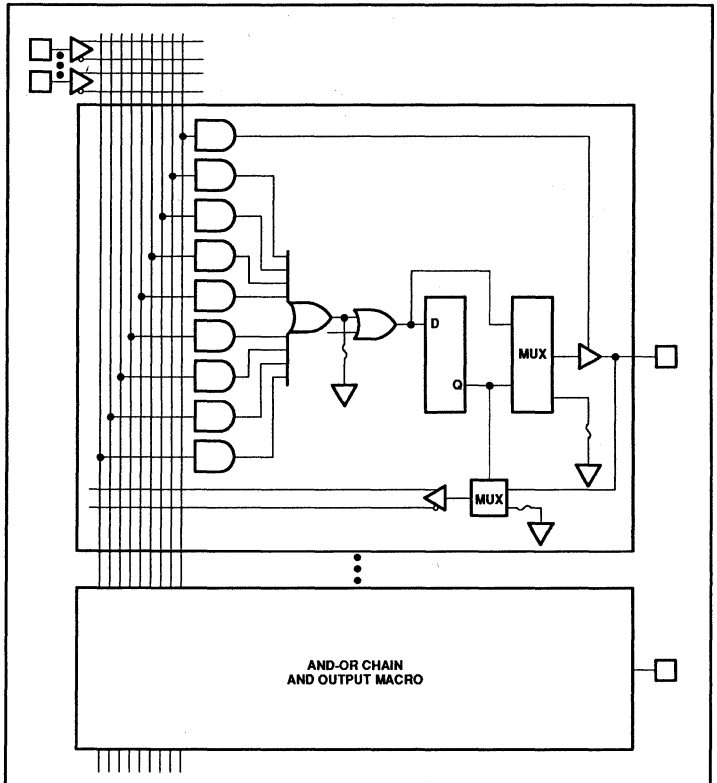


Figure 1. One of the Latest Registered PALs

®PAL is a registered trademark of Advanced Micro Devices, Inc.

Programmable macro logic primer AN029

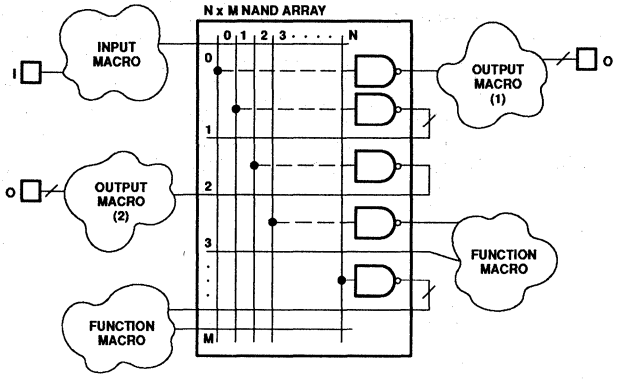


Figure 2. PML Fundamental Architecture

Programmable macro logic primer AN029

THE PLHS501

The PLHS501 architecture in Figure 3 exhibits an exquisite logic tool. The device provides a combination of 72 NAND terms, 24 dedicated inputs (I0–I23), eight bidirectional I/O's (B0–B7), eight exclusive-OR outputs (X0–X7), and eight dedicated outputs (O0–O7).

Since the output of each NAND term feeds back to the inputs of the NAND array, intricate logic functions can be implemented without wasting valuable I/O pins. For example, in order to implement an internal 'RS' latch in a combinational PAL/PLD, at least two inputs and two outputs are required. The same internal latch can be configured by the PLHS501 without using any I/O pins.

The shorthand notation of Figure 3 hides something with which many designers have been impressed in the PLHS501, the wide input NAND gates. Figure 4 shows just how wide the internal NANDs are, from a logical viewpoint. Each NAND can accommodate up to 32 external inputs and 72 internal inputs. Hence the part is ideal for wide decoding of 32-bit address and data busses. With 72 copies of the wide NAND, the PLHS501 is often compared against low–end gate arrays.

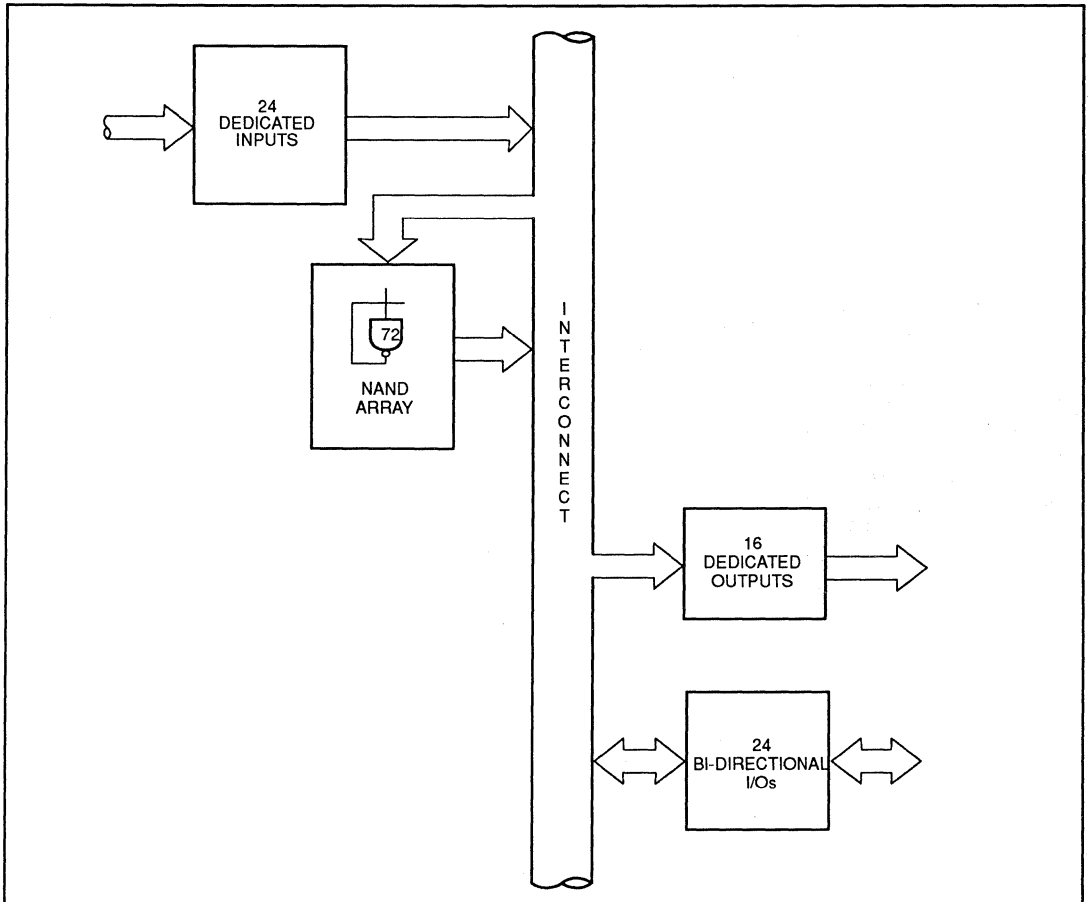


Figure 3. PLHS501 Functional Block Diagram

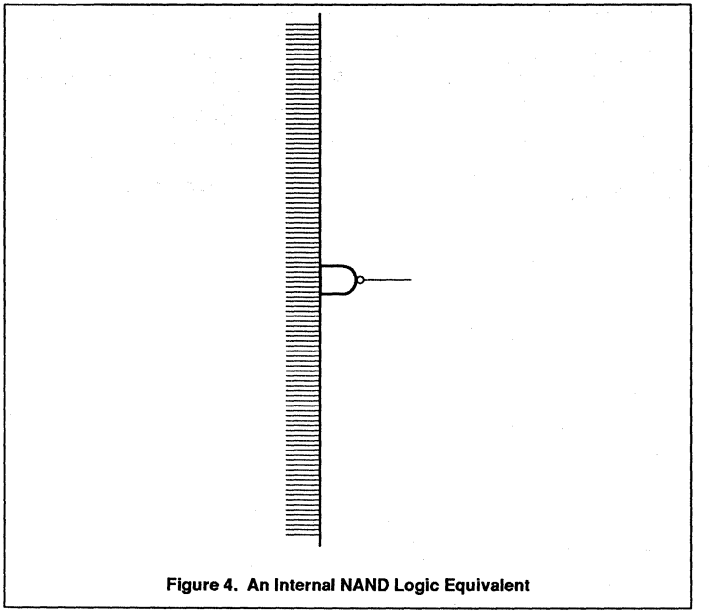


Figure 4. An Internal NAND Logic Equivalent

LATCHES

Figure 5 illustrates how 'RS' and 'D' latches are implemented in the PLHS501.

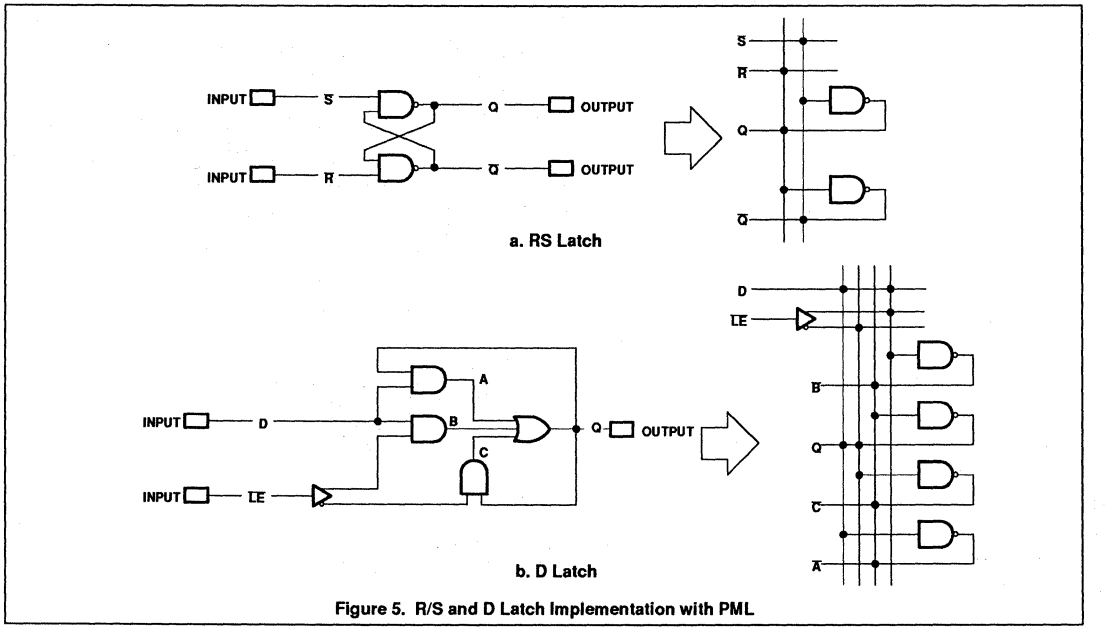


Figure 5. R/S and D Latch Implementation with PML

Flip-Flop Merging

Figure 6a shows the positive edge triggered D flip-flop structure. By putting a two-level AND/OR structure in front of the data input, the D flip-flop can be steered from state to state.

Figure 6b shows such an input structure realized from a two-level NAND gate section.

Figure 6c shows this "AND-OR" structure rolled inside of the flip-flop. The gating was merged with the flip-flop inwards to make a faster, composite function. Whereas this may appear as a trick to the uninitiated, this degree of flexibility allowed gate array designers to merge a multitude of logic into a fixed foundation. For highest efficiency, similar thinking allows the designer to break up decoders and multiplexers into their building blocks and generate only the pieces needed.

PLHS501 DESIGN EXAMPLE

The following example intends to manifest the capabilities of the PLHS501. Figure 7 shows a system formed with TTL logic. The system requirements make it imperative only to use discrete asynchronous latches. Thus, none of the 7 latches in the system can be directly replaced by registers. The system is partitioned into two PLS173s and one PLS153. The specified PLD's are labeled with the same labels as those on the system schematic (Figure 7). Figure 8 shows the overall system implemented with PLDs. The logic condensation capabilities of PML makes it feasible to replace the whole system by a single PLHS501 (Figure 9). The PLHS501 in this design will still have ample space for any future additions.

The above example demonstrates only part of the PLHS501 capabilities. The introduction of PML devices and their immense logic power will pave the way for a new generation of efficient and elegant systems.

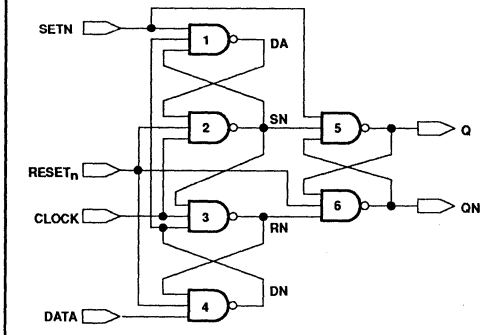


Figure 6a. Positive Edge Triggered D-Flip-Flop with Reset and Set

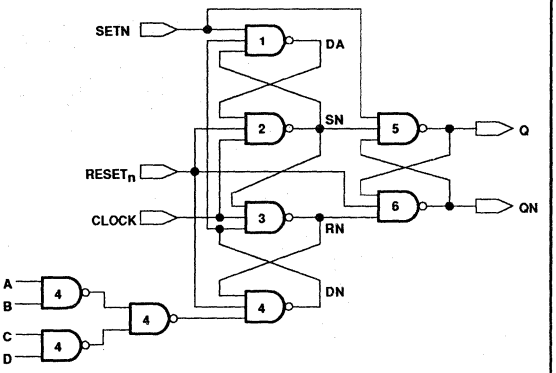


Figure 6b. As in (A), with Input AND-OR Function

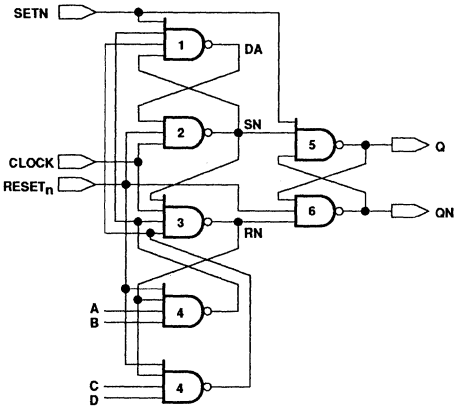


Figure 6c. As Above, with Integral AND-OR Input Function

Figure 6. Flip-Flop Merging

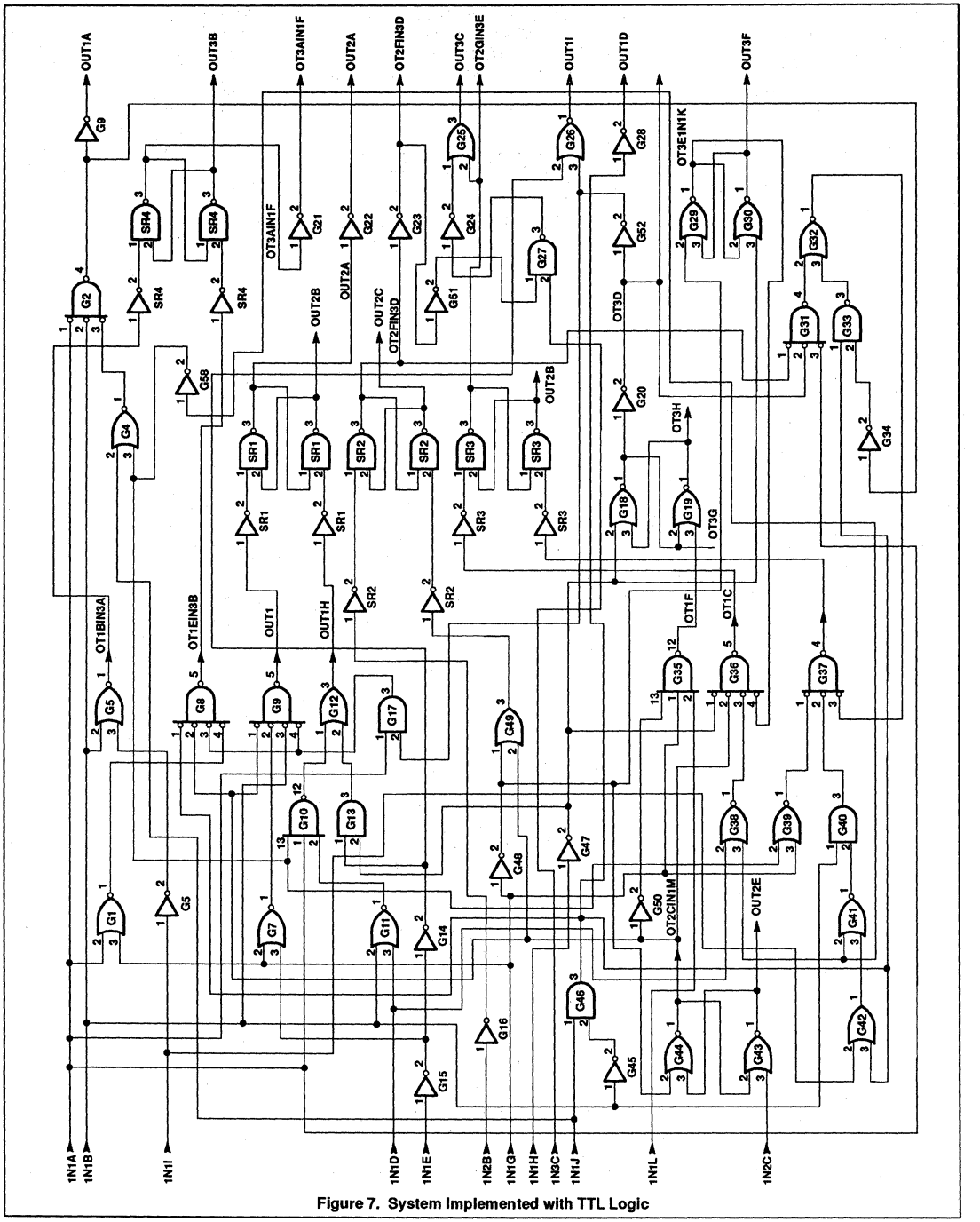


Figure 7. System Implemented with TTL Logic

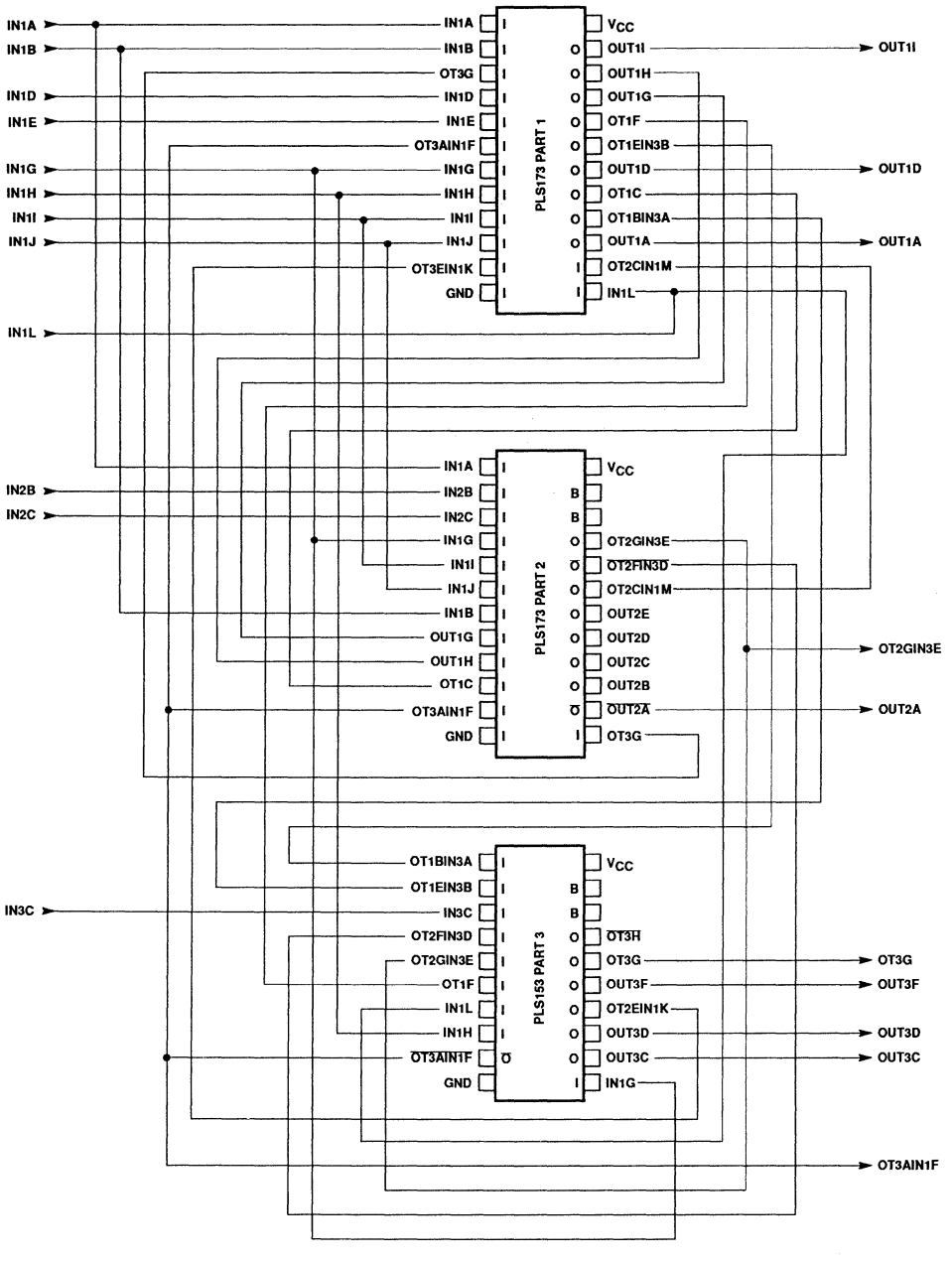


Figure 8. System in Figure 7 Implemented with PLDs

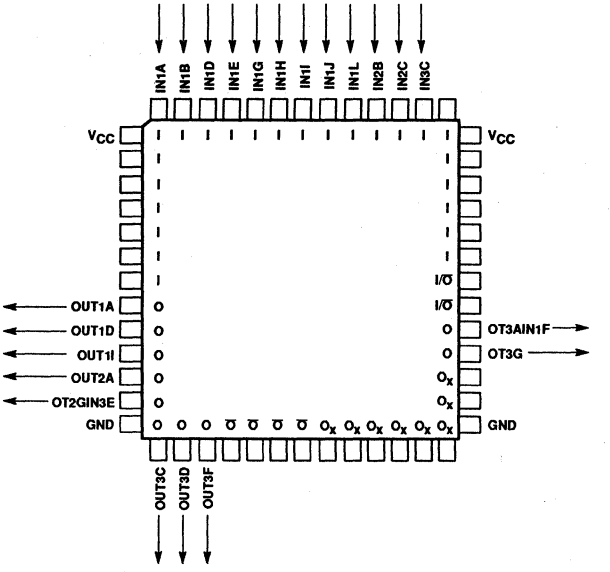


Figure 9. System in Figure 7 Implemented with PLHS501

Programmable macro logic primer AN029

THE PML2552

The PML2552 is the first CMOS PML part. Using an EPROM process, the PML2552 is a dense, fast multilevel PLD capable of making multiple independent state machines in a single part. The PML2552 has been optimized for handling data within the part and is ideal for making bus controllers and other microprocessor peripherals. Useful PML2552 applications areas include DMA controllers, DRAM controllers, Video memory controllers, FAX machines, hand held instruments, laser printers, digital and telecommunications.

The PML2552 combines 96 foldback NAND gates with 52 flip-flops in a 68 pin package. See the block diagram in Figure 10.

Foldback NAND gates are the easiest to use programmable elements. Because any function can be built from NAND gates, they have no inherent logic limitation. Foldback gates permit free connection of any gates in the foldback region. This allows designers to make any number of logic levels necessary to solve problems. Usually, the design software flattens the logic to exploit the wide inputs of these gates, but the designer easily maintains full control.

The choice of flip-flops has been done judiciously optimizing two 8 bit data paths within the part. Two groups of D flip-flops are assigned to input pins, and two groups of D flip-flops are associated with output pins. The latter group may also be buried. Additionally, two groups of 10 JK flip-flops are buried for building efficient counters, shifters and other state machines. JK flip-flops require very little additional gating circuitry to make state machines.

Figure 11 shows a more detailed diagram of the PML2552, with the internal cell names broken out according to the conventions used with Philips Semiconductors SNAP software. An important point should be made with Figure 11 regarding internal timing specifications. Each cell shown in Figure 11 has been specified as if it were a gate array type cell. These specifications include the cell time delay (min,typ,max) and how the time delay increases with incremental loading. The details of the time delays are shown in the data sheet. The exact same values are automatically included in the simulation model which SNAP makes after a design compiles into the PML2552.

THE PML2852

Figure 10 shows a block diagram of the PML2852. The PML2852 is similar to the PML2552 except it has an additional 16 outputs. The 3-State control structure for pins I/O0 – I/O15 was changed from independent control to two 8-bit groups in the PML2852. All chip resources connect through the central array. It is possible to form logic functions of up to 96 levels by passing input signals repetitively through the 96 foldback NAND gates. Folded architectures permit free internal connection of buried gates and flip-flops. It is also possible to form logic functions from input to output in a single NAND level. Two groups of 10 buried JK flip-flops are within the programming array, along with two groups of 8 D flip-flops. Additional input flip-flops are available at the device pins. Several clocking options are available for the different flip-flop groups. The PML2852 has 84 pins. The PML2852 and PML2552 both are available in reprogrammable and one time programmable packages.

Typical propagation delays for the internal NAND gates are 15 nanoseconds. Flip-flop toggle rates are at 50 MHz, with some JK based counters operating at that rate.

The basic architectural rationale is to use the input D flip-flops to capture data. Then, the buried D flip-flops pipeline data and the JK flip-flops form counters, timers and control flip-flops. Of course, the JKs may act as shift registers as can the buried D flip-flops. Using D flip-flops for counter design is discouraged, but allowed. A wide class of data oriented applications fit this architecture well.

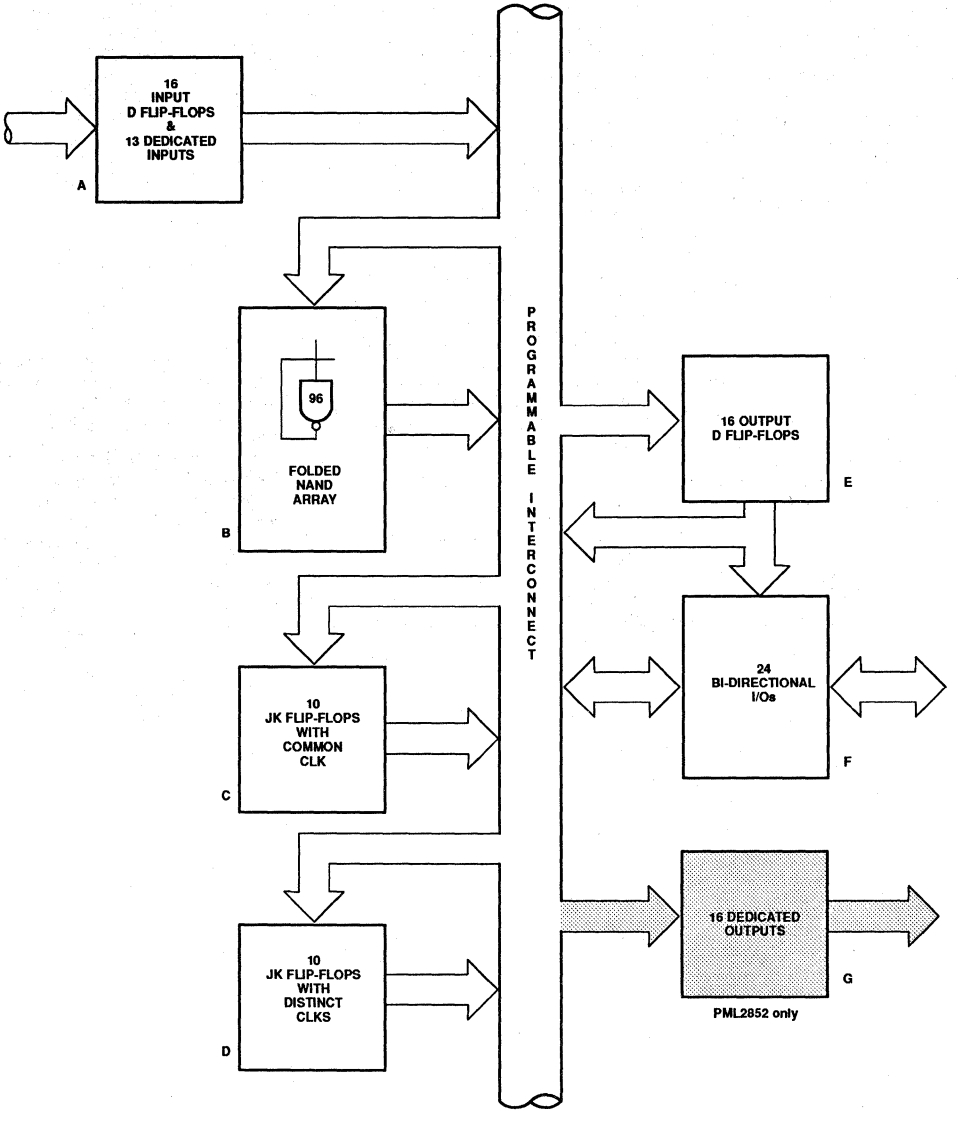


Figure 10. PML2552/PML2852 Functional Block Diagram

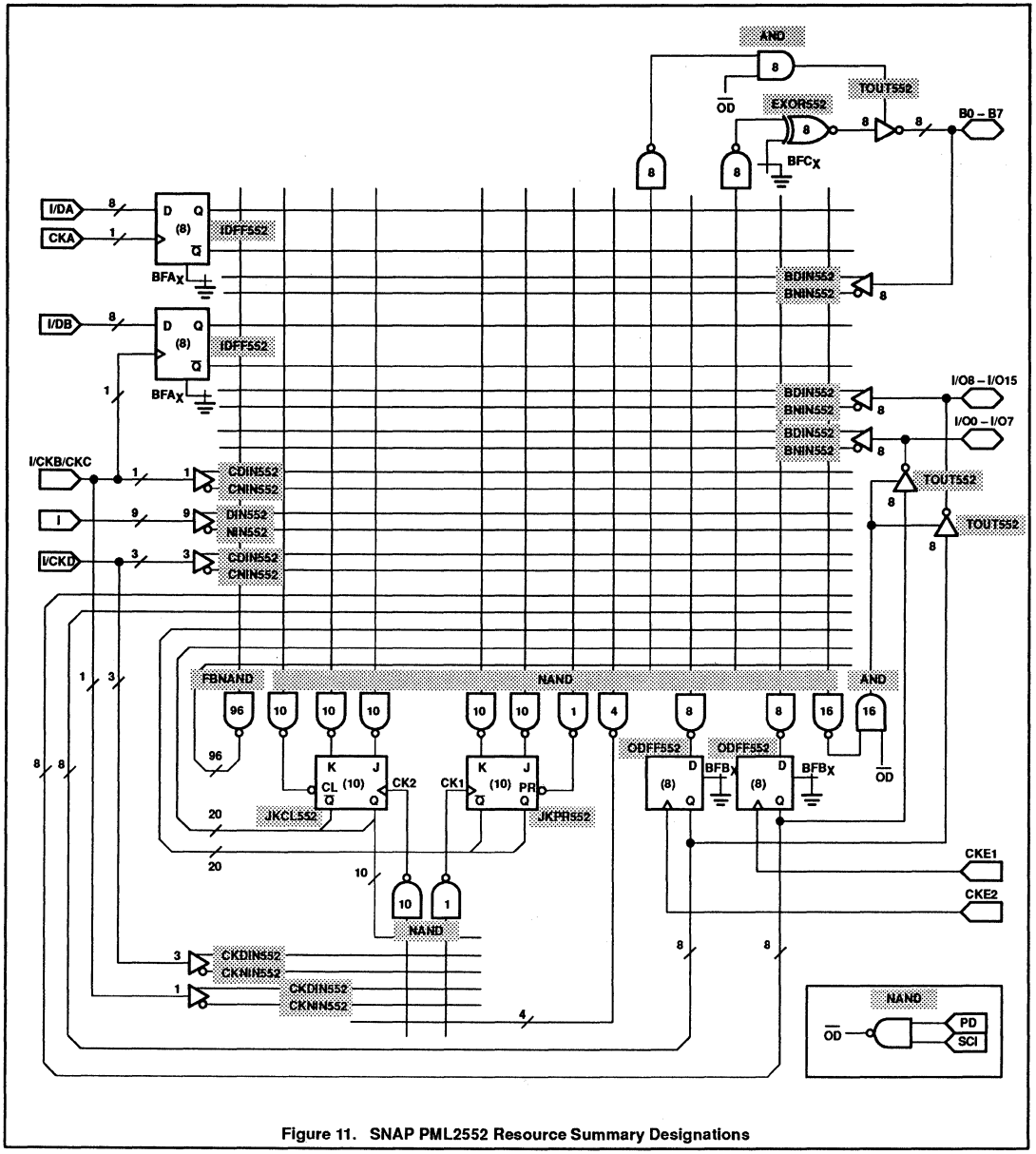


Figure 11. SNAP PML2552 Resource Summary Designations

TESTABILITY

Deep nesting of the buried flip-flops makes testability an important issue. Both controllability and observability of the flip-flops would be jeopardized if they were not scan chain configurable. By asserting the Scan Mode pin, the buried flip-flops form a large, multiplexed shift register (Figure 12). The shifter is controllable and observable. Parallel output of the most significant 16 bits of the shifter goes to the outside world. Alternatively, multiple PML2852s or PML2552s can cascade if correctly connected.

POWER DISSIPATION

All EE and EPROM based PLDs contain an NMOS core. An external CMOS image occurs because the I/O pads are CMOS. To improve on the power consumption, two techniques have evolved. One technique, input transition detection, automatically reduces power consumption by sensing input pin activity. The other technique uses a special power down pin. The power down pin blocks current flow into the chip core, reducing power consumption. The PML2552 and PML2852 use a power down pin. Power down presents the problem of maintaining the internal device state during power transition. Internal node conditions automatically latch when power down occurs, and restore upon power up. Full power consumption is 525 mW at maximum speed. In low power mode, this drops to one tenth that value.

DESIGN SOFTWARE

To support PML, a gate array style software package, called SNAP, is available from Philips. SNAP contains resources for equation and schematic capture (incorporating OrCAD™ SDT and FutureNet™ DASH). SNAP includes a proprietary simulator and logic compilers for the entire Philips Semiconductors PLD family. The compilers use netlist synthesis and netlist optimization methods, and ultimately compile to JEDEC fusemap formats. A logic netlist model, including all internal node delays, is derived automatically from the JEDEC format. The derived model permits accurate device simulation.

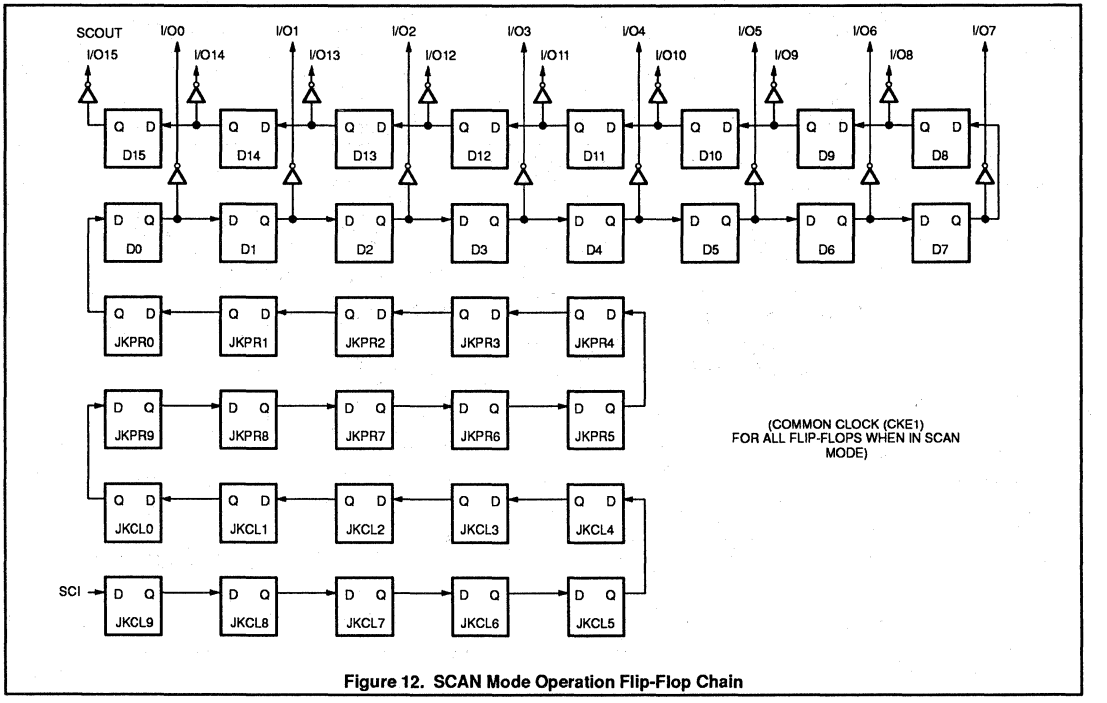


Figure 12. SCAN Mode Operation Flip-Flop Chain

Targeting a PML2552/2852 Design

Key to a successful design is recognizing an appropriate mapping of internal resources to a specific design. The designer should think of JK flip-flops for counters and state machines, maximizing efficiency. As well, the input D flip-flops should be aimed at handling byte wide data applied to the input pins. The buried D flip-flops can be thought of as a natural landing place for output data, internally buffered data, shift registers, or simple state machines. Care should be taken to assign external clocks to the appropriate pins of the PML2552 or PML2852, because specific pins are dedicated as clock inputs to the internal flip-flops. SNAP may sometimes be used with automatic pinning mode, but this is often less than optimal.

By careful use of key flip-flop properties, a design can be easily forced to land in specific internal flip-flops. Remember, the PML2552/2852 has input D flip-flops, output D flip-flops, JK flip-flops with independent resets and JK flip-flops with independent sets. As an example, if a JK flip-flop equation is written, and its asynchronous set input is to a logical 1, then SNAP assumes that the JK flip-flop is to be one which has an asynchronous set and assigns it accordingly. On the other hand, if the asynchronous reset is referred to, SNAP will map this to a JK flip-flop with an asynchronous reset input. It's usually that simple. If the designer uses a flip-flop which is not an available PML resource, SNAP will build it out of NAND gates in the foldback array. This is usually found during the compile phase when the resource summary exceeds the available foldback gates and the expected flip-flops go unused.

The most straightforward design method is simple incremental compiling. Incremental compiling is the logical approach to use with PML. Incremental compiling is simply taking a small piece of a design, capturing it (either equations or schematic) and compiling it into the part. During the compilation process, a small table is displayed showing the current usage of internal PML 2552 resources. Figure 13 shows such a small resource table. At this point, when the resources are examined, each cell should be accounted for. If not, there is either something wrong or the designer has misunderstood the part or the mapping process. When the resource is correctly mapped, it shows up in the table with the right tally. If not, the netlist may need examination.

By going incrementally, designs will progress systematically with only small surprises along the way. Usually, the surprises are that the design took fewer resources than expected. This is because the PML architecture is almost fully connectible, and SNAP optimizer is focused on flattening the design (for speed) to maximize wide internal gate usage.

As a suggestion, one very appropriate design method is to layout the data paths first, compile the design, then when everything fits appropriately, design in the control sections.

REFERENCES

1. Cavlan, Napoleone 1985. "Third Generation PLD Architecture Breaks AND-OR Bottleneck", WESCON 1985 Conference Proceedings.
2. Wong, David K. "Third Generation PLD Architecture and its Applications", Electro 1986 Conference Proceedings.

```
Design from PML2552B.N2 - Created on Fri Jun 22, 1990  3:19PM
Device PML2552
   Cell name   used/total     %
=========================================
     CKDIN552     0 /   4    0%
     CKNIN552     0 /   4    0%
       FBNAND    59 /  96   61%
          NAND   96 / 104   92%
      DIN552     11 /  25   44%
      NIN552     11 /  25   44%
     CDIN552      3 /   4   75%
     CNIN552      1 /   4   25%
       CK552      1 /   4   25%
     IDFF552      8 /  16   50%
     BDIN552      0 /  24    0%
     JKCL552     10 /  10  100%
     JKPR552     10 /  10  100%
     EXOR552      8 /   8  100%
     TOUT552     20 /  24   83%
     ODFF552     12 /  16   83%
```

Figure 13. PML2552 Resources Summary

PLHS501 design examples

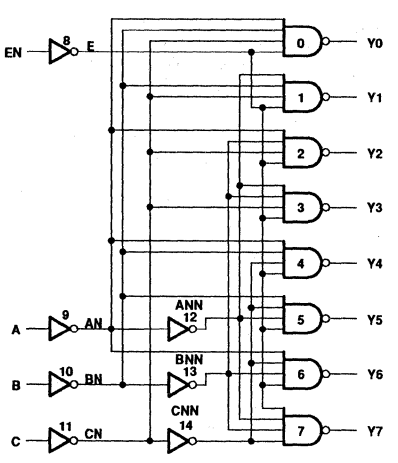
AN049

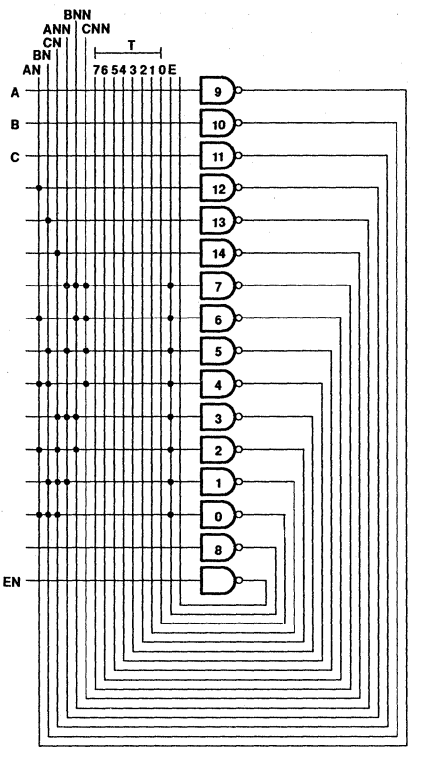
DESIGN EXAMPLES

Most designers tend to view a PLD as a mechanism for collecting logical glue within a system. That is, those pieces which tie together the larger LSI microprocessors, controllers, RAMs, ROMs, UARTs, etc. However, there is a tendency of viewing a gate array as an entire system on a chip. PML based products will fit well in either casting as will be demonstrated by a series of small but straightforward examples. For starters, we shall examine how the fusing process embeds function, progress to glue-like decoding operations and finally demonstrate some coprocessor like functions as well as homemade "standard products".

The method of associating gates within the NAND foldback structure is depicted in Figure 1 wherein a simple three to eight decoder is fused into the array. The corresponding inputs are on the left and outputs at the top. This figure shows inputs and their inverse formed in the array resulting in a solution that requires 6 inverting NANDs that would probably be best generated at the input receivers. Hence, this diagram could be trimmed by six gates, down to eight to achieve the function. Figure 2 shows two consecutive D flip-flop fusing images. Note that asynchronous sets and resets may be achieved for free, in this version. In both Figures 1 and 2 the gates are numbered in a one-to-one arrangement. As well, the accompanying equations are in the format used by Philips SNAP design software. For clarity, consider the gate labeled 2A in Figure 1. Schematically, this is shown as a 3 input NAND. However, in the fused depiction, it combines from three intermediate output points with the dot intersect designation. Hence, all gates are drawn as single input NANDs whose inputs span the complete NAND gate foldback structure.

1 OF 8 DECODER/DEMULITPLEXER






```
@LOGIC EQUATION
AN = /A;
BN = /B;
CN = /C;
ANN = /AN;
BNN = /BN;
CNN = /CN;
E = /EN;
Y0 = / (AN * BN * CN * E);
Y1 = / ANN * BN * CN * E);
Y2 = / AN * BNN * CN * E);
Y3 = / ANN * BNN * CN * E);
Y4 = / AN * BN * CNN * E);
Y5 = / ANN * BN * CNN * E);
Y6 = / AN * BNN * CNN * E);
Y7 = / ANN * BNN * CNN * E);
```

Figure 1. Decoder Implementation in NAND Foldback Structure

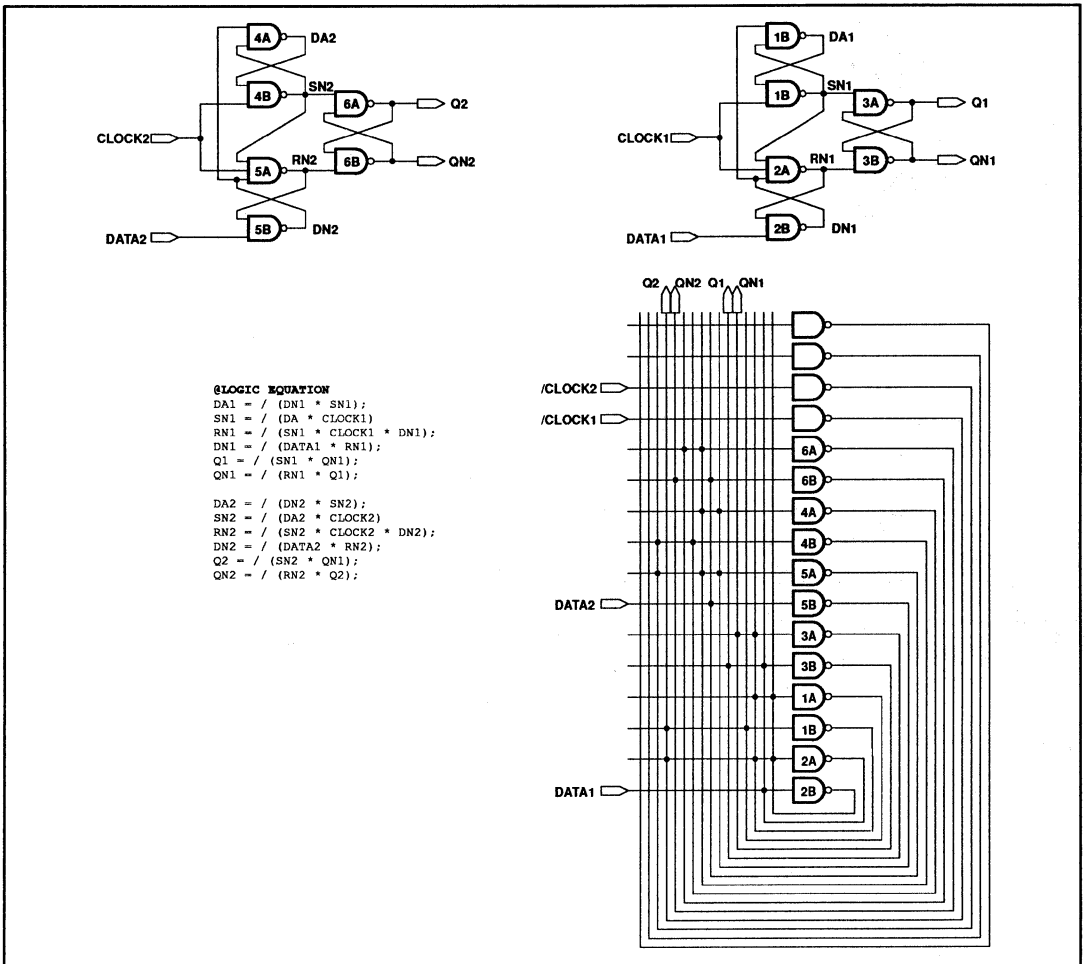


Figure 2. Two Flip-Flops Implemented in the NAND Foldback Strucutre

One straightforward example of using a PLHS501 is shown in Figure 3. Here, the device is configured to accept the 23 upper address lines generated by a 68000 microprocessor. By selecting the direct and complemented variables, at least 16 distinct address selections can be made using only the dedicated outputs. The designer can combine additional VME bus strobes, or other control signals to qualify the decode or, define 8 additional outputs for expanded selection.

As well, the designer could transform the bidirectionals to inputs and decode over a 32 bit space, selecting combinations off of a 32 bit wide address bus. Because this simple level of design requires only NAND output terms plus 4 NAND gates in the foldback array (for inversion of signals connected to O3.O0), there may be as many as 68 remaining gates to accomplish additional handshaking or logical operations on the input variables.

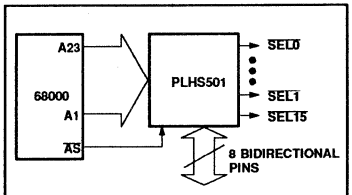


Figure 3. 68000 Microprocessor Address Decode

PLHS501 design examples AN049

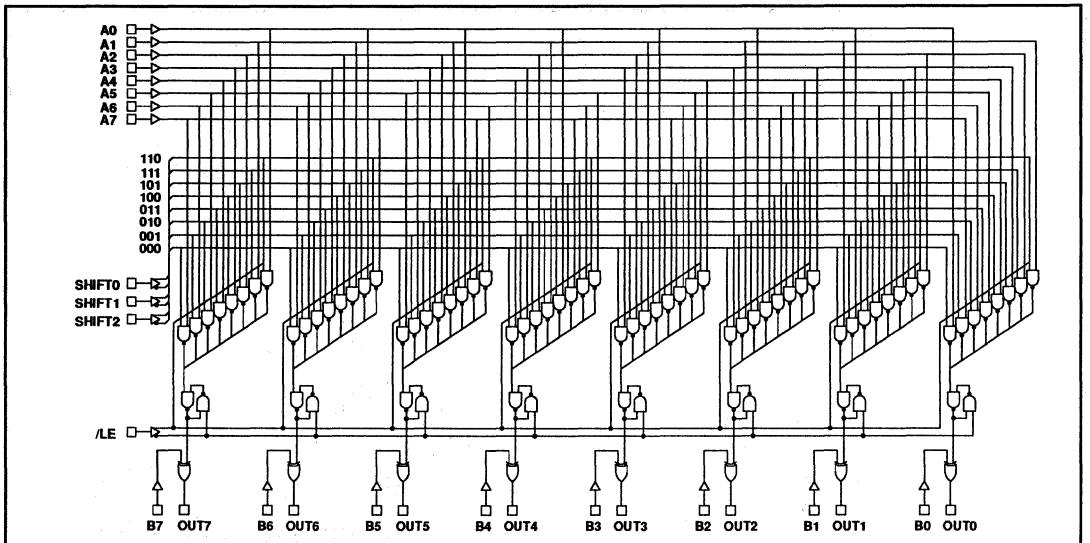


Figure 4. 8-Bit Barrel Shifter Implemented with the PLHS501

An eight bit barrel shifter exploits most of the PLHS501 as depicted in Figure 4. This implementation utilizes all 72 internal foldback NANDs in a relatively brute force configuration as well as 8 output NANDs to generate transparent latched and shifted results. The shift position here is generated by the shift 0, shift 1 and shift 2 inputs which are distinguished and selected from the input cells. Variations on this idea of data manipulation could include direct passing data, mirror imaged data (bit reversal) or byte swapping to name a few.

Part of an eight bit, look-ahead parallel adder is shown in Figure 5. Gates necessary to form the level-0 generate and propagate, as well as the XOR output gates generating the resulting sum are not shown. The reader should be aware that this solution exploits four layers of pyramided gates and only utilizes a total of about 58 gates. Additional comparison or Boolean operations could still be generated with remaining NAND functions to achieve additional arithmetic operations. This application should make the reader aware of a new class of applications achievable with third generation PLDs - user definable I/O coprocessors. The approach of increasing microprocessor performance by designing dedicated task coprocessors is now within the grasp of user definable single chip solutions.

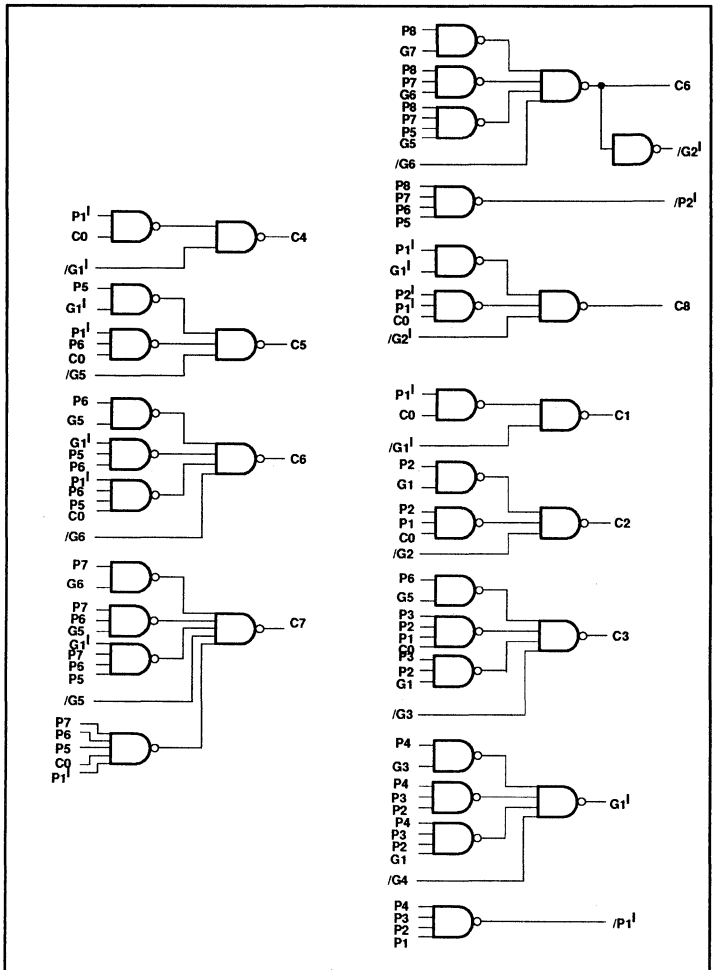


Figure 5. Partial NAND Gate Equivalence of the 8-Bit Look-Ahead Adder

An example of one of the least efficient structures realizable on the PLHS501 is shown in Figure 6. Here, a cascade of 12 flip-flops are formed into a toggle chain that used all available NAND gates in the main logic array. In the PLHS501 simple cross-coupled latches or transparent D latches are preferred over edge triggered flip-flops simply because they conserve NAND gates. Applications for structures like this include timing generators, rate multiplication, etc. Rearranging Figure 6 as a 12-bit shifter, picking off states at the output terms could result in a general purpose sequence recognizer capable of recognizing binary string sequences. These strings could be up to 13 bits long (in a Mealy configuration) and 24 distinct sequences could be sensed and detected.

Figure 7 shows a 32 to 5-bit priority encoder. This sort of device could generate encoded vector interrupts for 32 contending devices. Of particular interest is the fact that ordinary encoders are not this wide. The designer is, of course, not constrained to generating combinational functions in even powers of two. Thus, the PLHS501 can easily perform customized functions like a 5 to 27 decoder or a 14 to 4 encoder or, even an 18 to 7 multiplexor. For the sake of optimization, the designer is encouraged to implement precisely the function he needs, no more and no less!

The design examples given are illustrative of some typical operations used in ordinary systems. In each case, the example could be thought of as simply an "off the shelf" standard solution to an every day problem (i.e., a de facto standard product).

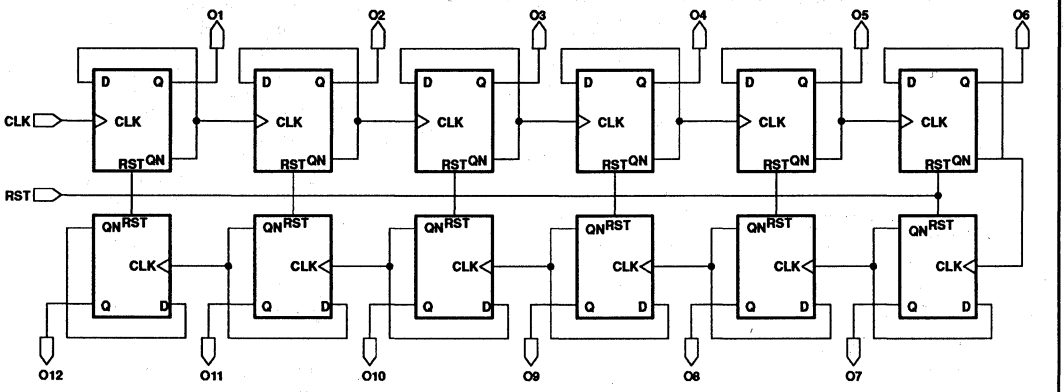


Figure 6. 12-Bit Ripple Counter

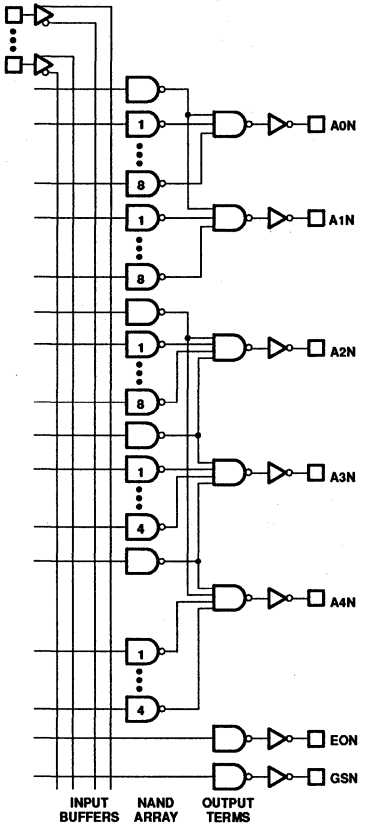


Figure 7. Encoder

DEVELOPMENT SUPPORT

SNAP

Because the architecture encourages deep functional nesting, a new support tool has been developed. Synthesis, Netlist, Analysis and Program (SNAP) software defines a gate array type development environment. SNAP permits several forms of design capture (schematic, Boolean equations, state equations, etc.), a gate array simulator with back annotation, waveform display and a complete fault analyzer and final fusemap compilation and model extraction. SNAP comes with a library of cells, and designs may be captured independently of the ultimate device that will implement the design. This permits the designer to migrate his design among a family of PML devices just as gate array designs can be moved to larger foundations when they do not route on smaller ones. Figure 8 shows the SNAP user interface "Shell" which dictates one sequence of operations to complete a design. Other sequences may be used.

The top portion of the shell depicts the paths available for design entry. Any design may be implemented in any one or a blend of all methods. For instance, a shift register might best be described schematically but a decoder by logic equations. These may be united with a multiplexor described by a text netlist as well. Ultimately, each form of input will be transformed to a function netlist and passed either to the simulation section or to the compiler section. Waveform entry is for simulation stimuli.

The simulator portion of SNAP is a 5-State gate array simulator with full timing information, setup and hold time checking, toggle and fault grade analysis and the ability to display in a wide range of formats, any set of nodes within the design. This permits a designer to zoom in with a synthetic logic state analyzer and view the behavior of any point in the design. Simulations can occur with unit delays, estimations or exact delays. The sequence of operations depicted in Figure 8 is entirely arbitrary, as many other paths exist.

It should be noted that the output of the "merger" block represents the composite design, but as yet is not associated to a PML device. This occurs in the compiler portion wherein association to the device occurs and a fusemap is compiled. This is analogous to placement and routing in a gate array environment. Because of the inter-connectibility of PML, this is not difficult. Once compiled, the exact assignment of pins, gates and flip-flops is known, so timing parameters may be associated and a new simulation model generated with exact detailed timing embedded. The design may be simulated very accurately at this point, and if correct, a part should be programmed.

To facilitate future migration to workstations, SNAP has been written largely in C. The internal design representation is EDIF (Electronic Design Interchange Format) compatible which permits straightforward porting to many commercially viable environments. SNAP currently utilizes OrCAD for schematic entry with eminent availability of FutureNet™ DASH.

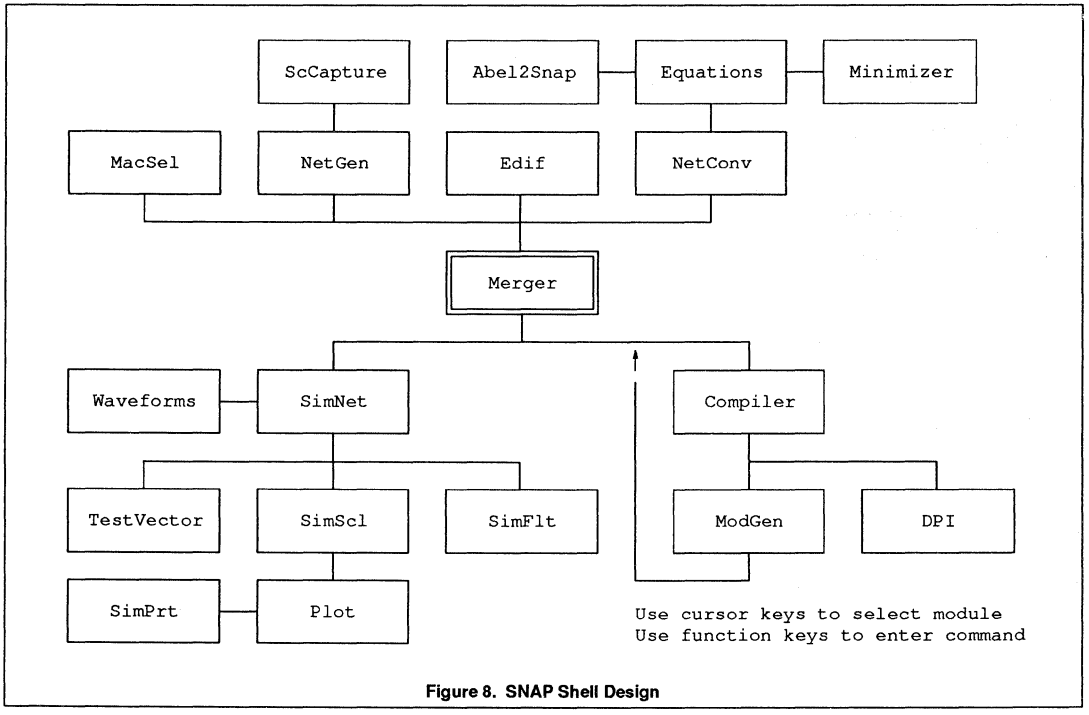


Figure 8. SNAP Shell Design

FutureNet and DASH are trademarks of DATA I/O

PLHS501 design examples AN049

PLHS501 EXAMPLES USING SNAP

- 8-bit barrel shifter
- 12-bit comparator with dual 1 of 8 decoders
- 8-bit carry look-ahead adder
- 32 to 5 priority encoder
- 4-bit synchronous counter
- VME EPROM interface
- Microchannel interface
- NuBus interface
- Data bus parity generator
- 16-bit comparator

Following are example applications for the PLHS501 using SNAP. They should not be viewed as showing all possible capabilities of the device. They have been designed to demonstrate some of the PLHS501 features, syntax of SNAP, and to give the reader some ideas for possible circuit implementations.

Note that these examples were written using SNAP Rev. 1.90. Although Philips will try to keep succeeding versions of SNAP compatible, it may be necessary to change some syntax rules. Therefore, please refer to your SNAP manual for any notes on differences, if using a revision later than Rev. 1.90.

8-BIT BARREL SHIFTER

This 8-bit shifter will shift to the right, data applied to A7 – A9 with the result appearing on OUT7 – OUT0. Data may be shifted by 1 to 7 places by indicating the desired binary count on pins SHIFT2 – SHIFT0. Data applied to the OUT0 position for a shift of 1. For a shift of 0, A7 will appear on OUT7.

Also included is a transparent latch for the output bits. The input 'COMPLMTO' will invert all output bits simultaneously and input /OE will 3-State all outputs.

This design was done by using OrCAD's SDT with SNAP. The top level drawing is shown in Figure 11. The PLHS501 has various output structures. For the best fit, it was necessary to alter the portion of the schematic connecting to pins 15 – 18 compared to pins 37–40. This is shown in Figures 12 and 13.

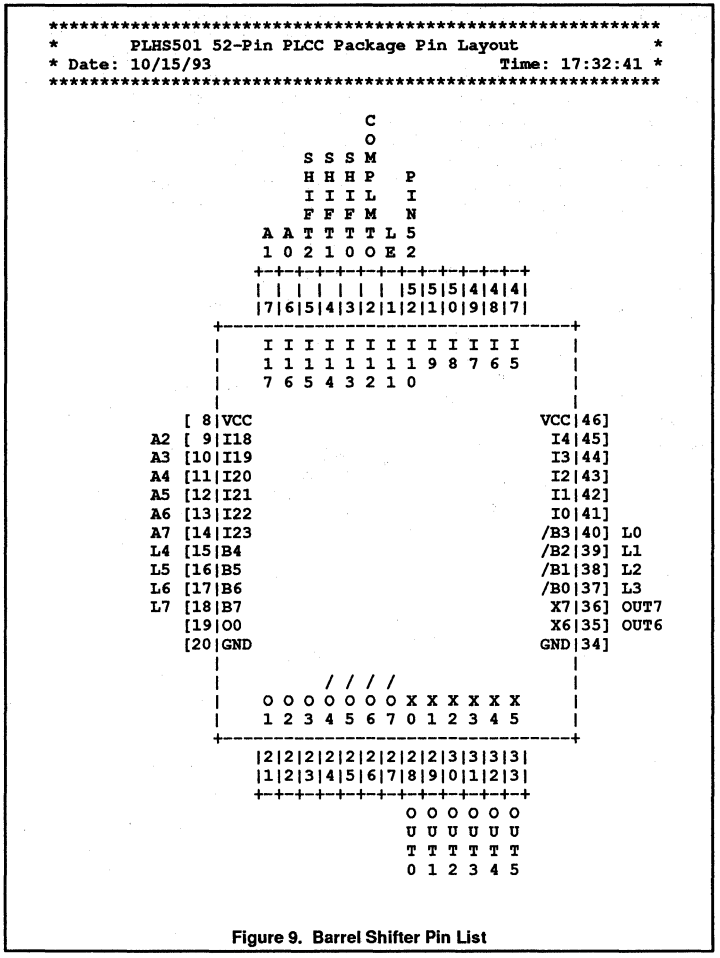


Figure 9. Barrel Shifter Pin List

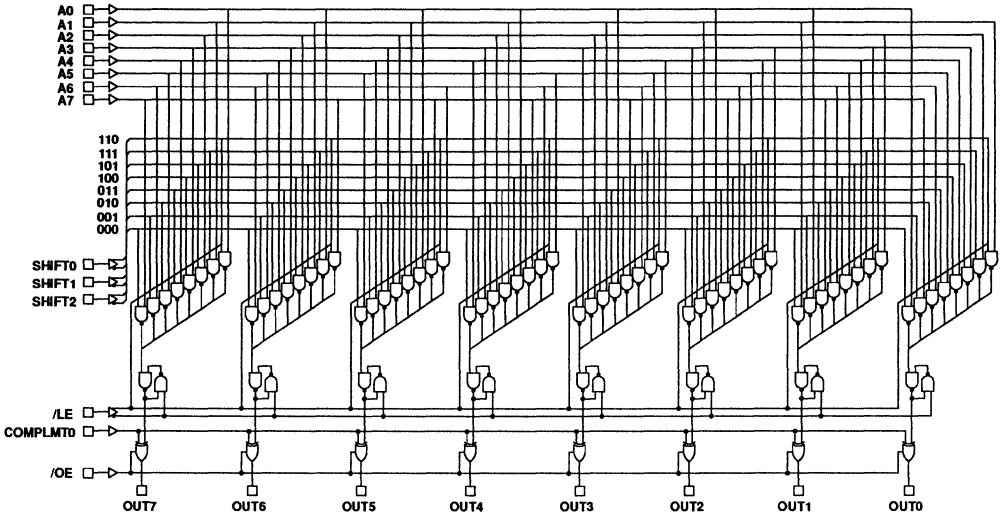


Figure 10. 8-Bit Barrel Shifter Schematic

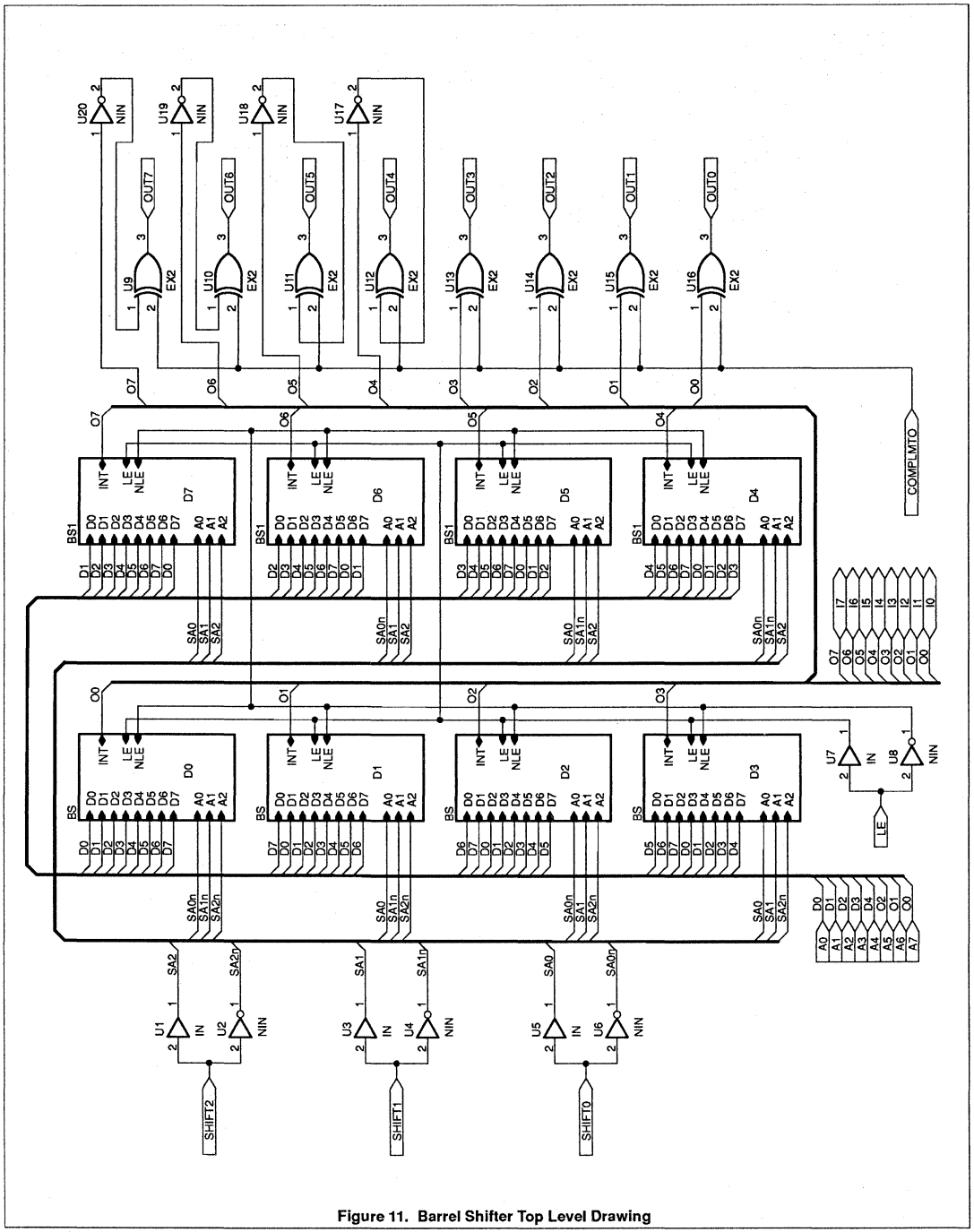


Figure 11. Barrel Shifter Top Level Drawing

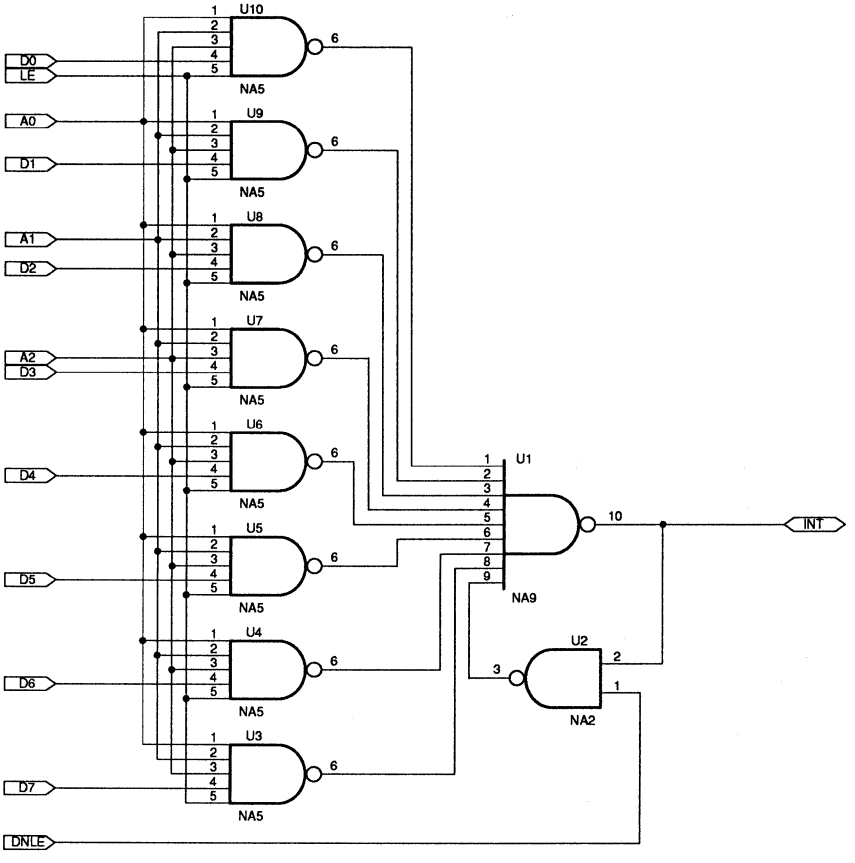


Figure 12. Portion of Shifter to Connect to NAND Output Pins

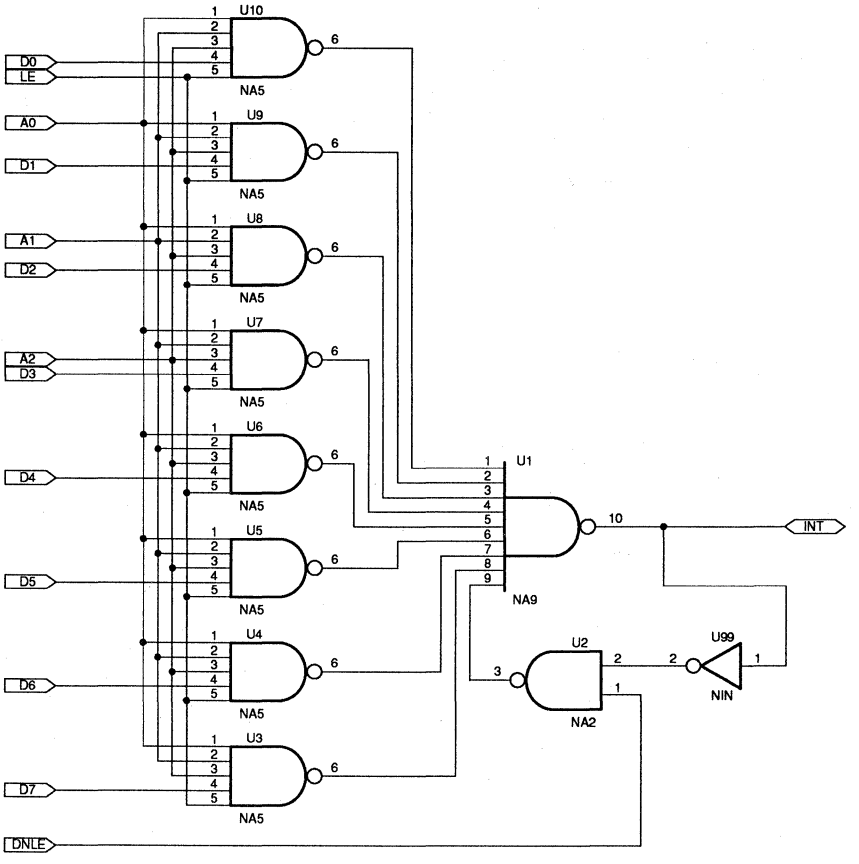


Figure 13. Portion of Shifter to Connect to AND Output Pins

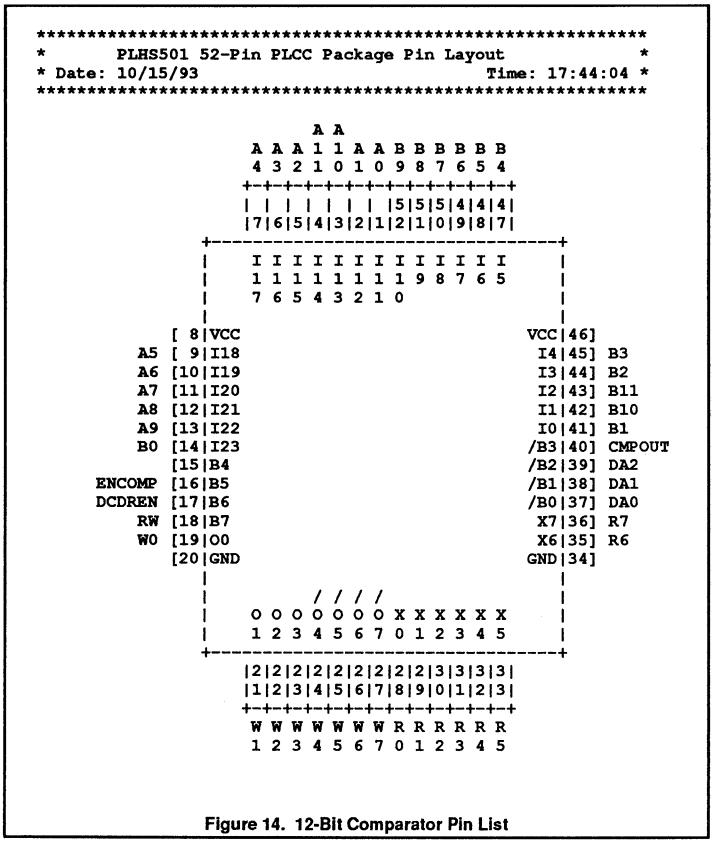


Figure 14. 12-Bit Comparator Pin List

12-BIT COMPARATOR WITH DUAL 1-OF-8 DECODERS

Two functions that are very often associated with controlling I/O parts are address comparison and address decoding. In this example, both functions are programmed into a PLHS501 using 52 out of the 72 foldback NAND terms.

The comparator compares 12 bits on inputs A11 – A0 to inputs B11 – B0 when the input 'ENCMP' is High. Output 'CMPOUT' will become Active-Low when all 12 bits of the A input match the B. Selection between the two decoders is done with input 'R/W'. Only one output may be active (Low) at a time. Although currently separate functions, the decoder enable may be derived internally from 'CMPOUT' freeing 2 bidirectional pins which together with available foldback NAND terms, may be used to incorporate a third function.

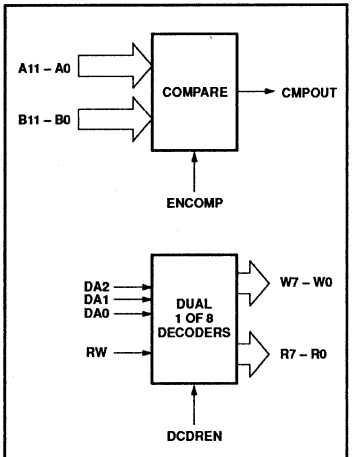


Figure 15. 12-Bit Comparator with Dual 1 – 8 Decoders Block Diagram

```
"FILENAME: CMP12BIT.EQN"
"          12-bit address comparator and dual 1 of 8 decoders"
@PINLIST
B0      I;
B1      I;
B2      I;
B3      I;
B4      I;
B5      I;
B6      I;
B7      I;
B8      I;
B9      I;
B10     I;
B11     I;
A0      I;
A1      I;
A2      I;
A3      I;
A4      I;
A5      I;
A6      I;
A7      I;
A8      I;
A9      I;
A10     I;
A11     I;
DA0     I;
DA1     I;
DA2     I;
RW      I;
DCDREN  I;
ENCOMP  I;
W0      O;
W1      O;
W2      O;
W3      O;
W4      O;
W5      O;
W6      O;
W7      O;
R0      O;
R1      O;
R2      O;
R3      O;
R4      O;
R5      O;
R6      O;
R7      O;
CMPOUT  O;

@LOGIC EQUATION

"COMMON PRODUCT TERM"
ad0=/da2*/da1*/da0*dcdren;
ad1=/da2*/da1* da0*dcdren;
ad2=/da2* da1*/da0*dcdren;
ad3=/da2* da1* da0*dcdren;
ad4= da2*/da1*/da0*dcdren;
ad5= da2*/da1* da0*dcdren;
ad6= da2* da1*/da0*dcdren;
ad7= da2* da1* da0*dcdren;
```

Figure 16. 12-Bit Comparator Boolean Equations (1 of 2)

```
"12-Bit Address Comparator"

axb0 = a0*/b0 + /a0*b0;
axb1 = a1*/b1 + /a1*b1;
axb2 = a2*/b2 + /a2*b2;
axb3 = a3*/b3 + /a3*b3;
axb4 = a4*/b4 + /a4*b4;
axb5 = a5*/b5 + /a5*b5;
axb6 = a6*/b6 + /a6*b6;
axb7 = a7*/b7 + /a7*b7;
axb8 = a8*/b8 + /a8*b8;
axb9 = a9*/b9 + /a9*b9;
axb10 = a10*/b10 + /a10*b10;
axb11 = a11*/b11 + /a11*b11;

cmpout = /(/axb0*/axb1*/axb2*/axb3*/axb4*/axb5*/axb6*/axb7*/axb8*/axb9*
          /axb10*/axb11*encomp);

"Dual 1 of 8 decoders

-  da2-da0 are address inputs
-  dcdren is an enable input
-  rw selects which group of 8 outputs r7-r0 or w7-w0
      will have the decoded active low output"

w7 = /(ad7*/rw);
w6 = /(ad6*/rw);
w5 = /(ad5*/rw);
w4 = /(ad4*/rw);
w3 = /(ad3*/rw);
w2 = /(ad2*/rw);
w1 = /(ad1*/rw);
w0 = /(ad0*/rw);

r7 = /(ad7* rw);
r6 = /(ad6* rw);
r5 = /(ad5* rw);
r4 = /(ad4* rw);
r3 = /(ad3* rw);
r2 = /(ad2* rw);
r1 = /(ad1* rw);
r0 = /(ad0* rw);
```

Figure 16. 12-Bit Comparator Boolean Equations (2 of 2)

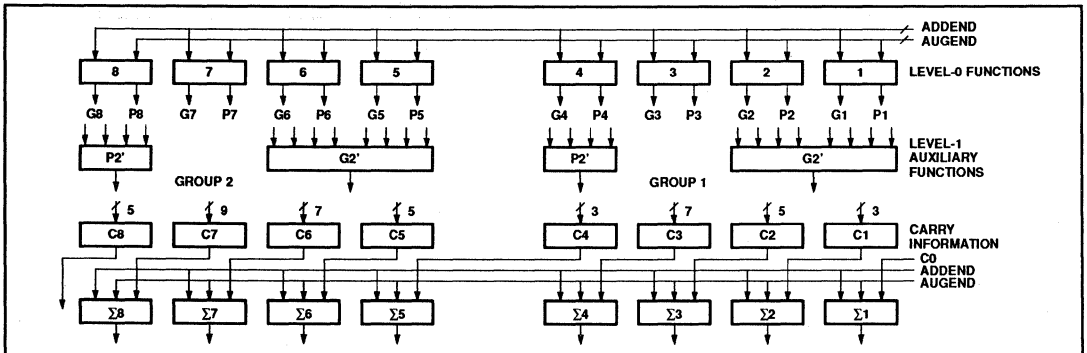
8-BIT CARRY LOOK-AHEAD ADDER

This function may be used as part of an ALU design or simply to off-load a microprocessor. Figure 18 is a block diagram showing the individual components needed for each bit.

A carry input (C0) is provided along with a carry output (C8). The result of an addition between the inputs A7 – A0 and B7 – B0 occurs on outputs SUM7 – SUM0.

```
**********************************************************
*       PLHS501 52-Pin PLCC Package Pin Layout           *
* Date: 10/15/93                          Time: 17:50:08 *
**********************************************************

                    A A A A A A A
                    6 5 4 3 2 1 0
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+
                   | | | | | | | |5|5|5|4|4|4|
                   |7|6|5|4|3|2|1|2|1|0|9|8|7|
               +-----------------------------------+
               |    I I I I I I I I I I I I I      |
               |    1 1 1 1 1 1 1 1 9 8 7 6 5      |
               |    7 6 5 4 3 2 1 0                |
               |                                   |
          [ 8|VCC                            VCC|46]
       A7 [ 9|I18                             I4|45]
       B0 [10|I19                             I3|44] C0
       B1 [11|I20                             I2|43] B7
       B2 [12|I21                             I1|42] B6
       B3 [13|I22                             I0|41] B5
       B4 [14|I23                            /B3|40]
          [15|B4                             /B2|39]
          [16|B5                             /B1|38]
          [17|B6                             /B0|37]
          [18|B7                              X7|36] SUM7
       C8 [19|O0                              X6|35] SUM6
          [20|GND                            GND|34]
               |                                   |
               |          / / / /                  |
               |    O O O O O O O O X X X X X X    |
               |    1 2 3 4 5 6 7 0 1 2 3 4 5      |
               +-----------------------------------+
                   |2|2|2|2|2|2|2|2|2|2|3|3|3|3|
                   |1|2|3|4|5|6|7|8|9|0|1|2|3|
                   +-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 S S S S S S
                                 U U U U U U
                                 M M M M M M
                                 0 1 2 3 4 5
```

Figure 17. 8-Bit Adder Pin List



NOTES:

G1' = G4 + P4 * G3 + P4 * P3 * G2 + P4 * P3 * P2 * G1;
P1' = P4 * P3 * P2 * P1
G2' = G8 + P8 * G7 + P8 * P7 * G6 + P8 * P7 * P6 * G5;
P2' = P8 * P7 * P6 * P5

C1 = G1 + P1 * C0;
C2 = G2 + P2 * G1 + P2 * P1 * C0;
C3 = G3 + P3 * G2 + P3 * P2 * G1 + P3 * P2 * P1 * C0;
C4 = G1' + P1' * C0;
C5 = G5 + P5 * G1' +P5 * P1' * C0;
C6 = G6 + P6 * G5 + P6 * P5 * G1' + P6 * P5 * P1' * C0;
C7 = G7 + P7 * G6 + P7 * P6 * G5 + P7 * P6 * P5 * G1' +P7 * P6 * P5 * P1' * C0;
C8 = G2' + P2' * G1' + P2' * P1' * C0;

Figure 18. 8-Bit Carry Look-Ahead Adder Block Diagram and Equations

```
"FILENAME: ADDR8BIT.EQN
           8 Bit Carry Look-Ahead Adder"

@PINLIST
A0      I;
A1      I;
A2      I;
A3      I;
A4      I;
A5      I;
A6      I;
A7      I;
B0      I;
B1      I;
B2      I;
B3      I;
B4      I;
B5      I;
B6      I;
B7      I;
C0      I;

C8      O;
SUM0    O;
SUM1    O;
SUM2    O;
SUM3    O;
SUM4    O;
SUM5    O;
SUM6    O;
SUM7    O;

@LOGIC EQUATION

"level-0 functions"

gn1 = /(a0*b0);
p1 = /(/a0*/b0);
g1 = /gn1;

gn2 = /(a1*b1);
p2 = /(/a1*/b1);
g2 = /gn2;

gn3 = /(a2*b2);
p3 = /(/a2*/b2);
g3 = /gn3;

gn4 = /(a3*b3);
p4 = /(/a3*/b3);
g4 = /gn4;

gn5 = /(a4*b4);
p5 = /(/a4*/b4);
g5 = /gn5;

gn6 = /(a5*b5);
p6 = /(/a5*/b5);
g6 = /gn6;

gn7 = /(a6*b6);
p7 = /(/a6*/b6);
g7 = /gn7;

gn8 = /(a7*b7);
p8 = /(/a7*/b7);
g8 = /gn8;

"level-1 functions"

g1_1 = g4 + p4*g3 + p4*p3*g2 + p4*p3*p2*g1;
g2_1 = g8 + p8*g7 + p8*p7*g6 + p8*p7*p6*g5;
```

Figure 19. 8-Bit Adder Boolean Equations (1 of 2)

```
"carry information"

c1 = g1   + p1*c0;
c2 = g2   + p2*g1   + p2*p1*c0;
c3 = g3   + p3*g2   + p3*p2*g1   + p3*p2*p1*c0;
c4 = g1_1 + p4*p3*p2*p1*c0;
c5 = g5   + p5*g1_1 + p5*p4*p3*p2*p1*c0;
c6 = g6   + p6*g5   + p6*p5*g1_1 + p6*p5*p4*p3*p2*p1*c0;
c7 = g7   + p7*g6   + p7*p6*g5   + p7*p6*p5*g1_1 + p7*p6*p5*p4*p3*p2*p1*c0;
c8 = g2_1 + p8*p7*p6*p5*g1_1     + p8*p7*p6*p5*p4*p3*p2*p1*c0;

"addition functions"

sum0 = c0 :+: (p1 * gn1);
sum1 = c1 :+: (p2 * gn2);
sum2 = c2 :+: (p3 * gn3);
sum3 = c3 :+: (p4 * gn4);
sum4 = c4 :+: (p5 * gn5);
sum5 = c5 :+: (p6 * gn6);
sum6 = c6 :+: (p7 * gn7);
sum7 = c7 :+: (p8 * gn8);
```

Figure 19. 8-Bit Adder Boolean Equations (2 of 2)

```
************************************************************
*       PLHS501 52-Pin PLCC Package Pin Layout             *
* Date: 10/15/93                          Time: 17:58:06 *
************************************************************

                  I I I I I I   I   I I I I
                  1 1 1 1 1 1 3 I 3 I 2 2 2 2
                  5 4 3 2 1 1 0 0 2 9 8 7 6
                 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                 | | | | | | | |5|5|5|4|4|4|
                 |7|6|5|4|3|2|1|2|1|0|9|8|7|
            +-----------------------------------+
            |    I I I I I I I I I I I I I     |
            |    1 1 1 1 1 1 1 1 9 8 7 6 5     |
            |    7 6 5 4 3 2 1 0               |
            |                                  |
          [ 8|VCC                          VCC|46]
      I16 [ 9|I18                           I4|45] I25
      I17 [10|I19                           I3|44] I24
      I18 [11|I20                           I2|43] I23
      I19 [12|I21                           I1|42] I22
       I1 [13|I22                           I0|41] I21
      I20 [14|I23                          /B3|40] I10
       I3 [15|B4                           /B2|39] I9
       I4 [16|B5                           /B1|38] I8
       I5 [17|B6                           /B0|37] I7
       I6 [18|B7                            X7|36]
       A0 [19|O0                            X6|35]
          [20|GND                          GND|34]
            |                                  |
            |          / / / /                 |
            |    O O O O O O O X X X X X X     |
            |    1 2 3 4 5 6 7 0 1 2 3 4 5     |
            +-----------------------------------+
                 |2|2|2|2|2|2|2|2|2|3|3|3|3|
                 |1|2|3|4|5|6|7|8|9|0|1|2|3|
                 +-+-+-+-+-+-+-+-+-+-+-+-+-+
                  A A A A       E G
                  1 2 3 4       O S
```

Figure 20. Encoder Pin List

32- to 5-BIT PRIORITY ENCODER

This relatively simple example demonstrates the capability of the PLHS501 to be programmed with functions that are not available in 'standard' device libraries. The equations may look difficult at first glance. However, there is a pattern to the encoding. Referring to Figure 21, Lab4 – Lab1 are terms that are common to several outputs (A4n – A0n). Separating them from the main equations allows a total reduction in the numbers of gates used.

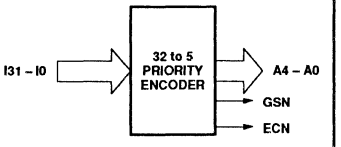


Figure 21. 32 to 5 Priority Encoder Block Diagram

```
"FILENAME: ENCODER.EQN
           32 TO 5 PRIORITY ENCODER"
@PINLIST
I0        I;
I1        I;
I2        I;
I3        I;
I4        I;
I5        I;
I6        I;
I7        I;
I8        I;
I9        I;
I10       I;
I11       I;
I12       I;
I13       I;
I14       I;
I15       I;
I16       I;
I17       I;
I18       I;
I19       I;
I20       I;
I21       I;
I22       I;
I23       I;
I24       I;
I25       I;
I26       I;
I27       I;
I28       I;
I29       I;
I30       I;
I31       I;
A0        O;
A1        O;
A2        O;
A3        O;
A4        O;
GS        O;
EO        O;
@LOGIC EQUATION
"COMMON PRODUCT TERM"
cpt1 = i26*i27*i28*i29*i30*i31;
cpt2 = i20*i21*i22*i23*i24*i25;
cpt3 = i14*i15*i16*i17*i18*i19;
cpt4 = i8*i9*i10*i11*i12*i13;
A0=/( /i31
     +/i29*i30*i31
     +/i27*i28*i29*i30*i31
     +/i25*cpt1
     +/i23*i24*i25*cpt1
     +/i21*i22*i23*i24*i25*cpt1
     +/i19*cpt2*cpt1
     +/i17*i18*i19*cpt2*cpt1
     +/i15*i16*i17*i18*i9*cpt2*cpt1
     +/i13*cpt3*cpt2*cpt1
     +/i11*i12*i13*cpt3*cpt2*cpt1
     +/i9 *i10*i11*i12*i13*cpt3*cpt2*cpt1
     +/i7 *cpt4*cpt3*cpt2
     +/i5 *i6*i7*cpt4*cpt3*cpt2*cpt1
     +/i3 *i4*i5*i6*i7*cpt4*cpt3*cpt2*cpt1
     +/i1 *i2*i3*i4*i5*i6*i7*cpt4*cpt3*cpt2*cpt1);
A1=/( /i31
     +/i30*i31
     +/i27*i28*i29*i30*i31
     +/i26*i27*i28*i29*i30*i31
```

Figure 22. Encoder Boolean Equations (1 of 2)

```
      +/i23*i24*i25*cpt1
      +/i22*i23*i24*i25*cpt1
      +/i19*cpt2*cpt1
      +/i18*i19*cpt2*cpt1
      +/i15*i16*i17*i18*i19*cpt2*cpt1
      +/i14*i15*i16*i17*i18*i19*cpt2*cpt1
      +/i11*i12*i13*cpt3*cpt2*cpt1
      +/i10*i11*i12*i13*cpt3*cpt2*cpt1
      +/i7 *cpt4*cpt3*cpt2*cpt1
      +/i6 *i7*cpt4*cpt3*cpt2*cpt1
      +/i3 *i4*i5*i6*i7*cpt4*cpt3*cpt2*cpt1
      +/i2 *i3*i4*i5*i6*i7*cpt4*cpt3*cpt2*cpt1);

A2=/( /i31
      +/i30*i31
      +/i29*i30*i31
      +/i28*i29*i30*i31
      +/i23*i24*i25*cpt1
      +/i22*i23*i24*i25*cpt1
      +/i21*i22*i23*i24*i25*cpt1
      +/i20*i21*i22*i23*i24*i25*cpt1
      +/i15*i16*i17*i18*i19*cpt2*cpt1
      +/i14*i15*i16*i17*i18*i19*cpt2*cpt1
      +/i13*cpt3*cpt2*cpt1
      +/i12*i13*cpt3*cpt2*cpt1
      +/i7 *cpt4*cpt3*cpt2*cpt1
      +/i6 *i7*cpt4*cpt3*cpt2*cpt1
      +/i5 *i6*i7*cpt4*cpt3*cpt2*cpt1
      +/i4 *i5*i6*i7*cpt4*cpt3*cpt2*cpt1);

A3=/( /i31
      +/i30*i31
      +/i29*i30*i31
      +/i28*i29*i30*i31
      +/i27*i28*i29*i30*i31
      +/i26*i27*i28*i29*i30*i31
      +/i25*cpt1
      +/i24*i25*cpt1
      +/i15*i16*i17*i18*i19*cpt2*cpt1
      +/i14*i15*i16*i17*i18*i19*cpt2*cpt1
      +/i13*cpt3*cpt2*cpt1
      +/i12*i13*cpt3*cpt2*cpt1
      +/i11*i12*i13*cpt3*cpt2*cpt1
      +/i10*i11*i12*i13*cpt3*cpt2*cpt1
      +/i9 *i10*i11*i12*i13*cpt3*cpt2*cpt1
      +/i8 *i9*i10*i11*i12*i13*cpt3*cpt2*cpt1);

A4=/( /i31
      +/i30*i31
      +/i29*i30*i31
      +/i28*i29*i30*i31
      +/i27*i28*i29*i30*i31
      +/i26*i27*i28*i29*i30*i31
      +/i25*cpt1
      +/i24*i25*cpt1
      +/i23*i24*i25*cpt1
      +/i22*i23*i24*i25*cpt1
      +/i21*i22*i23*i24*i25*cpt1
      +/i20*i21*i22*i23*i24*i25*cpt1
      +/i19*cpt2*cpt1
      +/i18*i19*cpt2*cpt1
      +/i17*i18*i19*cpt2*cpt1
      +/i16*i17*i18*i19*cpt2*cpt1);

eo  = /(i0*i1*i2*i3*i4*i5*i6*i7
        *i8*i9*i10*i11*i12*i13*i14*i15
        *i16*i17*i18*i19*i20*i21*i22*i23
        *i24*i25*cpt1);
gs = /eo;
```

Figure 22. Encoder Boolean Equations (2 of 2)

4-BIT SYNCHRONOUS COUNTER

This counter produces a binary count on outputs Count3 – Count0. Note the required reset (RST) input to initialize all of the flip-flops. The inputs for each flip-flop were first determined by drawing the desired output waveforms. Next, Karnaugh maps were used to reduce the number of terms and determine the logic equations for the input to each flip-flop. This technique could be used to construct a counter whose outputs produce some count other than binary.

The simulation only consists of a reset, followed by a number of clocks to count from 0 through 15 and back to 0.

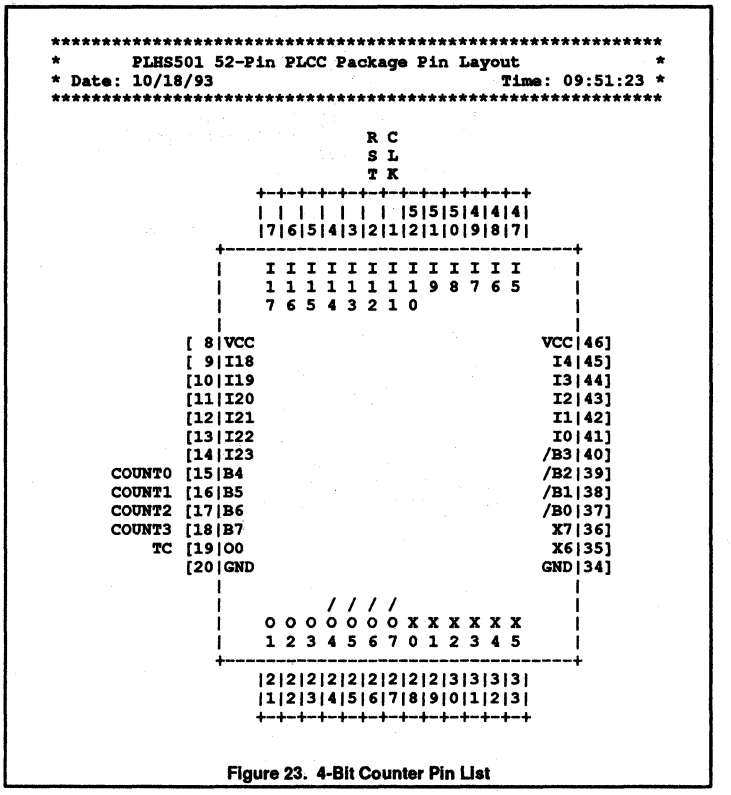


Figure 23. 4-Bit Counter Pin List

```
         "4 bit synchronous counter"
@PINLIST
CLK      I;
RST      I;
COUNT0   O;
COUNT1   O;
COUNT2   O;
COUNT3   O;
TC       O;

@LOGIC EQUATION

"INPUTS FOR EACH FLIP-FLOP"

DATA1 = ((CQ1*CQN0)+(CQN1*CQ0));
DATA2 = ((CQ0*CQ1*CQN2)+(CQN0*CQ2)+(CQN1*CQ2));
DATA3 = ((CQN2*CQ3)+(CQN0*CQ3)+(CQ0*CQ1*CQ2*CQN3)+(CQN1*CQ3));

"4 D-TYPE FLIP FLOPS CONNECTED AS A SYNCHRONOUS COUNTER"

CSN0 = /(CLK*RST*(/(CSN0*(/(CQN0*RST*CRN0)))));
CRN0 = /(CSN0*CLK*(/(CQN0*RST*CRN0)));
CQ0  = /(CSN0*CQN0);
CQN0 = /(CRN0*CQ0*RST);

CSN1 = /(CLK*RST*(/(CSN1*(/(DATA1*RST*CRN1)))));
CRN1 = /(CSN1*CLK*(/(DATA1*RST*CRN1)));
CQ1  = /(CSN1*CQN1);
CQN1 = /(CRN1*CQ1*RST);

CSN2 = /(CLK*RST*(/(CSN2*(/(DATA2*RST*CRN2)))));
CRN2 = /(CSN2*CLK*(/(DATA2*RST*CRN2)));
CQ2  = /(CSN2*CQN2);
CQN2 = /(CRN2*CQ2*RST);

CSN3 = /(CLK*RST*(/(CSN3*(/(DATA3*RST*CRN3)))));
CRN3 = /(CSN3*CLK*(/(DATA3*RST*CRN3)));
CQ3  = /(CSN3*CQN3);
CQN3 = /(CRN3*CQ3*RST);

"Connection to output pins"

count0=cq0;
count1=cq1;
count2=cq2;
count3=cq3;

"TERMINAL COUNT PIN"

TC=(CQ0*CQ1*CQ2*CQ3);
```

Figure 24. 4-Bit Counter Boolean Equations

VME Bus EPROM Interface

The idea for this VMEbus EPROM board came from *WIRELESS WORLD CIRCUIT IDEAS*, January, 1988. The implementation was done by a Philips' FAE, John McNally.

The board contains two banks of EPROMs. Each bank consists of either two 27128s or two 27256s; each of which can be enabled by comparing the address location fo the board. Decoding three other address bits selects which of the banks is accessed. A 4-bit shift register combined with four jumpers provide wait states.

The circuit drawing was entered onto a PC using FutureNet DASH, a schematic capture package (Figures 25, 26, and 27). It was then converted to logic equations using SNAP (Figures 29 and 30) and then assembled into a PLHS501.

This application, which originally needed eight ICs, used forty-four of the available seventy-two NAND Foldback Terms and forth of the available fifty-two pins. As the PLHS501 contains no registers, an edge-triggered D-type flip-flop was designed using NAND gates and this is used as a soft macro in order to implement the shift register function (Figure 27).

As suggested in the original article, the circuit could be expanded to access up to eight ROM banks (Figure 28B). This was achieved by editing the logic equation file and adding extra equations (Figure 32). Modifying the drawing, although fairly easy to do, was not considered necessary as the object was to design with PML and not TTL. The expanded circuit would require another three TTL IC packages, brining the total to eleven if done using TTL devices. The number of foldback terms increased to fifty-five, with the number of pins rising to fifty. Figure 28 shows the pinout of both versions.

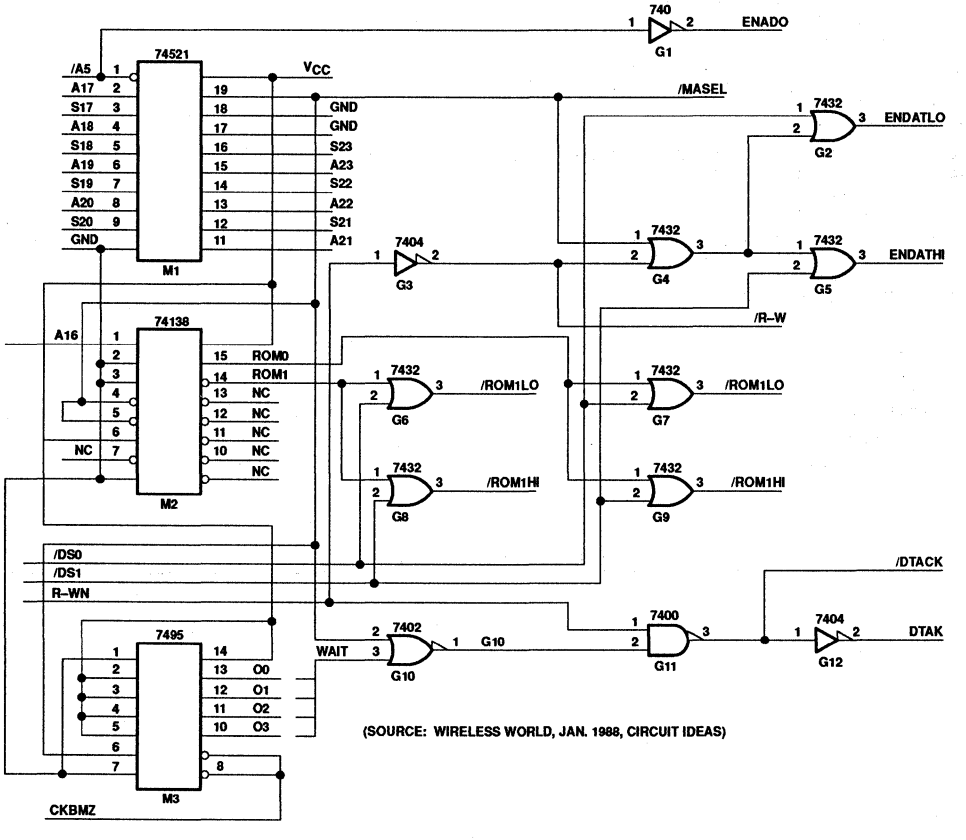


Figure 25. VME – EPROM Interface

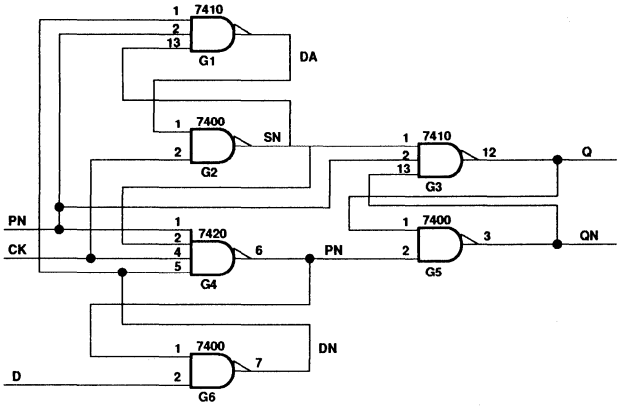


Figure 26. Edge-Triggered D Flip-Flop (DFFS)

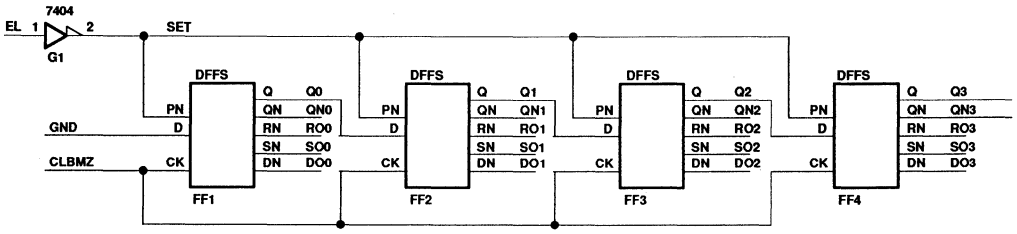
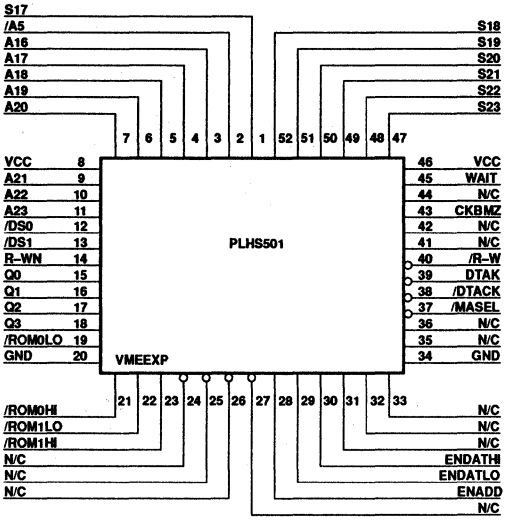


Figure 27. 4-Bit Shifter (7495)

A. 2 ROM BANKS AS ORIGINAL CIRCUIT.
REPLACES 8 PACKAGES – USES 46 FOLDBACK TERMS



B. EXPANDED TO 8 ROM BANKS
REPLACES 11 PACKAGES – USES 55 FOLDBACK TERMS

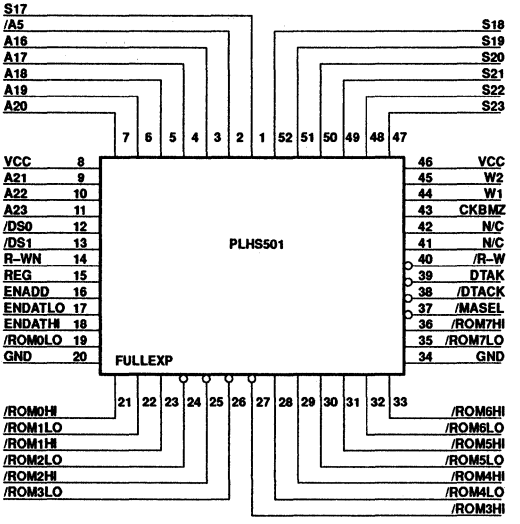


Figure 28. VMEEXP and FULLEXP

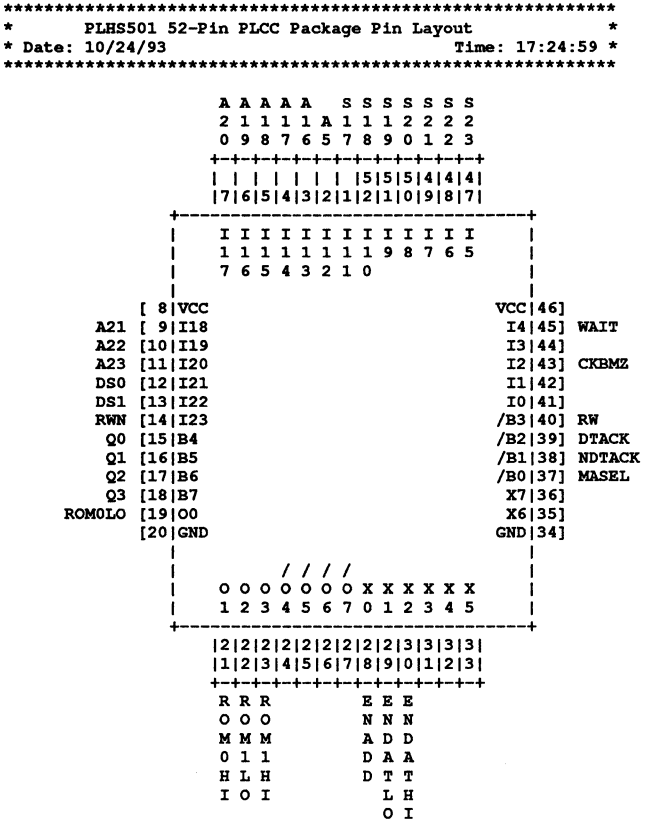


Figure 29. VMEEXP PLHS501 Pinlist

```
@PINLIST

WAIT        I;
CKBMZ       I;
DS[1..0]    I;
RWN         I;
A5          I;
A[16..23]   I;
S[17..23]   I;

Q[0..3]     O;      ENDATLO    O;
ROM0LO      O;      ENDATHI    O;
ROM0HI      O;      RW         O;
ROM1LO      O;      NDTACK     O;
ROM1HI      O;      DTACK      O;
ENADD       O;      MASEL      O;

@LOGIC EQUATIONS

RO3 = (/(MASEL*SO3*CKBMZ*DO3));
SO3 = (/(CKBMZ*(/(SO3*DO3*MASEL))));
DO3 = (/(Q2*RO3));
RO2 = (/(MASEL*SO2*CKBMZ*DO2));
SO2 = (/(CKBMZ*(/(SO2*DO2*MASEL))));
DO2 = (/(Q1*RO2));
RO1 = (/(MASEL*SO1*CKBMZ*DO1));
SO1 = (/(CKBMZ*(/(SO1*DO1*MASEL))));
DO1 = (/(Q0*RO1));
RO0 = (/(MASEL*SO0*CKBMZ*DO0));
SO0 = (/(CKBMZ*(/(SO0*DO0*MASEL))));
DO0 = (/(0*RO0));

ROM0LO = (/DS0+(/(/A16*MASEL)));
ROM0HI = (/DS1+(/(/A16*MASEL)));
ROM1LO = (/DS0+(/( A16*MASEL)));
ROM1HI = (/DS1+(/( A16*MASEL)));

Q0 = (/(((/(RO0*Q0))*SO0*(MASEL)));
Q1 = (/(((/(RO1*Q1))*SO1*(MASEL)));
Q2 = (/(((/(RO2*Q2))*SO2*(MASEL)));
Q3 = (/(((/(RO3*Q3))*SO3*(MASEL)));

MASEL = /(/(/(/(/((A17*S17+/A17*/S17)
                  *(A18*S18+/A18*/S18)
                  *(A19*S19+/A19*/S19)
                  *(A20*S20+/A20*/S20)
                  *(A21*S21+/A21*/S21)
                  *(A22*S22+/A22*/S22)
                  *(A23*S23+/A23*/S23)
                  *(A5)))));

NDTACK  = /(((/(MASEL+WAIT))*RWN);
DTACK   = /NDTACK;
RW      = /(RWN);
ENADD   = A5;
ENDATLO = ((RW+MASEL)+/DS0);
ENDATHI = ((RW+MASEL)+/DS1);
```

Figure 30. VMEEXP PLHS501 .BEE File

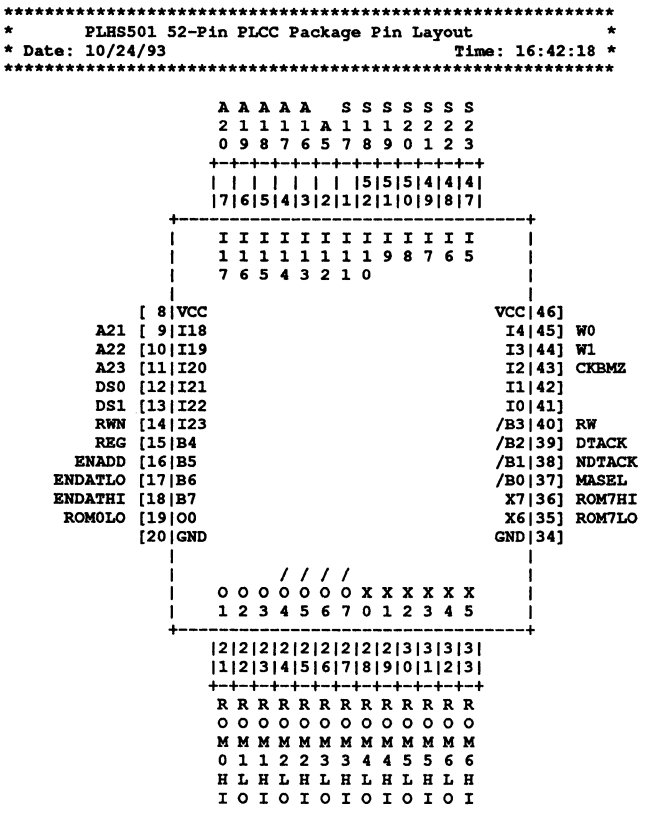


Figure 31. FULLEXP Pinlist

```
@PINLIST
CKBMZ       I;
DS[1..0]    I;
RWN         I;
A5          I;
A[16..23]   I;
S[17..23]   I;
W[1..0]     I;
                    ENDATLO    O;
ROM0LO      O;      ENDATHI    O;
ROM0HI      O;      RW         O;
ROM1LO      O;      NDTACK     O;
ROM1HI      O;      DTACK      O;
ENADD       O;      MASEL      O;
@GROUPS
ADDR = A[18..16];
@LOGIC EQUATIONS
RO3 = (/(MASEL*SO3*CKBMZ*DO3));
SO3 = (/(CKBMZ*(/(SO3*DO3*MASEL))));
DO3 = (/(Q2*RO3));
RO2 = (/(MASEL*SO2*CKBMZ*DO2));
SO2 = (/(CKBMZ*(/(SO2*DO2*MASEL))));
DO2 = (/(Q1*RO2));
RO1 = (/(MASEL*SO1*CKBMZ*DO1));
SO1 = (/(CKBMZ*(/(SO1*DO1*MASEL))));
DO1 = (/(Q0*RO1));
RO0 = (/(MASEL*SO0*CKBMZ*DO0));
SO0 = (/(CKBMZ*(/(SO0*DO0*MASEL))));
DO0 = (/(0*RO0));
ROM0LO = (/DS0+(/((addr == 0h)*MASEL)));
ROM0HI = (/DS1+(/((addr == 0h)*MASEL)));
ROM1LO = (/DS0+(/((addr == 1h)*MASEL)));
ROM1HI = (/DS1+(/((addr == 1h)*MASEL)));
ROM2LO = (/DS0+(/((addr == 2h)*MASEL)));
ROM2HI = (/DS1+(/((addr == 2h)*MASEL)));
ROM3LO = (/DS0+(/((addr == 3h)*MASEL)));
ROM3HI = (/DS1+(/((addr == 3h)*MASEL)));
ROM4LO = (/DS0+(/((addr == 4h)*MASEL)));
ROM4HI = (/DS1+(/((addr == 4h)*MASEL)));
ROM5LO = (/DS0+(/((addr == 5h)*MASEL)));
ROM5HI = (/DS1+(/((addr == 5h)*MASEL)));
ROM6LO = (/DS0+(/((addr == 6h)*MASEL)));
ROM6HI = (/DS1+(/((addr == 6h)*MASEL)));
ROM7LO = (/DS0+(/((addr == 7h)*MASEL)));
ROM7HI = (/DS1+(/((addr == 7h)*MASEL)));
Q0 = (/(((/RO0*Q0))*SO0*(MASEL)));
Q1 = (/(((/RO1*Q1))*SO1*(MASEL)));
Q2 = (/(((/RO2*Q2))*SO2*(MASEL)));
Q3 = (/(((/RO3*Q3))*SO3*(MASEL)));
MASEL = /(/(/(/((A17*S17+/A17*/S17)
                *(A18*S18+/A18*/S18)
                *(A19*S19+/A19*/S19)
                *(A20*S20+/A20*/S20)
                *(A21*S21+/A21*/S21)
                *(A22*S22+/A22*/S22)
                *(A23*S23+/A23*/S23)
                *(A5)))));
NDTACK  = /(((/(MASEL+/((/q0*w0*/w1)+(q1*w0*/w1)+(/q2*/w0*w1)
          +(/q3*w0*w1))))*RWN);
DTACK   = /NDTACK;
RW      = /(RWN);
ENADD   = A5;
ENDATLO = ((RW+MASEL)+/DS0);
ENDATHI = ((RW+MASEL)+/DS1);
```

Figure 32. FULLEXP PLHS501 .BEE File

PLHS501 design examples AN049

MICRO CHANNEL INTERFACE

IBM's new Micro Channel Architecture (MCA) bus implements new features not found on the XT/AT bus. One new requirement for adapter designers is that of Programmable Option Select (POS) circuitry. It allows system software to configure each adapter card upon power on, thereby eliminating option select switches or jumpers on the main logic board and on adapter cards.

Each adapter card slot has its own unique –CDSETUP signal routed to it. This allows the CPU to interrogate each card individually upon power up. By activating a card's –CDSETUP line along with appropriate address and control lines two unique 8 bit ID numbers are first read from the adapter. Based upon the ID number, the system then writes into the card's option latches configuration information that has been stored in the system's CMOS RAM. The CPU also activates POS latch address 102h bit 0, which is designated as a card enable bit.

If a new card is added to the system, an auto-configuration utility will be invoked. Each adapter card has associated with it a standardized Adapter Description File with filename of @XXXX.ADF, where XXXX is the hex ID number of the card. The configuration utility prompts the user according to the text provided in the .ADF file and updates the card's latches and the system's CMOS RAM.

IBM reserves 8 addresses for byte-wide POS latches, however, depending on the card's function, not all addresses need to be used. In addition, of those addresses that are used, only the bits used need to be latched. The first two addresses which are reserved for reading the ID bytes, and bit 0 of the third address, which is defined as a card enable bit, are mandatory. Some of the remaining bits of the third address are suggested by IBM to be used as inputs to an I/O or memory address comparator to provide for alternate card addresses. Many adapter cards will not use more than these three POS locations.

The following example describes an implementation of POS circuitry realized in a PLHS501. It uses only 56 of the possible 72 internal foldback NAND gates and only a portion of the device pins, allowing additional circuitry to ba added. Figure 33 shows a block diagram of the circuit, and Figures 35 and 36 are the SNAP files. Pins labeled DO0–DO7 must be connected externally to pins DIN0–DIN7. They also must be connected through a 74F245 transceiver to the Micro Channel. External transceiver direction and enable control is provided for by circuitry within the PLHS501. The external transceiver may also be used by other devices on the adapter card.

In this application, edge-triggered registers are not required and therefore should not be used, as transparent latches use fewer NAND gates to implement. Figure 34 shows the various latch circuits described by the SNAP equations. POS byte 2 was made using four of the /B device pins and four of the B pins. Notice however, from Figure 34(B) that the bits on the /B pins used the complement of the input pin, thereby implementing a non-inverting latch. Also, all 8 bits of this byte were brought to output pins. If some of the bits are not used by external circuitry, then the specific bit latch may not be needed or may be constructed entirely from foldback NAND gates freeing additional pins.

An external F521 may be added to provide for I/O address decoding. As the MCA bus requires all 16 bits of the I/O address to be decoded, 8 bits may be assigned to the F521 and 8 bits to the 501. Bit fields decoded in the 501 may be done so in conjunction with bits from POS byte 2 to provide for alternate I/O addressing. Additionally, some of the available 501 outputs may be used as device enables for other devices on the card.

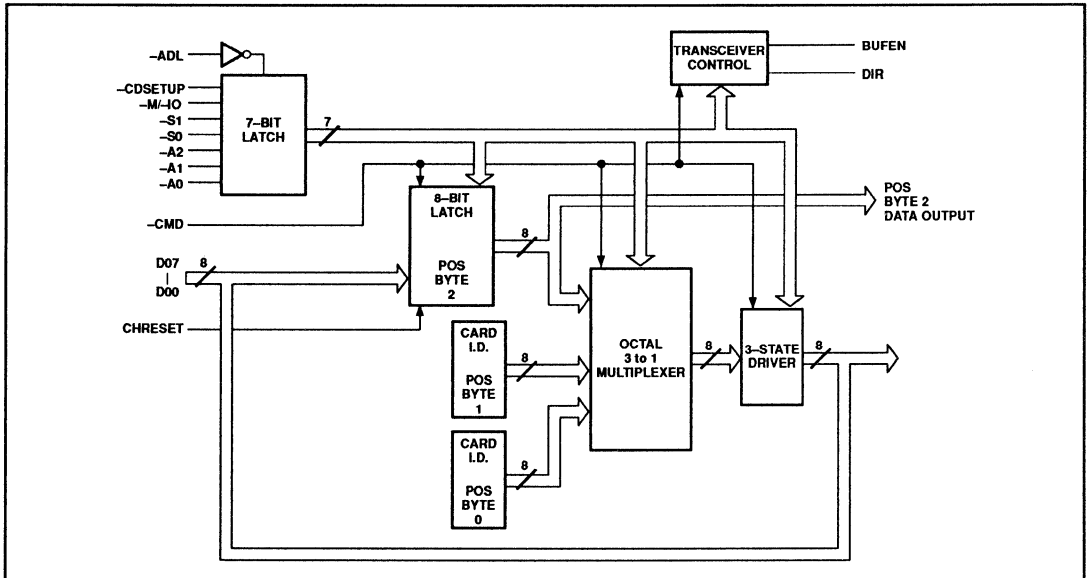
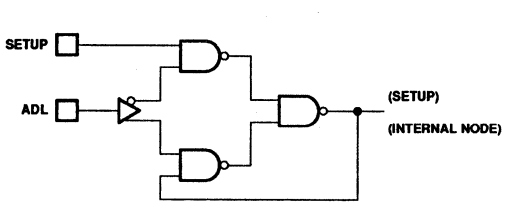
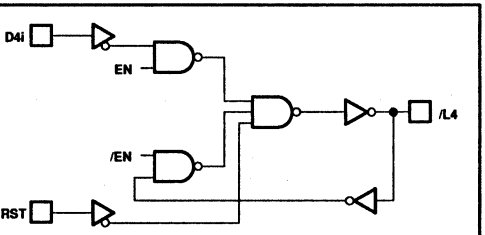


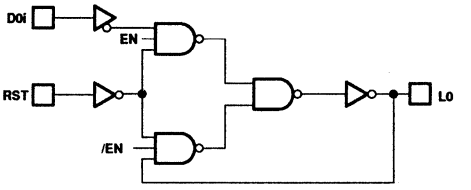
Figure 33. Block Diagram of Basic POS Implementation in PLHS501



(A) Control Signal Input Latch (1 of 7)



(B) Data Latch of Bits 4 – 7



(C) Data Latch of Bits 0 – 3

Figure 34. Latches Used in MCA Interface

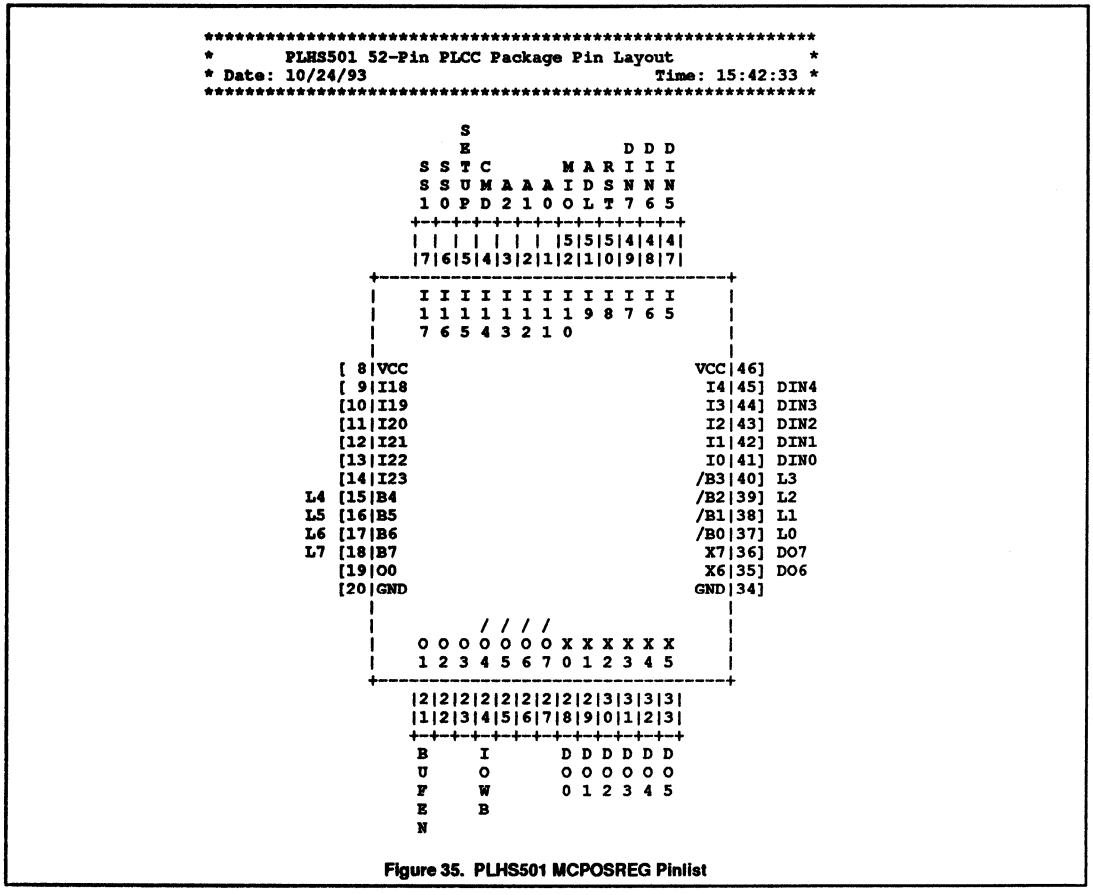


Figure 35. PLHS501 MCPOSREG Pinlist

```
"
 Basic Programmable Option Select circuitry
    for a Micro Channel Adaptor card
"

@PINLIST

a[2..0]   i;          l[7..0]  o;
cmd       i;          do[7..0] o;
setup     i;          bufen    o;
ss[1..0]  i;          iowb     o;
din[7..0] i;
adl       i;
mio       i;
rst       i;

@LOGIC EQUATIONS

 read0 = (nsetupl*/ssll*ss01*nmiol*/cmd*na2l*/all*/a01);
 read1 = (nsetupl*/ssll*ss01*nmiol*/cmd*na2l*/all* a01);
 read2 = (nsetupl*/ssll*ss01*nmiol*/cmd*na2l* all*/a01);

 b7hi = 0;  " Define high ID byte "
 b6hi = 1;  "    (POS byte #1)     "
 b5hi = 1;  "       7E hex         "
 b4hi = 1;
 b3hi = 1;
 b2hi = 1;
 b1hi = 1;
 b0hi = 0;

 b7lo = 1;    " Define low ID byte "
 b6lo = 1;    "     (POS byte #0)    "
 b5lo = 1;    "       FF hex         "
 b4lo = 1;
 b3lo = 1;
 b2lo = 1;
 b1lo = 1;
 b0lo = 1;

   " 7-Bit Input Latch for Control Signals "

nsetupl = /setup*/adl + nsetupl*adl;
nmiol   = /mio *adl   + nmiol  *adl;
ssll    = ss1  */adl  + ssll   *adl;
ss01    = ss0  */adl  + ss01   *adl;
na2l    = /a2  */adl  + na2l   *adl;
all     = a1   */adl  + all    *adl;
a01     = a0   */adl  + a01    *adl;

   " Option Select Octal Data Latch (POS byte #2) "
   " 10 is to be used as a card enable signal"

nen = /(nsetupl*/ss01*ssll*nmiol*/cmd*na2l*all*/a01); "write to latch"

17 = /(/din7 * /nen)   * /(/17 * nen) * (/rst);
16 = /(/din6 * /nen)   * /(/16 * nen) * (/rst);
15 = /(/din5 * /nen)   * /(/15 * nen) * (/rst);
14 = /(/din4 * /nen)   * /(/14 * nen) * (/rst);
13 = /(/( din3 * /nen * /rst) * /(13 * nen  * /rst));
12 = /(/( din2 * /nen * /rst) * /(12 * nen  * /rst));
11 = /(/( din1 * /nen * /rst) * /(11 * nen  * /rst));
10 = /(/( din0 * /nen * /rst) * /(10 * nen  * /rst));
```

Figure 36. PLHS501 MCPOSREG .EQN File (1 of 2)

```
    " Octal 3 to 1 Multiplexer "
    " This multiplexer selects between reading
      POS[0], POS[1] or POS[2] onto the data bus"

ido7 = (b7hi*read1 + b7lo*read0 + 17*read2);
ido6 = (b6hi*read1 + b6lo*read0 + 16*read2);
ido5 = (b5hi*read1 + b5lo*read0 + 15*read2);
ido4 = (b4hi*read1 + b4lo*read0 + 14*read2);
ido3 = (b3hi*read1 + b3lo*read0 + 13*read2);
ido2 = (b2hi*read1 + b2lo*read0 + 12*read2);
ido1 = (b1hi*read1 + b1lo*read0 + 11*read2);
ido0 = (b0hi*read1 + b0lo*read0 + 10*read2);

  "3-State output control for do7-do0"

do[7..0] = ido[7..0];
do[7..0].oe = (nsetupl*/ss1l*ss0l*nmiol*/cmd*na2l*outen);
outen =/(a1l*a0l);

  "External F245 transceiver control"

iowb  = /(na2l * nsetupl * nmiol * ss1l * /ss0l);
niow  = /(na2l * nsetupl * nmiol * ss1l * /ss0l);
bufen = cmd * niow;
```

Figure 36. PLHS501 MCPOSREG .EQN File (2 of 2)

NuBus INTERFACE

In Apple Computer's book* "*Designing Cards and Drivers for Macintosh II and Macintosh SE*", an application was described for interfacing an 8-bit I/O controller to the NuBus. The controller used was a SCSI controller of the type used on the main Macintosh logic board. Seven devices (three of which were PAL architecture) were used as control circuitry interfacing the SCSI controller and two RAM chips to the bus.

This example of using the PLHS501 shows a method of interfacing the same SCSI controller and RAM chips to the NuBus using only three parts. The adapter card schematic is shown in Figure 38, the SNAP pin listing is in Figure 42, and the SNAP .EQN listing is in Figure 43. Although the SNAP listing may seem confusing at first glance, the circuitry fused into the PLHS501 can be broken down into small blocks of latches, flip-flops, and schematically in Figures 40 and 41. Circuit timing is shown in Figure 39.

Referring to Figure 40 and Figure 41, the circuitry starts a transaction by first detecting a valid address in either the slot or super slot range. The detection is accomplished by two wide-input NAND gates, and controlled by the /CLK signal. Following each NAND gate is an S-R latch to hold the signal until near the end of the cycle. The two S-R latch signals are combined into one signal named ST0 such that if either NAND gate output was low, then some delay time after the rising edge of /CLK, ST0 will go low. The next rising edge of /CLK will cause signal ST1 to go low. This sets signal DE2 low, which is an input to an external flip-flop to cause ST2 to go low at the next rising /CLK edge terminating the cycle. An external flip-flop was necessary to achieve a high-speed /CLK to /IOR and /ACK transition. Also, an external FI25 buffer was added to meed the soon to be approved IEEE P1196 specification requirement of 60mA I_{OL} for signal /NMRQ and 24mA I_{OL} for signals /TM0/TM1 and /ACK. Figure 41(B) shows an easily implemented latch which controls interrupts generated by the SCSI controller passing onto the bus. Upon /RESET the latch is put into a known state. Under software control, by writing to a decoded address, the latch may be set or reset, thereby gating or blocking the interrupt signals.

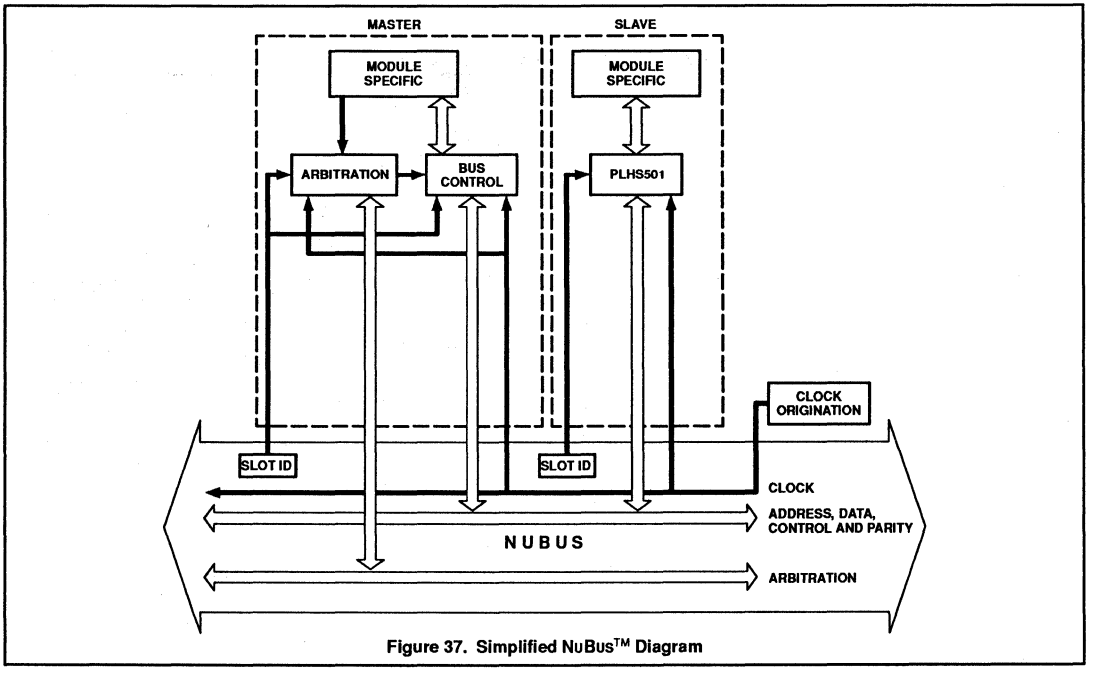


Figure 37. Simplified NuBus™ Diagram

* *Designing Cards and Drivers for Macintosh II and Macintosh SE*, Addison–Wesley Publishing Company, Inc. 1987.

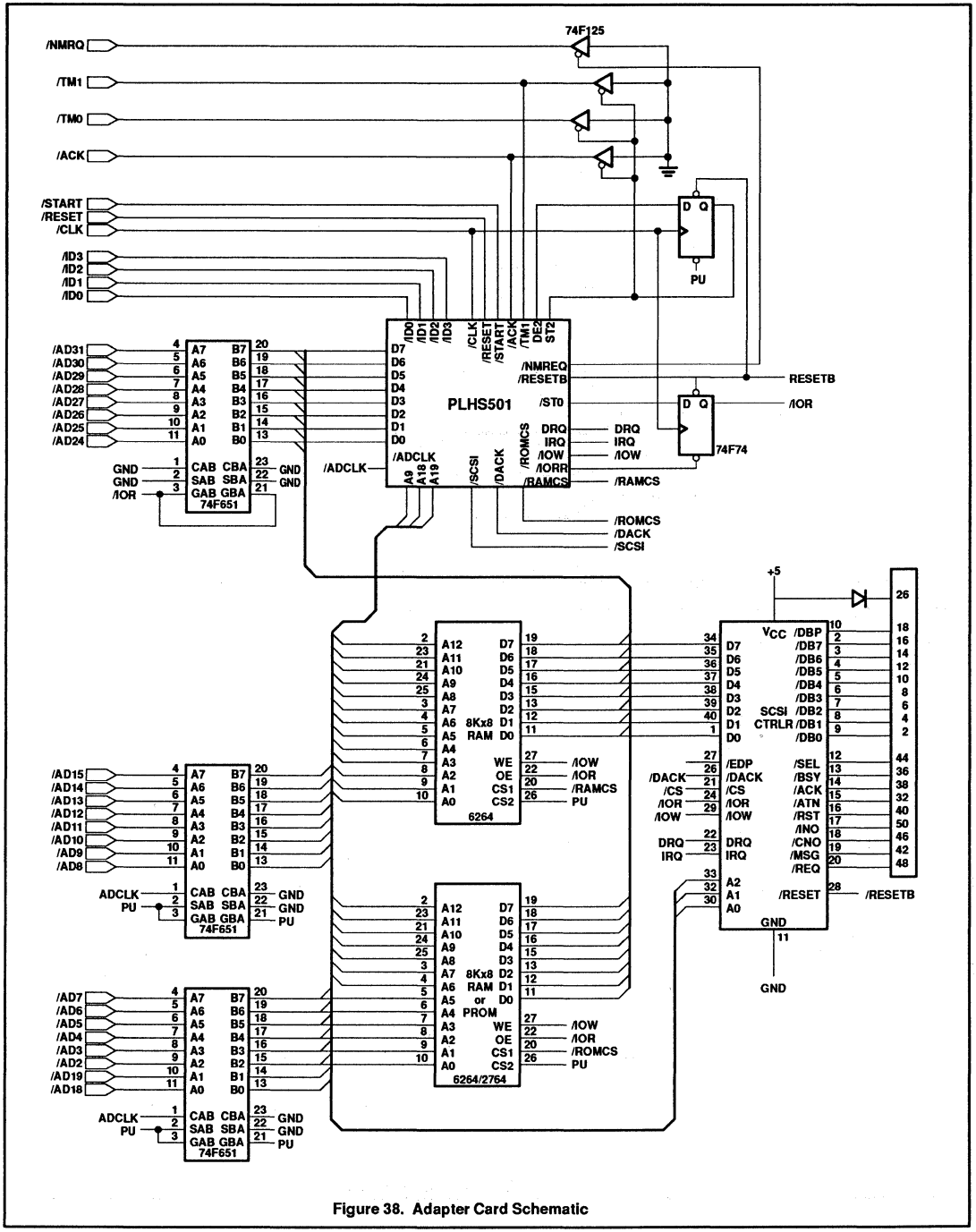


Figure 38. Adapter Card Schematic

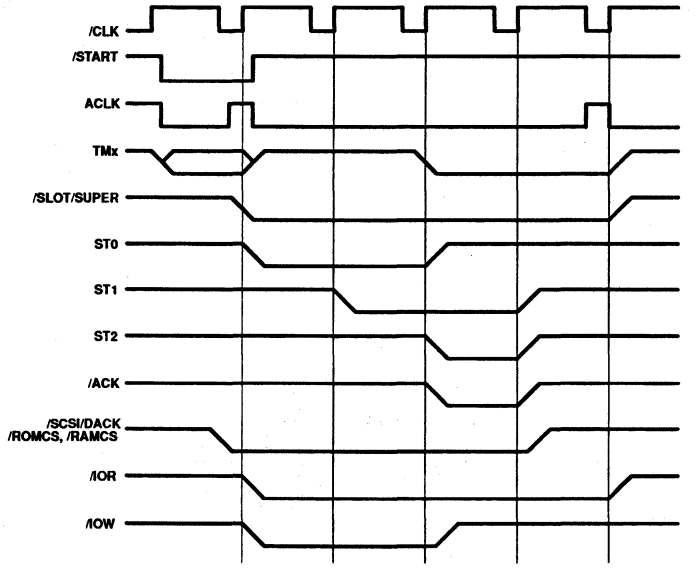


Figure 39. Timing Diagram

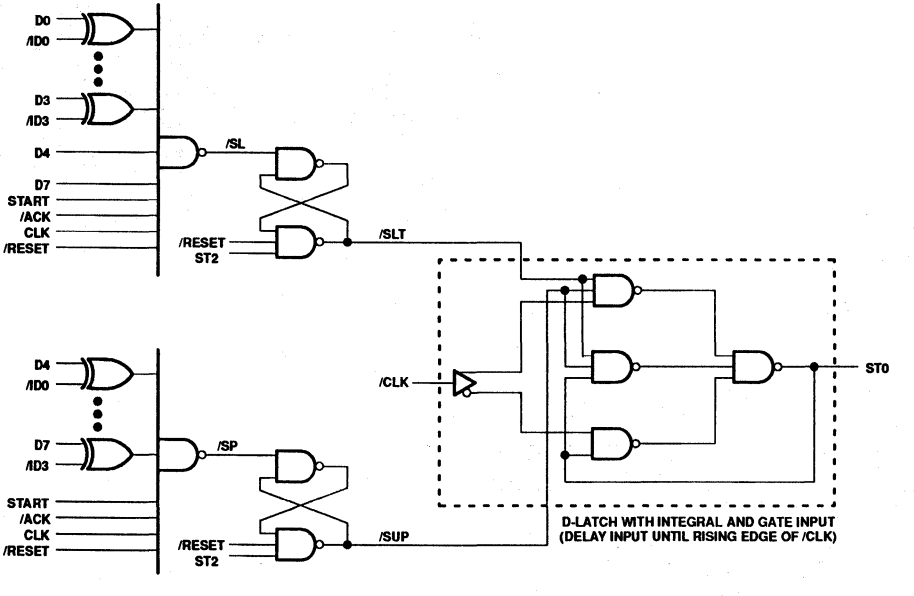
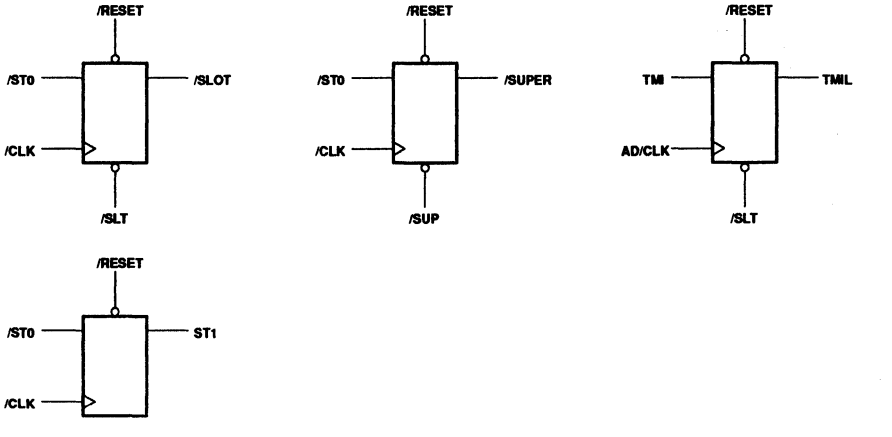
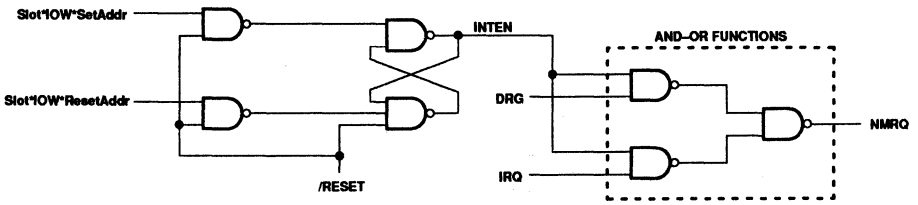


Figure 40. Decoding and Latch Circuitry



(A) Four Internal Flip-Flops Constructed from NAND Gates.



(B) Interrupt Enable Control Latch
Internal Flip-Flops and Latches

Figure 41. Internal Flip-Flops and Latches

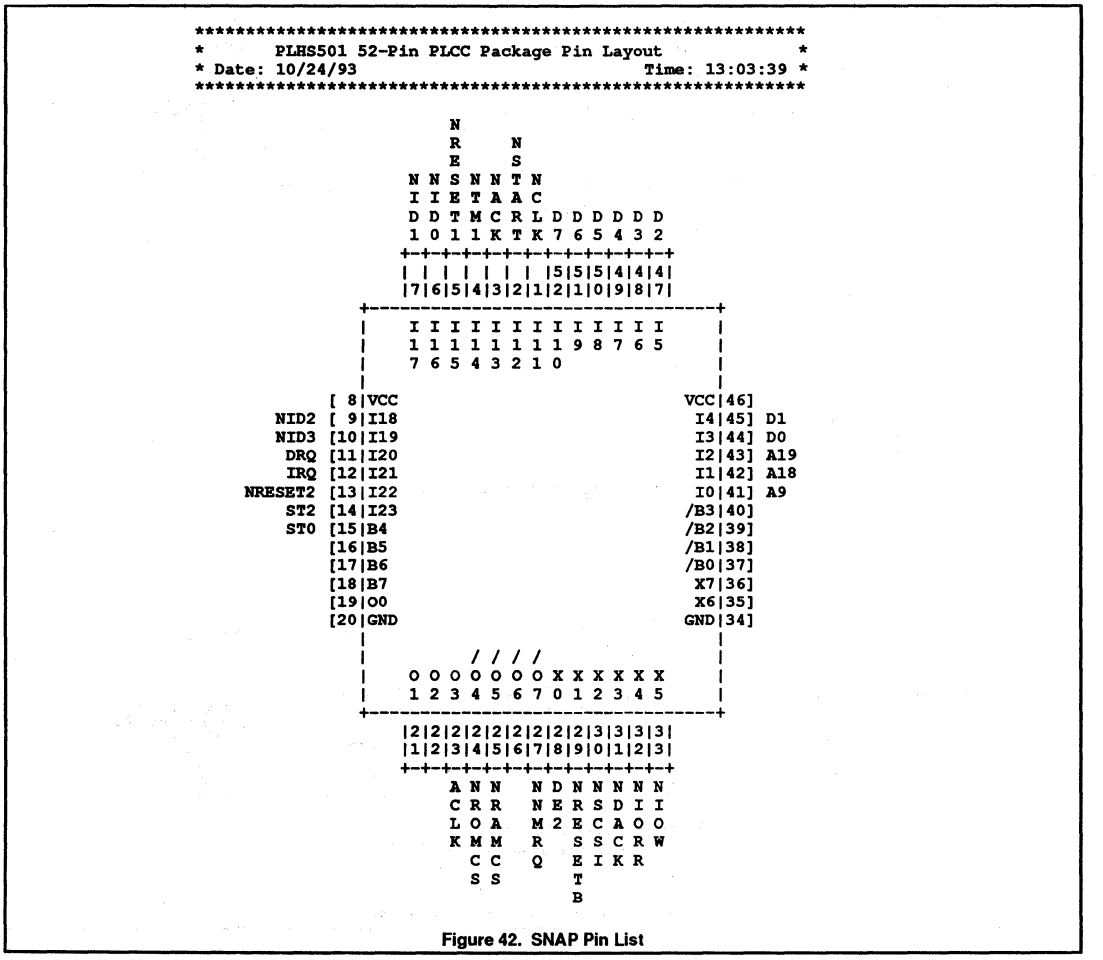


Figure 42. SNAP Pin List

```
   "SCSI-NuBus Interface"
@PINLIST
nid[0..3] i;          nromcs   o;
d[7..0]   i;          nramcs   o;
a9        i;          nnmrq    o;
a[19..18] i;          nresetb  o;
nreset1   i;          nscsi    o;
nreset2   i;
ntml      i;          ndack    o;
nack      i;          niorr    o;
nstart    i;          niow     o;
nclk      i;          st0      b;
drq       i;          aclk     o;
irq       i;          de2      o;
st2       i;
@LOGIC EQUATION
                          "Address Decode"
cmp0a = (d0*/nid0+/d0*nid0);
cmp1a = (d1*/nid1+/d1*nid1);
cmp2a = (d2*/nid2+/d2*nid2);
cmp3a = (d3*/nid3+/d3*nid3);
cmp0b = (d4*/nid0+/d4*nid0);
cmp1b = (d5*/nid1+/d5*nid1);
cmp2b = (d6*/nid2+/d6*nid2);
cmp3b = (d7*/nid3+/d7*nid3);
nsl = /(d7*d6*d5*d4*cmp0a*cmp1a*cmp2a*cmp3a*/nstart*nack*/nclk);
nsp = /(cmp0b*cmp1b*cmp2b*cmp3b*/nstart*nack*/nclk);
                        "latch slot signal"
nslt = /(nreset1*st2*/(nsl*nslt));
                        "latch super signal"
nsup = /(nreset1*st2*/(nsp*nsup));
                       "Let nslt or nsup through only
                        until after the rising edge
                        of nclk"
ist0 = /(/(nslt*nsup*nclk) * /(st0*/nclk) * /(nslt*nsup*st0) * nreset1);
st0 = ist0;
st0.oe = 1;
                        "Slot signal D-type Flip Flop"
nslot.d = st0;
nslot.clk = nclk;
nslot.set = nreset2;
nslot.rst = nslt;
                        "Super signal D-type Flip Flop"
nsuper.d = st0;
nsuper.clk = nclk;
nsuper.set = nreset2;
nsuper.rst = nsup;
                         "State 1 D-type Flip Flop"
st1.d = st0;
st1.clk = nclk;
st1.set = nreset2;
                         "output to external flop"
de2  = /(/st1 * st2);
                         "address latch clock"
adclk = /nclk*st0*st1;
aclk  = /nclk*st0*st1;
                         "latch tml signal for r/w info"
tml1.d = ntml;
tml1.set = nreset2;
tml1.clk = adclk;
tml1.rst = nslt;
```

Figure 43. SNAP .EQN Listing (1 of 2)

```
"
tmll  -> 1 read, 0 write
"
                        "straight decode stuff"
niorr = /(/st0*tmll                     * nreset1);
niow  = /(/tmll*/st0                    * nreset1);
nscsi = /(/nslot*/al9*/al8*/a9          * nreset1);
ndack = /(/nslot*/al9*/al8* a9          * nreset2);
nromcs= /(/nslot* al9* al8              * nreset2);
nramcs= /(/nsuper                       * nreset2);
nresetb= nreset2;

                         "interrupt control latch"
setad = /(/tmll*/st0*/nslot* al9*/al8* a9);
rstad = /(/tmll*/st0*/nslot* al9*/al8*/a9);
inten = /(setad*(/(inten*rstad*nreset2)));
nnmrq = /(inten*drq+inten*irq);
```

Figure 43. SNAP .EQN Listing (1 of 2)

PLHS501 design examples AN049

Data Bus Parity

The PLHS501 can span 32 bits of input data. It has four output Ex-OR gates, and the ability to generate literally any function of the inputs. It would seem that there must be some "best" way to generate and detect parity. Recall that the PLHS501 can generate both deep logic functions (lots of levels) and wide logic functions (lots of inputs). The best solution would require the fewest gates and the fewest number of logic levels. Let's review the basics, first. Table 1(A) shows the parity function for two variables and Table 1(B) shows it for three variables. The Ex-OR function generates even parity.

It is noticeable that there are precisely 50% logical 1 entries in the truth tables. This yields the famous checkerboard Karnaugh Maps. With a checkerboard K-map, no simplification of Ex-OR functions is possible by Boolean simplification. The two variable Ex-OR has two ones (implying 3 gates to generate), the 3 variable has four ones (implying 5 gates to generate). In general, $2^{n-1}+1$ product terms could generate Ex-OR functions in two levels of NAND gates (assuming complementary input variables exist). You must have an unlimited number of gate inputs for this to hold.

The PLHS501 could do this for 7 input variables in two levels ($2^6+1=65$), but cannot support 8 ($2^7+1=129$). Hence, it is appropriate to seek a cascaded solution, hopefully taking advantage of the available output Ex-OR functions. Let's solve a 16 input Ex-OR function, by subpartitioning. First, consider Figure 44(A) where two literals are Exclusive-ORed to generate an intermediate Ex-OR function. This requires available complementary inputs and generates even parity in two levels. Figure 44(B) also does this (by factoring), requiring 3 gate levels, but does not require complementary inputs.

Assuming inputs must get into the PLHS501 through the pin receivers, it is best to generate as wide of an initial Ex-OR as possible, so a structure like Figure 44(A) expanded is appropriate. Figure 44 shows a 2-level 4 input Ex-OR function which may be viewed as a building block. This structure may be repeated four times, across four sets of four input bits generating partial intermediate parity values which may then be treated through two boxes similar to Figure 44(B). These outputs are finally combined through an output Ex-OR at a PLHS501 output pin. Figure 46 shows the complete solution which requires 44 NANDs plus one Ex-OR.

Figures 47 and 48 show the pin layout and SNAP equations for a parity generator. This example uses a cascade with a different partitioning than just previously discussed.

Table 1. Even Parity Functions

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Table 1(A).

A	B	C	A ⊕ B
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Table 1(B).

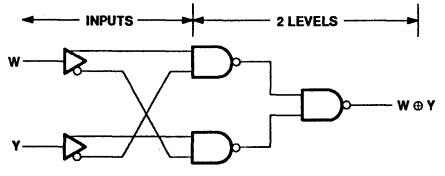


Figure 44(A).

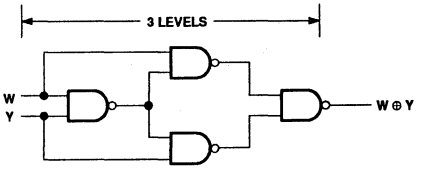


Figure 44(B).

Figure 44. Complementary Input Levels

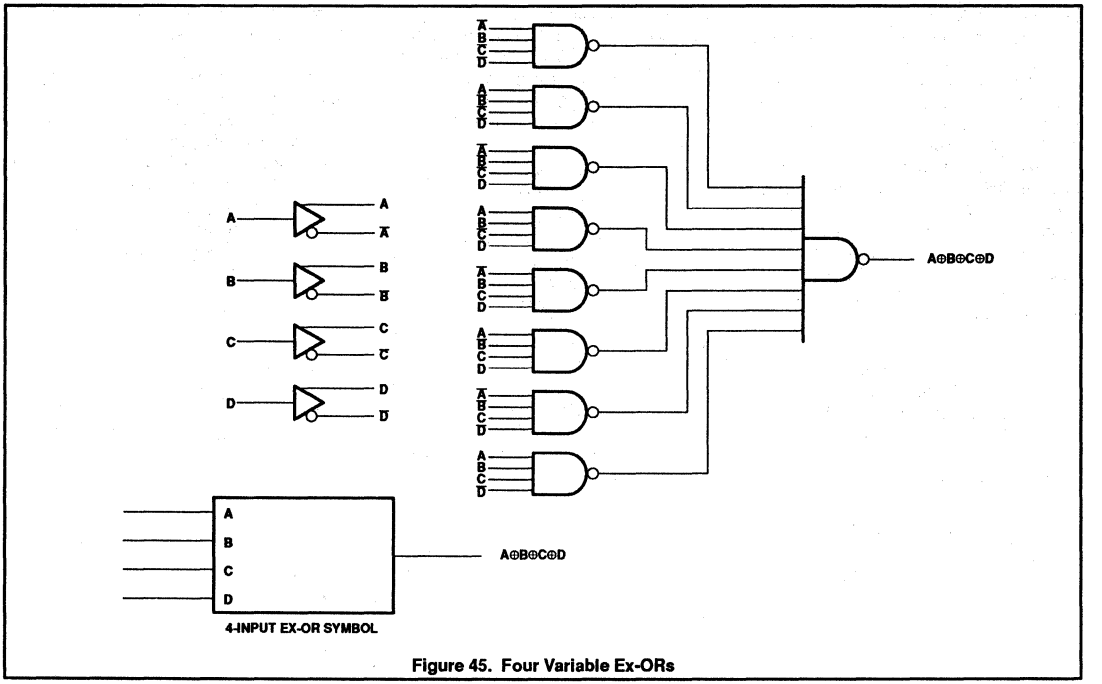


Figure 45. Four Variable Ex-ORs

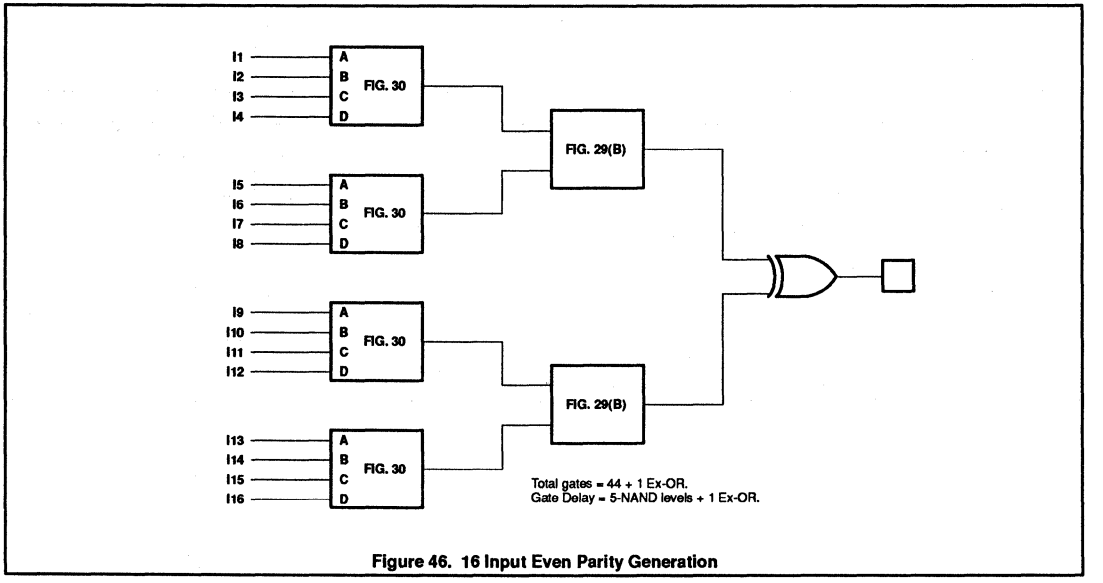


Figure 46. 16 Input Even Parity Generation

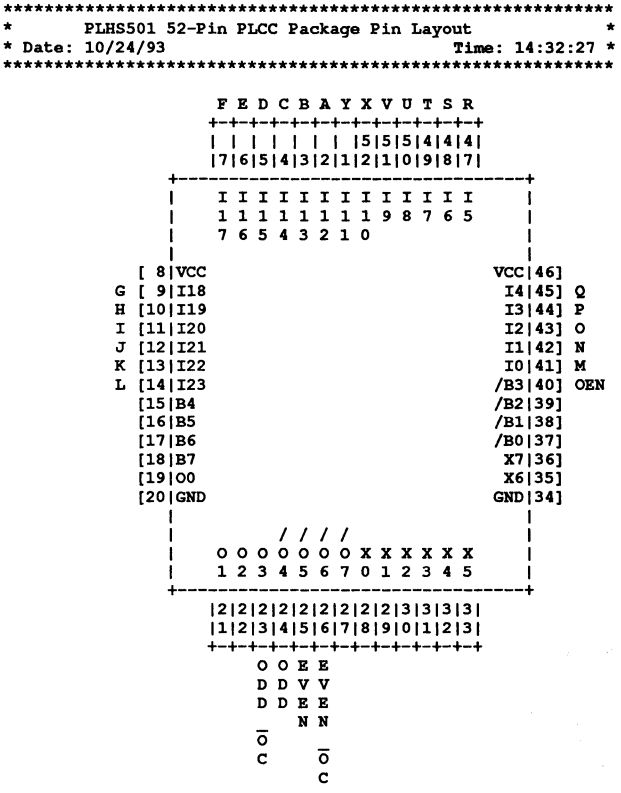


Figure 47. PARITET PLHS501 Pinlist

```
"FILENAME:PARITET.EQN
          24 BIT PARITY CIRCUIT"
@PINLIST
A       I;
B       I;
C       I;
D       I;
E       I;
F       I;
G       I;
H       I;
I       I;
J       I;
K       I;
L       I;
M       I;
N       I;
O       I;
P       I;
Q       I;
R       I;
S       I;
T       I;
U       I;
V       I;
X       I;
Y       I;
OEN     I;
ODD     O;
EVEN    O;
ODD_OC  O;
EVEN_OC O;

@LOGIC EQUATION

     "FIRST LEVEL: 'EVEN' FROM GROUPS OF THREE INPUTS"
   J0=/A*/B*/C + /A*B*C + A*/B*C + A*B*/C;
   J1=/D*/E*/F + /D*E*F + D*/E*F + D*E*/F;
   J2=/G*/H*/I + /G*H*I + G*/H*I + G*H*/I;
   J3=/J*/K*/L + /J*K*L + J*/K*L + J*K*/L;
   J4=/M*/N*/O + /M*N*O + M*/N*O + M*N*/O;
   J5=/P*/Q*/R + /P*Q*R + P*/Q*R + P*Q*/R;
   J6=/S*/T*/U + /S*T*U + S*/T*U + S*T*/U;
   J7=/V*/X*/Y + /V*X*Y + V*/X*Y + V*X*/Y;

     "SECOND LEVEL: 'EVEN' FROM FOUR GROUPS AT A TIME"
   J8=/J0*/J1*/J2*/J3 + /J0*/J1*J2*J3 + J0*J1*/J2*/J3 + /J0*J1*J2*/J3
      + J0*/J1*/J2*J3 + /J0*J1*/J2*J3 + J0*/J1*J2*/J3 + J0*J1*J2*J3;
   J9=/J4*/J5*/J6*/J7 + /J4*/J5*J6*J7 + J4*J5*/J6*/J7 + /J4*J5*J6*/J7
      + J4*/J5*/J6*J7 + /J4*J5*/J6*J7 + J4*/J5*J6*/J7 + J4*J5*J6*J7;

   T0=/(J8*J9);
   T1=/(/J8*/J9);
   T2=/(J8*/J9);
   T3=/(/J8*J9);

   ODDI=/(T2*T3);
   EVENI=/(T0*T1);
   ODD=ODDI;
   EVEN=EVENI;
   ODD.OE = /OEN;
   EVEN.OE = /OEN;
   ODD_OCI = 0;
   EVEN_OCI = 0;
   ODD_OC=ODD_OCI;
   EVEN_OC=EVEN_OCI;
   ODD_OC.OE = T2*T3*/OEN;
   EVEN_OC.OE = T0*T1*/OEN;
```

Figure 48. PARITET PLHS501 .BEE File

16-Bit Comparator

This example "compare", implements, a 16-bit comparator over 32 input bits. The design generates outputs for conditions representing the classic "EQUAL", "AGTB" (A>B) and BGTA (B>A). The long, triangularized equation for T42 suggests a clever editing approach to accurately enter a relatively long design equation into SNAP.

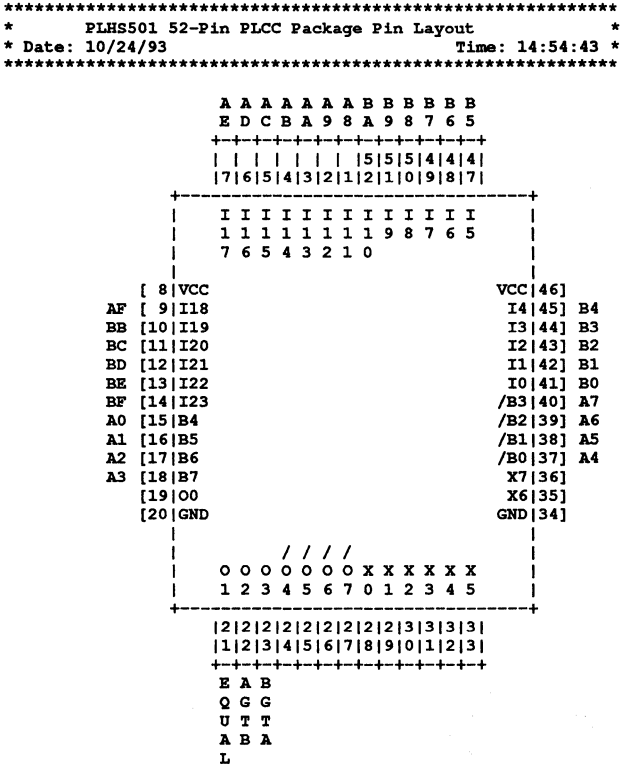


Figure 49. PLHS501 Pinlist for 16–Bit Comparator

```
"FILENAME:PARITET.EQN
         16 BIT COMPARATOR WITH THREE OUTPUTS:
         EQUAL,AGTB (A>B), AND BGTA (B>A)"

@PINLIST

A[9..0] I;
AA      I;   AD  I;
AB      I;   AE  I;
AC      I;   AF  I;

B[9..0] I;
BA      I;   BD  I;
BB      I;   BE  I;
BC      I;   BF  I;

EQUAL   O;
AGTB    O;
BGTA    O;

@LOGIC EQUATION
   T1=/(AF*/BF);     T2=/(/AF*BF);
   T3=/(AE*/BE);     T4=/(/AE*BE);
   T5=/(AD*/BD);     T6=/(/AD*BD);
   T7=/(AC*/BC);     T8=/(/AC*BC);
   T9=/(AB*/BB);     T10=/(/AB*BB);
   T11=/(AA*/BA);    T12=/(/AA*BA);
   T13=/(A9*/B9);    T14=/(/A9*B9);
   T15=/(A8*/B8);    T16=/(/A8*B8);
   T17=/(A7*/B7);    T18=/(/A7*B7);
   T19=/(A6*/B6);    T20=/(/A6*B6);
   T21=/(A5*/B5);    T22=/(/A5*B5);
   T23=/(A4*/B4);    T24=/(/A4*B4);
   T25=/(A3*/B3);    T26=/(/A3*B3);
   T27=/(A2*/B2);    T28=/(/A2*B2);
   T29=/(A1*/B2);    T30=/(/A1*B1);
   T31=/(A0*/B0);    T32=/(/A0*B0);

   T41=T1*T2*T3*T4*T5*T6*T7*T8*T9*T10*T11*T12*T13*T14*T15*T16*T17*
     T18*T19*T20*T21*T22*T23*T24*T25*T26*T27*T28*T29*T30*T31*T32;
   T42=                                                          /T1+
                                                               /T3*T2+
                                                             /T5*T4*T2+
                                                           /T7*T6*T4*T2+
                                                         /T9*T8*T6*T4*T2+
                                                      /T11*T10*T8*T6*T4*T2+
                                                   /T13*T12*T10*T8*T6*T4*T2+
                                                /T15*T14*T12*T10*T8*T6*T4*T2+
                                             /T17*T16*T14*T12*T10*T8*T6*T4*T2+
                                          /T19*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                                       /T21*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                                    /T23*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                                 /T25*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                              /T27*T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
                           /T29*T28*T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2+
      /T31*T30*T28 *T26*T24*T22*T20*T18*T16*T14*T12*T10*T8*T6*T4*T2;

   EQUAL=T41;
   AGTB=T42;
   BGTA=/(T41+T42);
```

Figure 50. Compare PLHS501 .BEE File

I²C I/O ports AN038

INTRODUCTION

This application note describes how to implement expansion I/O ports for a microcontroller via the I²C bus using a Philips Semiconductors PML2552 programmable logic device. This design provides 24 dedicated inputs and 16 outputs using only a single PML2552. Sixteen of the inputs may be configured on the bit level to be registered or direct inputs. The remaining eight are fixed as direct inputs.

I²C BUS

The I²C bus is a popular two-wire serial bus developed by Philips Semiconductors for communication between one or more microcontrollers and peripheral devices within a system. Although not as fast as parallel buses, it is designed to reduce packaging and board layout costs by requiring less wiring and fewer device pins to interconnect ICs or modules. Another feature is that additional circuitry may be easily plugged onto the two wires for testing or expansion purposes.

A complete specification for the I²C bus may be found in Philips' Microcontroller Data Handbook. A brief overview of the bus follows.

CHARACTERISTICS OF THE I²C BUS

The I²C bus provides 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data may be initiated only when the bus is not busy.

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (Figure 2).

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P). Figure 3 shows start and stop conditions.

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices that are controlled by the master are "slaves".

Figure 4 shows a block diagram of a system configuration.

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of the eight bits is followed by one acknowledge bit. The master supplies the clock pulse for the acknowledge bit. An acknowledge is a LOW level on the SDA line during the acknowledge clock. Therefore the transmitter must leave the bus HIGH so the receiver may pull the SDA line LOW to acknowledge.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte (Figure 5) A master receiver must generate an acknowledge after the reception of each byte provided the master wants to continue receiving bytes. A master receiver must signal an end of data to the slave transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the slave transmitter must leave the data line HIGH to enable to master to generate a stop condition).

Data transfers follow the formats shown in Figure 6. After the START condition, the master sends the slave address. This address is 7 bits long, the eighth bit is a data direction bit (R/W). A 'zero' indicates a transmission (WRITE) and a 'one" indicates a request for data (READ). A master always terminates a data transfer by a STOP condition. However, if a master still wishes to communicate on the bus, it can generate another START condition and address another slave without first generating a STOP condition. Various combinations of read and write formats are possible within such a transfer.

Each port in this design is assigned a unique address, so only 8 bits are available to or from each address. However, a master may read or write a port multiple times during the same message. During a read of a port (slave), if the master acknowledges the data, then the same port will put data on the bus again. When the master not-acknowledges the data, then the slave will release the bus so the master may generate a stop or another start condition. During a write, the master may simply continue to write data after the address. All data will go into and be acknowledged by the addressed slave output port.

PML2552 OVERVIEW

A functional block diagram of the Philips PML2552 is shown is Figure 1. The PML2552 is a CMOS device built on an EPROM process. This device contains 16 bypassable input D flip-flops, 16 bypassable output D flip-flops, 20 internal JK flip-flops, and 96 foldback NAND gates in a 68 pin package. The core of the PML2552 is a programmable array of 96 NAND gates and 20 buried JK flip-flops. The output of each NAND gate folds back upon itself and all other NAND gates and flip-flops. The Q and /Q output of each flip-flop folds back in a similar manner. All inputs (true and complement) also connect to each of the NAND gate and flip-flop inputs. Thus, total connectivity of all logic functions is achieved.

A NAND operation is functionally complete meaning that using only NAND gates, AND, OR and INVERT operations may be realized. Any logic function may therefore be constructed from the NAND gates in the folded NAND array, from simple combinatorial logic to additional edge-triggered registers. Because the NAND gates have very wide inputs (258 possible input connections), some functions which require many logic levels using simple 2 or 4 input gates, may be 'flattened' or implemented in fewer delay inducing logic levels.

I²C I/O PORT DESIGN

Shown in Figure 7 is a block diagram of the I²C I/O port design for the PML2552. It is set up as an I²C slave device to provide three 8 bit input ports and two 8 bit output registers. Addresses are programmed into the device and may be altered by changing 3 lines in the equations file. Two of the 8 bit input ports may be configured to be registered or direct inputs. These inputs are mapped to the 16 input D flip-flops of the PML2552 which may be individually bypassed. The third 8 bit port uses 8 direct input pins. The dual 8 bit output registers are mapped to the 16 output D flip-flops. An external connection is necessary to clock the output D flip-flops.

An 8-bit shift register was constructed from 8 internal JK flip-flops. Addresses and data appearing on the I²C bus are clocked into this register. A state machine looks at this register for a valid address, and if appropriate, clocks the data following the address into the output registers.

In addition to the shift register the state machine also looks at a three bit counter output to determine when a specific bit or data bytes are available. Using a counter to mark time instead of adding states to the state machine can save device resources for other functions. A counter may be constucted very efficiently using JK flip-flops and this one uses a unique design which uses only three NAND gates in the PML2552 flip-flop clock array, none in the foldback NAND array! The state machine is built using 7 of the PML2552's internal JK flip-flops. The state machine has an output, DSBLCNT, which holds the counter at zero for one clock cycle during the acknowledge bit time. DSBLCNT is also gated with a couple other signals to generate an acknowledge on the bus.

Two signals are involved in reseting or initializing the state machine and counter. They are called START and BUSY. Their relative timing is shown in Figure 8. The derivation of these signals is described later, but for now note that the counter and state machine are reset whenever START is HIGH or BUSY is low. START and BUSY are generated from a circuit which detects the I²C bus start and stop conditions. If an address is placed on the I²C bus that does not match one programmed into the device, the state machine will go into a state which can only be exited by the START or BUSY signals resetting the state machine. Also, if a bus master issues a repeated start condition, START will go HIGH initializing the state machine to begin looking for an address match.

Instead of loading the input register data into a shift register, a 24 to 1 multiplexer, along with the counter, select the proper input bit to send out onto the I²C bus. An additional flip-flop (U11) inside of the interface logic synchronizes the data to the I²C clock, SCL. The multiplexer inputs are offset in relation to the counter value. It was necessary to offset the bits relative to the counter due to the one bit delay introduced by U11 and DSBLCNT holding the counter at zero for one bit time.

U11's output connects to another flop-flop called U10. U10 holds the data (port data or acknowledge bit) to be sent to the master device for one data bit time. The I²C data must change only when the SCL clock is LOW. The falling edge of SCL may be up to 300ns long. For this reason, U10 is clocked by a delayed falling edge of SCL. The delay is created by a bidirectional output pin (DLYOUT) and an external RC network. Pin DLYOUT is configured to operate like an open-collector output. The input to DLYOUT's output buffer is connected to ground. The SCL input is internally connected to the three state output enable control line of this output buffer. DLYOUT should be externally connected to a 10K pull-up resistor and a 33pF capacitor connected to ground. An input buffer is internally connected to DLYOUT and clocks U10. U10's output controls the 3-State control line of output SDA. Output SDA behaves like an open-collector as the input to the 3-State output buffer is connected to ground.

Detecting valid I²C start and stop conditions was done without using internal flip-flops. Instead, an asychronous state machine was designed to output a pulse during start conditions and hold a line (busy) high until a stop condition. The start pulse initializes (resets) the state machine and counter.

START/STOP DETECTION DETAILS

The waveforms produced by the detection circuitry are shown in Figure 8. The first step to achive these waveforms is to draw waveforms of the desired circuit operation showing all possible input transitions. First, let's start with the START signal waveforms. These timing waveforms are shown in Figure 9a.

Next, a primitive flow table should be constructed. A primitive flow table has only a single stable state on each row. It is shown in Figure 9b. From the primitive flow table, a reduced flow table can be constructed by merging rows. Two rows may be merged, if, when comparing each column entry, they are the same or at least one is a Don't Care. Merging of rows may be described by a merger diagram. For this case one is shown in Figure 9c. So, from the primitive diagram, rows with stable states 0,1 and 2 are combined. Also, rows with stable states 6 and 3 are combined.

The merged flow table (Figure 9d) has two rows so only one state variable (Y_O) is required. The output matrix table is shown in Figure 9e. It was generated by simply assigning the output value associated with each stable state. Unstable entries were assigned the output value associated with the corresponding stable state.

A table showing the next state value for Y_O is shown in Figure 9f. It was generated by noting which states should remain stable and what value Y must be to either enter or remain in that state.

Generation of the BUSY signal required two state variables as shown in Figure 10. Tables were constructed in a manner similar to the process described above. However when transitioning from unstable state 1 to stable state 1 (see reduced flow table, stable state 1 is circled) an intermediate state was added. This was done to avoid switching the two transistion variables, y1 & y0, from 00 to 11 simultaneously. Adding the intermediate state makes the variables change from 00 to 10 to 11. Additionally, an extra state was added to transistion from unstable state 3 to stable state 3.

Waveforms for bus transations along with some internal signals are shown in Figures 11 and 12. The SNAP listing is shown in Figure 14.

CONCLUSION

PLDs may be used in embedded systems for more than simple decoding and glue logic collection functions. This design combined several low level functions (counter, shift register, state machine, random logic) to produce a useful microcontroller peripheral using one programmable logic device. This design may be easily altered to provide for specific system requirements. It has already been altered to make a simple I²C bus activity monitor which is the subject of another application note (AN039)!

A breadboard of this circuit was constructed for functional testing purposes. A listing for the design is shown in Figure 14. Additional circuit details are written as comments in the listing. A copy may be downloaded from the Philips PLD and Microcontroller Bulletin Board. The phone number is (800) 451-6644 or (408) 991-2406. Use a modem set to 8 bits, no parity and one stop bit.

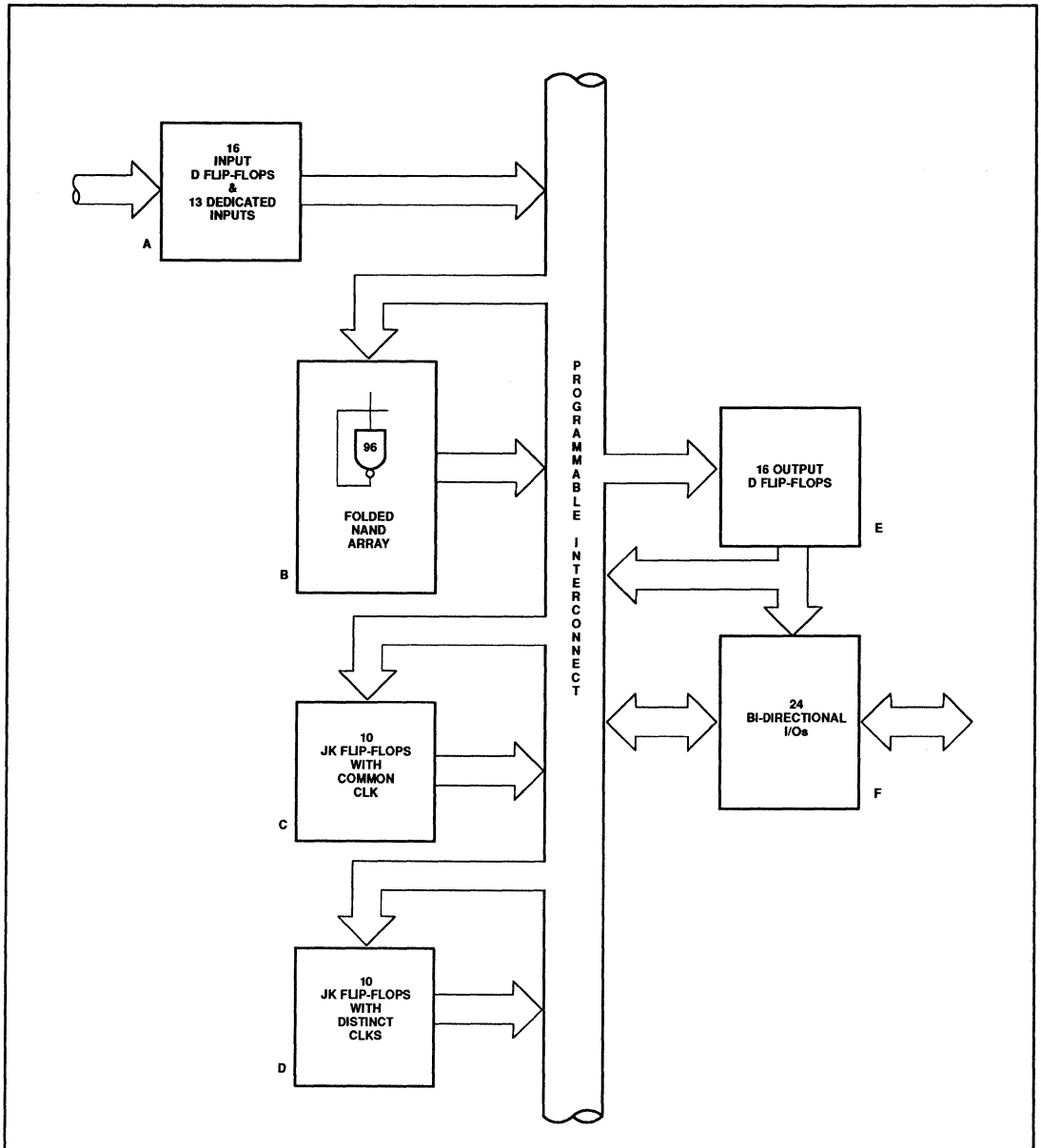


Figure 1. PML2552 Functional Block Diagram

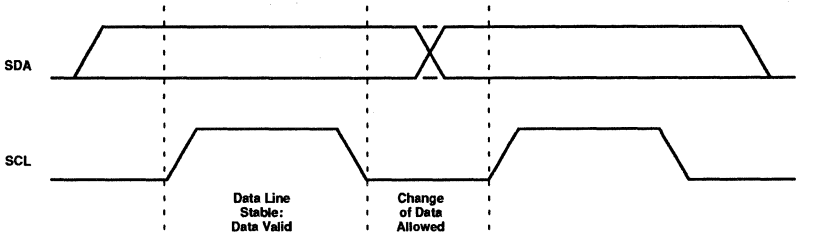


Figure 2. Bit Transfer

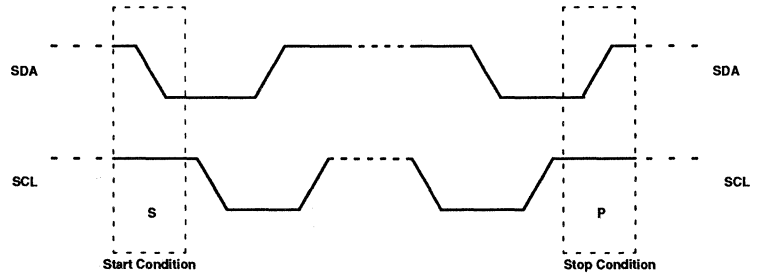


Figure 3. Definition of Start and Stop Conditions

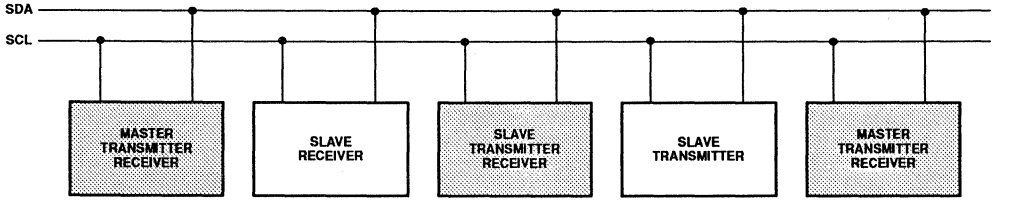


Figure 4. System Configuration Using I²C-Bus

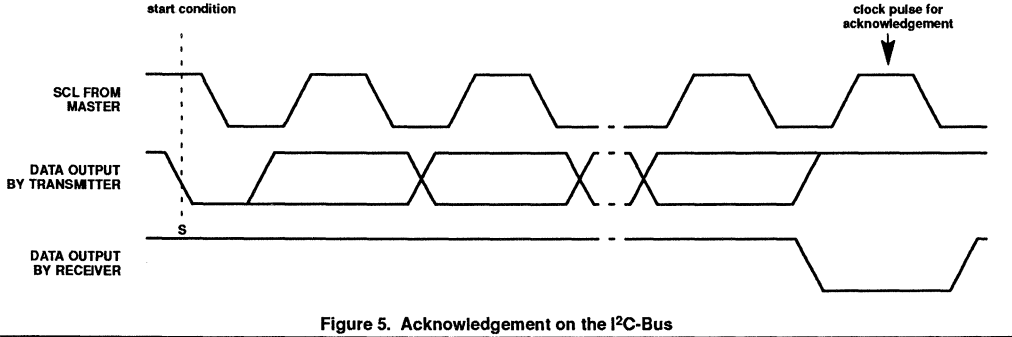
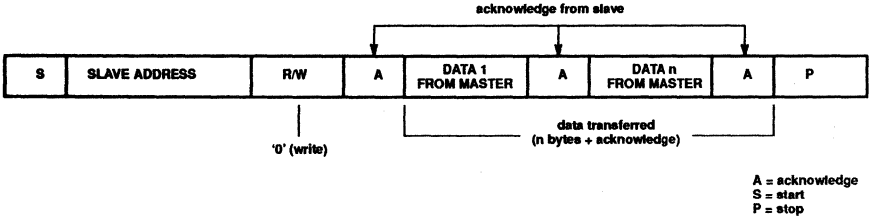


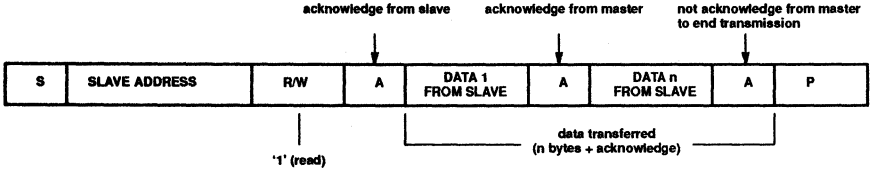
Figure 5. Acknowledgement on the I²C-Bus

Possible data transfer formats are:

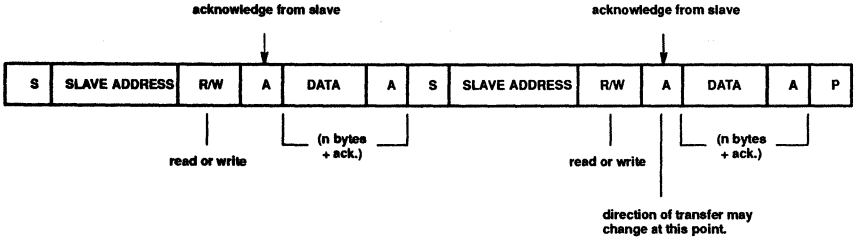
a. Master transmitter to slave receiver. Direction is not changed.



b. Master reads slave immediately after first byte.



c. Combined format.



During a change of direction within a transfer, the START condition and the slave address are both repeated, but the R/W bit reversed. Start, stop, slave addresses and R/W bits are generated by the master.

Figure 6. Data formats of the I²C-Bus

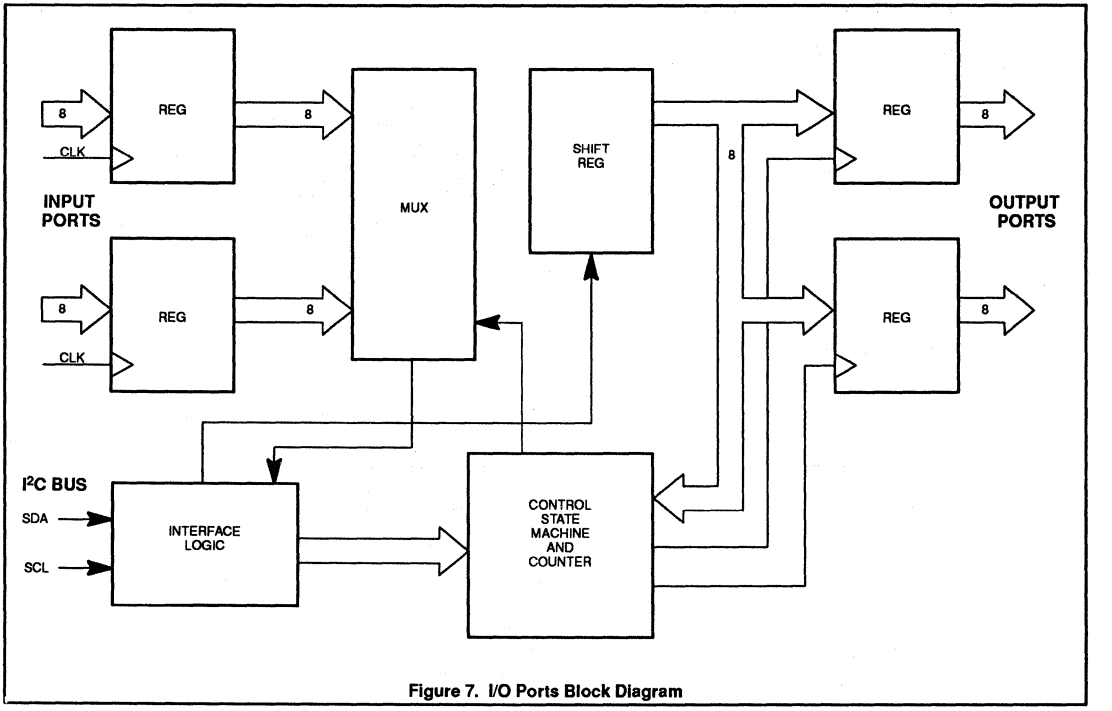


Figure 7. I/O Ports Block Diagram

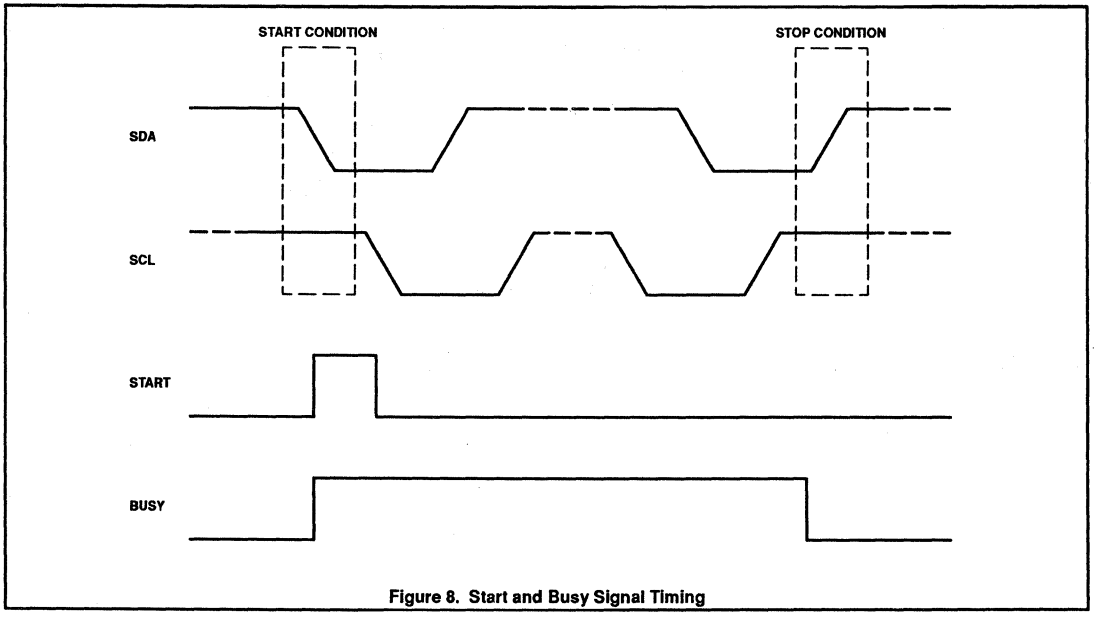
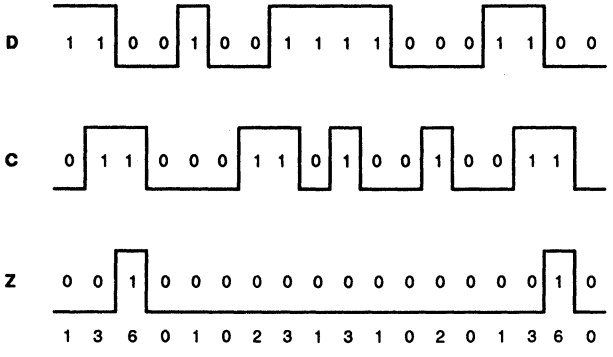
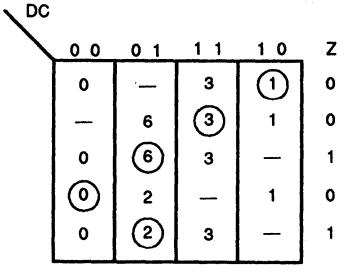


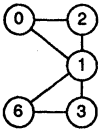
Figure 8. Start and Busy Signal Timing



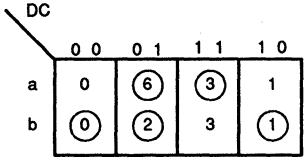
9a. Waveforms with Assigned State Values



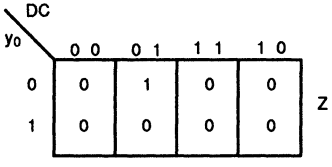
9b. Primitive Flow Table



9c. Merger Diagram

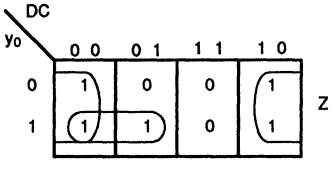


9d. Reduced Flow Table



$START = \overline{D} \cdot C \cdot \overline{y_0}$

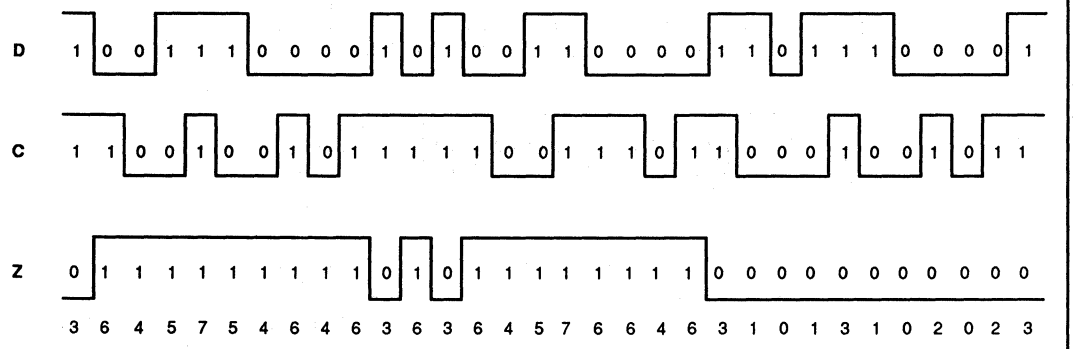
9e. Output Matrix

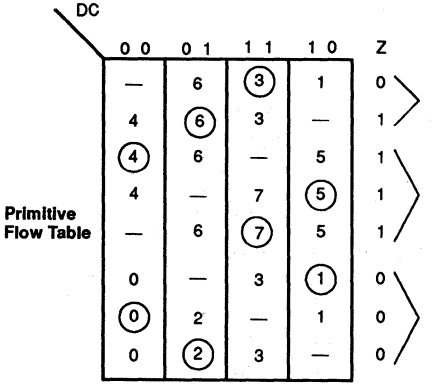


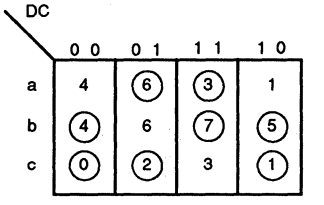
$y_0 = \overline{C} + \overline{D}y_0$

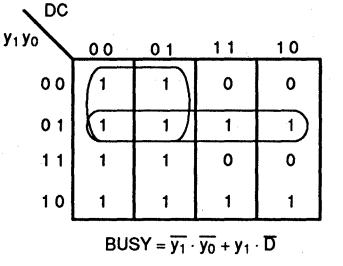
9f. Transition Matrix

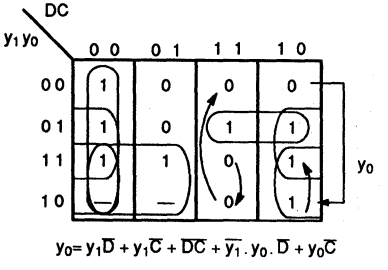
Figure 9. START Equation Generation











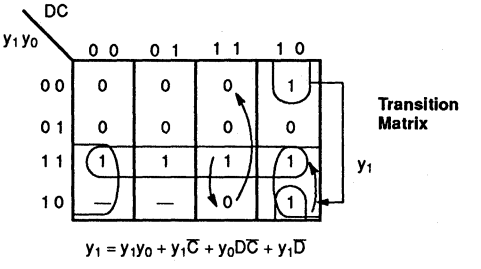


Figure 10. BUSY Signal Generation

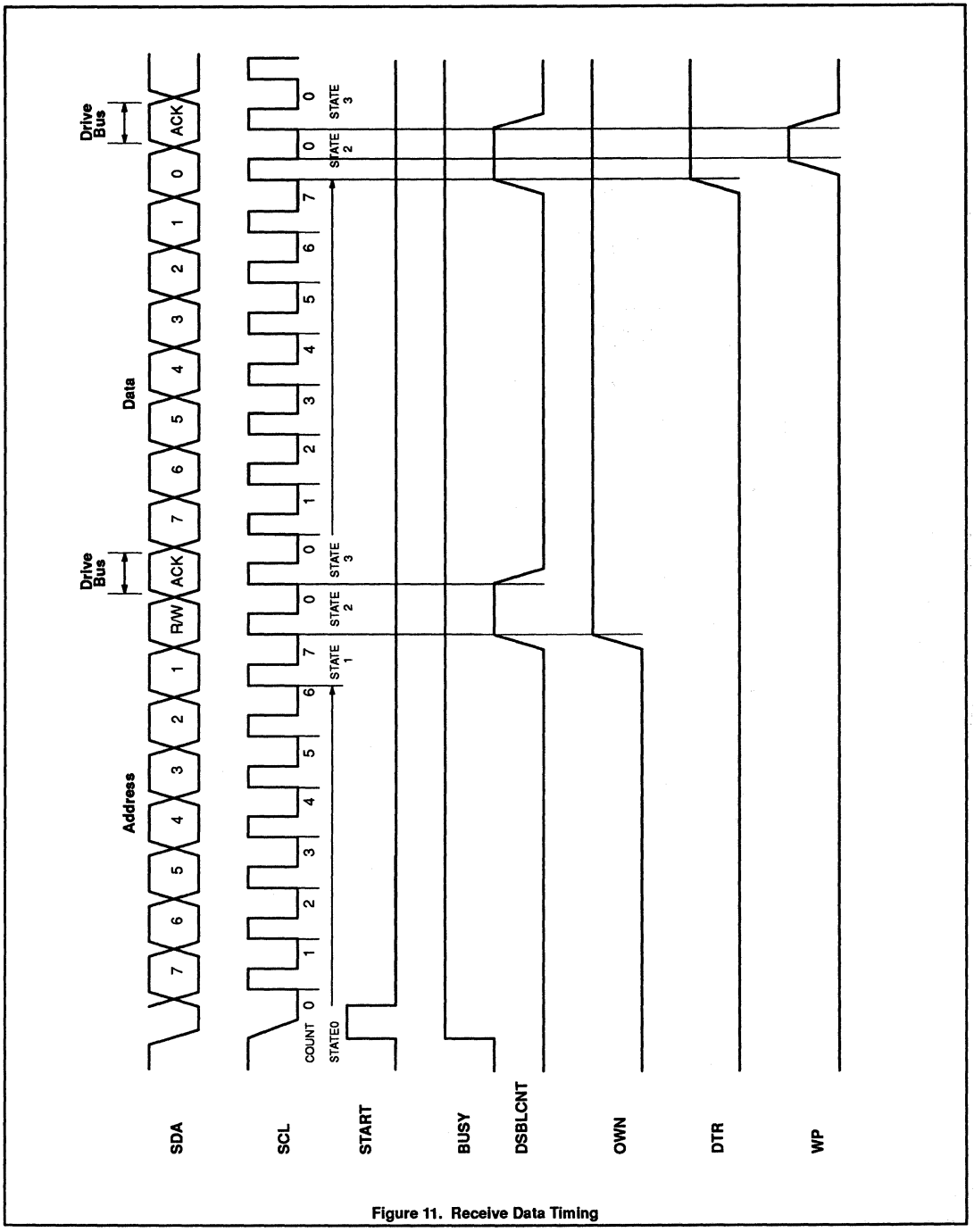


Figure 11. Receive Data Timing

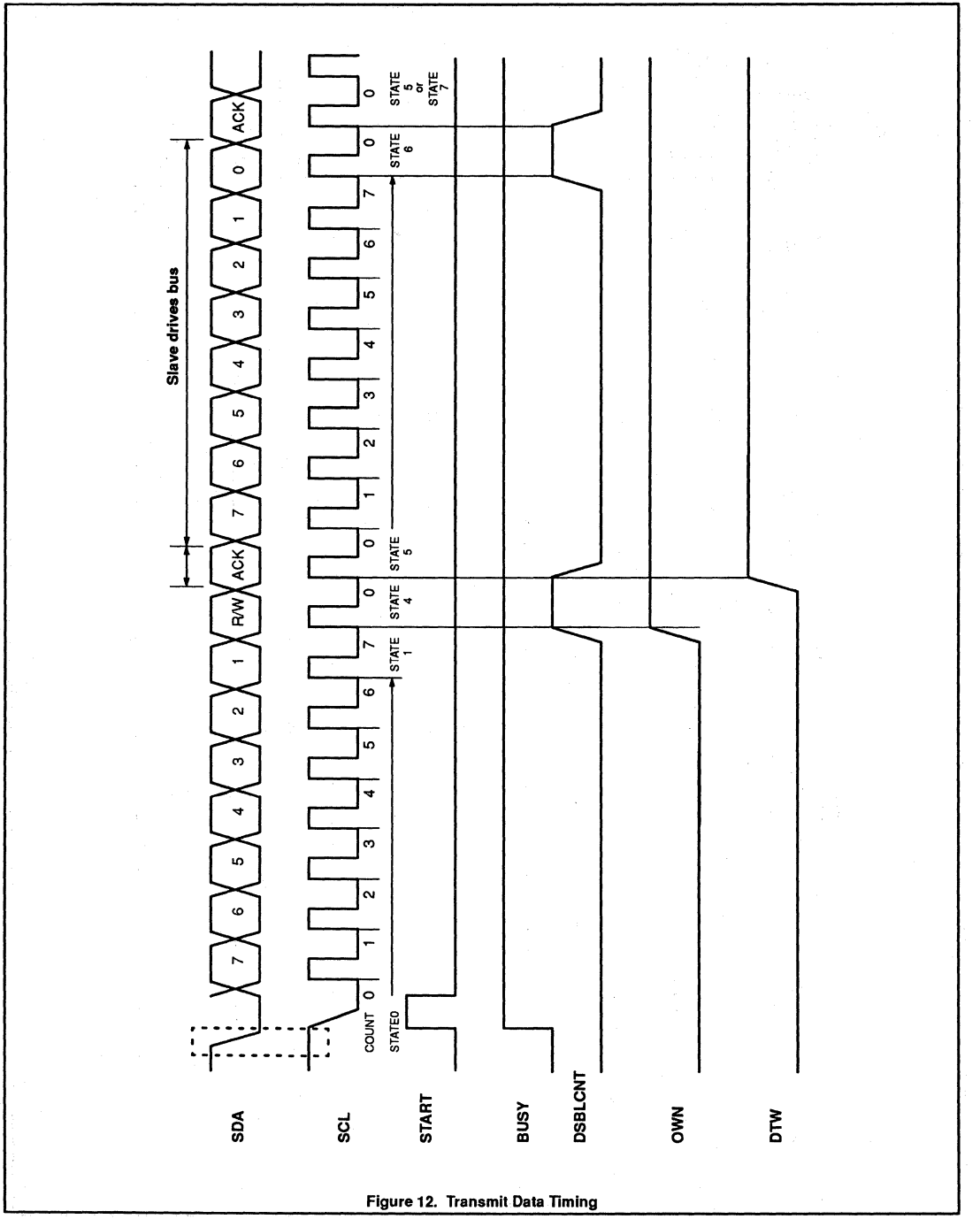


Figure 12. Transmit Data Timing

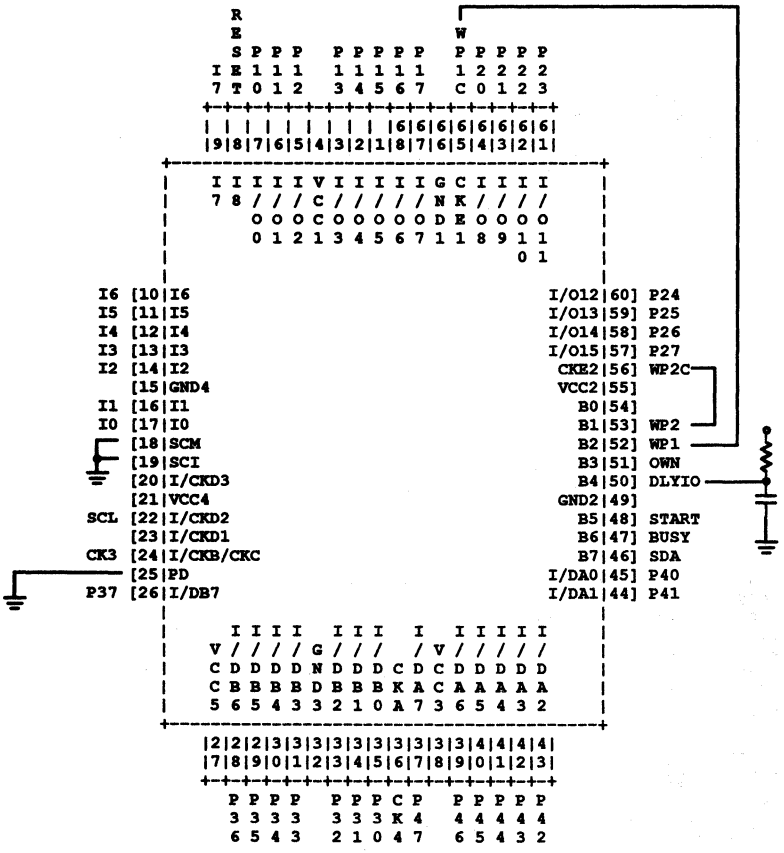


Figure 13. Pin Layout with External Connections

```
"
                    ------------------
                      I2C I/O Ports
                    ------------------
    This design is for a PML2552 device. It configures the PML2552
    to operate as several I2C slave remote I/O ports. It provides
    for 16 parallel output lines and 24 inputs. Specifically, the
    I/O ports are arranged as two 8-bit output ports, two 8-bit input
    ports (which may be modified on the bit level to be registered
    or direct inputs), and one 8-bit direct input port. The two 8-bit
    output ports will power-up to a HIGH level on the device pins.

    SNAP 1.90 or later must be used to compile this file.

"

@PINLIST

"input pins"
scl       i;    "I2C clock"
reset     i;    "reset input is active LOW"
p4[7..0]  i;    "port 4"
p3[7..0]  i;    "port 3"
ck3       i;    "clock for port 3"
ck4       i;    "clock for port 4"
wp1c      i;    "connect pin to wp1 pin"
wp2c      i;    "connect pin to wp2 pin"
i[7..0]   i;    "port 5"

"output pins"
sda       b;    "I2C data"
start     b;    "pulse during start condition"
busy      b;    "bus busy - high from start to stop"
own       o;    "address compare"
wp1       o;    "connect pin to wp1c pin"
wp2       o;    "connect pin to wp2c pin"
dlyio     b;    "connect pin to 10K pullup and 33pf cap to gnd"
p2[0..7]  o;    "port 2"
p1[0..7]  o;    "port 1"

@GROUPS
adr   = sr[6..0];
cx    = c[2..0];
ct    = c[2..0];
port5 = i[7..0];     "direct inputs"
port4 = p4[7..0];    "registered or direct inputs"
port3 = p3[7..0];    "registered or direct inputs"
port2 = p2[7..0];    "outputs"
port1 = p1[7..0];    "outputs"

@TRUTHTABLE
@LOGIC EQUATIONS

"------------------"
" Input registers "
"------------------"
"
d[15..8].id  = port4;
d[15..8].clk = ck4;
d[7..0].id   = port3;
d[7..0].clk  = ck3;
The breadboard doesn't use input registers
so the above section is commented out and
the following section used.
"
```

Figure 14. (1 of 5)

```
"------------"
" Input pins "
"------------"

d[15..8] = port4;
d[7..0]  = port3;

di[7..0] = port5;

"------------------"
" Output registers "
"------------------"

port2.od = sr[7..0];
port2.clk = wp2c;
port1.od = sr[7..0];
port1.clk = wp1c;

"---------------------------"
" Shift register equations "
"---------------------------"

u2.clk = scl;                       "8 bit shift register for input"
u3.clk = scl;                       "of I2C address and data"
u4.clk = scl;
u5.clk = scl;                       "the state machine reads this register"
u6.clk = scl;                       "for address comparison and clocks"
u7.clk = scl;                       "data from this register into"
u8.clk = scl;                       "appropriate output register"
u9.clk = scl;
u2.set = /(start + /busy);          "this register is clocked by the I2C"
u3.set = /(start + /busy);          "bus clock"
u4.set = /(start + /busy);
u5.set = /(start + /busy);          "reset or set signals defined so SNAP"
u6.set = /(start + /busy);          "will use specific internal JK flip-flops"
u7.set = /(start + /busy);
u8.set = /(start + /busy);
u9.rst = /(start + /busy);
u2.j =  sda;
u2.k = /sda;
u3.j =  u2;
u3.k = /u2;
u4.j =  u3;
u4.k = /u3;
u5.j =  u4;
u5.k = /u4;
u6.j =  u5;
u6.k = /u5;
u7.j =  u6;
u7.k = /u6;
u8.j =  u7;
u8.k = /u7;
u9.j =  u8;
u9.k = /u8;

sr[0..7] = u[2..9];

"------------------"
" Counter section "
"------------------"

c2.j = 1;                           "3-bit binary counter"
c2.k = 1;                           "outputs connected to state machine"
c1.j = 1;                           "and to multiplexer "
c1.k = 1;
c0.j = 1;
c0.k = 1;
```

Figure 14. (2 of 5)

```
c0.rst = /(/busy + start);
c1.rst = /(/busy + start);
c2.rst = /(/busy + start);

c0.clk = /(/dsblcnt * /scl);                "counter uses NAND gates"
c1.clk = /(/dsblcnt * /scl * c0);           "in clock NAND array, not"
c2.clk = /(/dsblcnt * /scl * c0 * c1);      "from the foldback array"

"-----------------------"
" Multiplexer equations "
"-----------------------"

uuuOUT =                                      "convert parallel input data"
      (D6*OWN1*/OWN2*CT==0H*/dsblcnt)         "to serial format"
    + (D5*OWN1*/OWN2*CT==1H*/dsblcnt)
    + (D4*OWN1*/OWN2*CT==2H*/dsblcnt)
    + (D3*OWN1*/OWN2*CT==3H*/dsblcnt)          "port 3"
    + (D2*OWN1*/OWN2*CT==4H*/dsblcnt)
    + (D1*OWN1*/OWN2*CT==5H*/dsblcnt)
    + (D0*OWN1*/OWN2*CT==6H*/dsblcnt)
    + (D7*OWN1*/OWN2* dsblcnt)

    + (D14*OWN2*/OWN1*CT==0H*/dsblcnt)
    + (D13*OWN2*/OWN1*CT==1H*/dsblcnt)         "port 4"
    + (D12*OWN2*/OWN1*CT==2H*/dsblcnt)
    + (D11*OWN2*/OWN1*CT==3H*/dsblcnt)
    + (D10*OWN2*/OWN1*CT==4H*/dsblcnt)
    + (D9 *OWN2*/OWN1*CT==5H*/dsblcnt)
    + (D8 *OWN2*/OWN1*CT==6H*/dsblcnt)
    + (D15*OWN2*/OWN1* dsblcnt)

    + (di6*OWN1*OWN2*CT==0H*/dsblcnt)
    + (di5*OWN1*OWN2*CT==1H*/dsblcnt)          "port 5"
    + (di4*OWN1*OWN2*CT==2H*/dsblcnt)
    + (di3*OWN1*OWN2*CT==3H*/dsblcnt)
    + (di2*OWN1*OWN2*CT==4H*/dsblcnt)
    + (di1*OWN1*OWN2*CT==5H*/dsblcnt)
    + (di0*OWN1*OWN2*CT==6H*/dsblcnt)
    + (di7*OWN1*OWN2* dsblcnt);

"--------------------------------------"
" Detect I2C start & stop conditions "
"--------------------------------------"

uuuy     = /scl + uuuy*/sda;                    "start condition pulse"
uuustart = /uuuy*/sda*scl;
start    = uuustart;
start.oe = 1;
                                                "bus busy signal is HIGH"
uuuy0    = uuuy1*/sda + uuuy1*/scl              "from start to stop"
         + /sda*/scl  + /uuuy1*uuuy0*sda        "conditions"
         + uuuy0*/scl + /reset;

uuuy1    = uuuy1*uuuy0 + uuuy1*/scl
         + /uuuy0*sda*/scl + uuuy1*/sda + /reset;

uuubusy  = /uuuy1*uuuy0 + /uuuy1*/sda;
busy     = uuubusy;
busy.oe  = 1;
```

Figure 14. (3 of 5)

```
"----------------"
" Output control "
"----------------"

sda      = 0;                                "I2C data"
sda.oe   = u10;

u10in    = /dtw *  dsblcnt * own             "send ACK  on bus"
         +  dtw * /dsblcnt * own * /u11;     "send data on bus"
u10.j    = u10in;
u10.k    = /u10in;                      "u10 holds data stable on bus"
u10.clk  = dlyio;
u10.rst  = /(start + /busy);

u11.j    =  uuuout;              "u11 synchronizes input data to I2C clock"
u11.k    = /uuuout;
u11.clk  = scl;
u11.rst  = /(start + /busy);

dlyio    = 0;                 "pin dlyio should be connected to a 10K"
dlyio.oe = scl;               "pull-up and a 33pf cap to ground"

"------------------------"
" State machine portion "
"------------------------"

      "port address declarations"
      "these addresses may be changed to"
      "any valid I2C address"

adrp1 = adr==0111000b;      "this address is for port1 (write) & port3 (read)"
adrp2 = adr==0111010b;      "this address is for port2 (write) & port4 (read)"
adrp5 = adr==0111011b;      "this address is to read pins I0 to I7"

state2i =  S2*/S1*S0;               "duplication of some state machine"
state4i = /S2* S1*S0;               "states to be used in following"
state6i = /S2*/S1*S0;               "boolean equation"

dsblcnt = state2i+state4i+state6i;  "dsblcnt HIGH when in state 2,4 or 6"

s2.set    = /(start + /busy);       "reset state machine upon start"
s1.set    = /(start + /busy);       "condition and whenever bus is "
s0.set    = /(start + /busy);       "not busy (stop detected).     "
own1.rst  = /(start + /busy);
own2.rst  = /(start + /busy);       "own1 & own2 denote address match"
dtw.rst   = /(start + /busy);       "write to I2C bus (read port)"
dtr.rst   = /(start + /busy);       "read I2C bus     (write port)"
s2.clk    = scl;
s1.clk    = scl;                    "clock state machine from I2C clock"
s0.clk    = scl;
own1.clk  = scl;
own2.clk  = scl;
dtw.clk   = scl;
dtr.clk   = scl;

cntr7 = cx==7h;             "state machine reads counter to"
cntr6 = cx==6h;             "wait for specific bits"

wp = dtr * dsblcnt * u10;
wp2 = wp * own2;            "wp2 and wp1 clock data from shift register"
wp1 = wp * own1;            "into proper output port. external connection"
own = own1 + own2;          "from wp1 to wp1c and wp2 to wp2c is required"
```

Figure 14. (4 of 5)

```
@INPUT VECTORS
@OUTPUT VECTORS
[dtw,own1,own2] JKFFR
low = 000b;
[dtr] JKFFR
dtrl = 0b;
dtrh = 1b;

@STATE VECTORS
[s2,s1,s0] JKFFS
state0 = 111b;
state1 = 110b;
state2 = 101b;
state3 = 100b;
state4 = 011b;
state5 = 010b;
state6 = 001b;
state7 = 000b;      "bus start or stop condition required to leave state7"

@TRANSITIONS

while [state0]                                    "start detected, "
 if [cntr6] then [state1]                         "wait for address"

while [state1]                                   "compare address & r/w bit"
 if [adrp1* sda] then [state4] with [own1]       "addressed port 3 to read"
 if [adrp2* sda] then [state4] with [own2]       "addressed port 4 to read"
 if [adrp5* sda] then [state4] with [own1,own2] "addressed port 5 to read"
 if [adrp1*/sda] then [state2] with [own1]       "addressed port 1 to write"
 if [adrp2*/sda] then [state2] with [own2]       "addressed port 2 to write"
 if [/adrp1*/adrp2*/adrp5] then [state7]         "no address compare"
 if [adrp5*/sda] then [state7]                   "ignore port 5 for write"

while [state2]                            "write port"
 if [] then [state3]

while [state3]
 if [cntr7] then [state2] with [dtrh]          "wait for 7 bits"

while [state4]                            "read port"
 if [] then [state5] with [dtw]

while [state5]
 if [cntr7] then [state6]                      "wait for 7 bits"

while [state6]
 if [/sda] then [state5]             "master acknowledged so send byte again"
 if [ sda] then [state7] with [low]  "master not-acknowedged so release bus"
```

Figure 14. (5 of 5)

I²C bus monitor AN039

INTRODUCTION

The PML2552 device may be used to build a simple real time non-intrusive I²C bus monitor. This monitor displays all addresses and data sent on the bus as well as acknowledge and read/write bit conditions. It also outputs START and BUSY signals which may be used for synchronizing additional test equipment to the bus. The outputs of this monitor may be connected to a microcontroller for additional data formatting capabilities.

A block diagram of this design is shown in Figure 1. This design is a modification of an I²C I/O port application using the PML2552. The input registers and multiplexor functions were removed from the I²C I/O port and the control state machine was modified to latch all addresses and data bytes acknowledged on the bus. Additionally, since the monitor does not drive the I²C bus, the circuitry in the I/O port application that generated acknowledge bits during a read was also removed. Two signals, START and BUSY are generated by an asynchronous state machine. For detailed information on the derivation of these signals, the reader should refer to the PML2552 I²C I/O port application note (AN038).

SHIFT REGISTER

Referring to Figure 1 and Figure 3, an 8-bit shift register was constructed from 8 internal JK flip-flops. Addresses and data appearing on the I²C bus are clocked into this register. A state machine looks at this register, and clocks the byte into the appropriate output register for address or data display. The byte will only be clocked into an output register if an acknowledge of the address or data occurred.

In addition to the shift register the state machine also looks at a three bit counter output to determine when a specific bit or data bytes are available. Using a counter to mark time instead of adding states to the state machine can save device resources for other functions. A counter may be constructed very efficiently using JK flip-flops and this one uses a unique design which uses only three NAND gates in the PML2552 flip-flop clock array, none in the foldback NAND array! The state machine is built using 7 of the PML2552's internal JK flip-flops. The state machine has an output, DSBLCNT, which holds the counter at zero for one clock cycle during the acknowledge bit time. DSBLCNT is also gated with a couple other signals to generate the clock pulse that updates the output registers.

Two signals are involved in resetting or initializing the state machine and counter. They are called START and BUSY. These signals are generated from a circuit which detects the I²C bus start and stop conditions. Refer to Figure 2 to see the relative timing of these signals. If a bus master issues a repeated start condition, START will go HIGH initializing the state machine to begin looking for an address match.

STATE MACHINE DETAILS

After the BUSY line goes HIGH, the state machine will be in state 0. It simply waits in this state until the counter counts to six. The shift register should now contain the address sent on the bus.

Upon the next I²C clock, the state machine to transitions to state 1. When the next I²C clock occurs, the RW bit status will be transferred to a register and an output pin. In addition, the RW bit will also be shifted into the shift register to be displayed along with the address when the shift register is clocked into the address display output port. The state machine will transition to state 2.

Being in state 2 causes a signal DSBLCNT to go HIGH. This signal disables the counter for one count but also enables WP1 to clock the address and data into the display port upon the falling edge of the I²C clock.

After the address is found and latched into the output port, the state machine will transition to state 3. While in state 3 the state machine waits for the counter to count to seven, indicating that a byte of data is in the shift register. The state machine now transitions back to state 2 but sets a control bit named DATA. Once set, this bit can only be reset by another bus START condition or by a bus STOP condition. When the acknowledge clock pulse occurs, the byte in the shift register will be transferred to the data display output port.

The state machine continues in a like manner transitioning between state 2 and state 3, latching data into the data display output port. When a bus STOP condition occurs, the state machine and counter circuitry will be reset, and will be held in the reset state until a bus START condition occurs.

CONCLUSION

This design combined several low level functions (counter, shift register, state machine, random logic) to produce a useful function using one programmable logic device. Although a microcontroller may be programmed to read addresses and data on the I²C bus, as the speed of the I²C bus is increased to 400 Kbits/sec the microcontroller will have difficulty doing anything else. Moving some high speed functions to hardware makes sense, allowing the microcontroller time to analyze and format the data . This design may be easily altered to provide for specific system requirements.

A breadboard of this circuit was constructed for functional testing purposes. A listing for the design is shown in Figure 3. Additional circuit details are written as comments in the listing. A copy may be downloaded from the Philips PLD and Microcontroller Bulletin Board. The phone number is (800) 451-6644 or (408) 991-2406. Use a modem set to 8 bits, no parity and one stop bit.

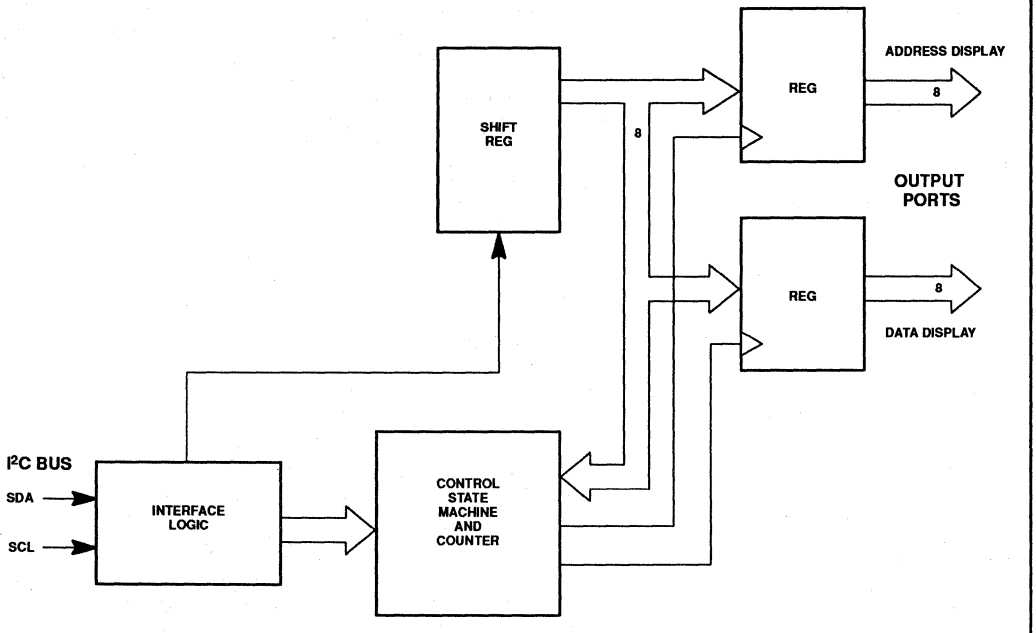


Figure 1. Monitor Block Diagram

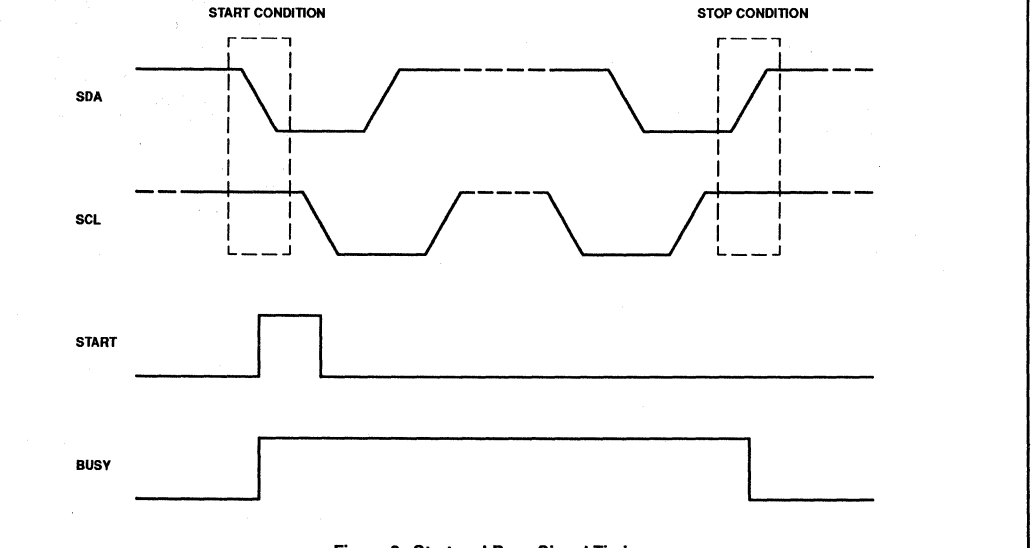


Figure 2. Start and Busy Signal Timing

```
"                    --------------------
                       I²C Bus Monitor
                     --------------------
  This design is for a PML2552 device. It configures the PML2552 to work
  as a simple real time non-intrusive I2C bus monitor. When connected to
  the I2C bus, one 8-bit output port displays addresses sent on the bus
  while another 8-bit port displays data. Additional output pins display
  address acknowledge, data acknowledge, rw mode, bus start condition,
  and bus busy status.

  SNAP 1.90 or later must be used to compile this file.
"

@PINLIST
"inputs"
scl       i;  "I2C clock"
sda       i;  "I2C data"
reset     i;  "reset input active LOW"
wp1c      i;  "connect this pin to wp1 pin "
wp2c      i;  "connect this pin to wp2 pin "

"outputs"
start     b;  "start pulse"
busy      b;  "bus busy"
aack      o;  "address acknowledge"
dack      o;  "data acknowledge"
rwo       o;  "read/write"
wp1       o;  "connect this pin to wp1c pin"
wp2       o;  "connect this pin to wp2c pin"
p2[0..7]  o;  "data display port"
p1[0..7]  o;  "address display port"

@GROUPS
adr   = sr[6..0];
cx    = c[2..0];
ct    = c[2..0];
port2 = p2[7..0]; "data display port"
port1 = p1[7..0]; "address display port"

@TRUTHTABLE
@LOGIC EQUATIONS

"------------------"
" Output registers "
"------------------"

port2.od  = sr[7..0];
port2.clk = wp2c;
port1.od  = sr[7..0];
port1.clk = wp1c;

"---------------------------"
" Shift register equations "
"---------------------------"

u2.clk = scl;
u3.clk = scl;                    "8 bit shift register for input"
u4.clk = scl;                    "of I2C addresses and data"
u5.clk = scl;
u6.clk = scl;                    "the state machine will clock address"
u7.clk = scl;                    "or data bits from this register into"
u8.clk = scl;                    "the appropriate output register after"
u9.clk = scl;                    "eight bits have been clocked in"
u2.set = /(start + /busy);
u3.set = /(start + /busy);       "this register is clocked by the I2C"
u4.set = /(start + /busy);       "bus clock"
u5.set = /(start + /busy);
u6.set = /(start + /busy);       "reset or set signals are defined so SNAP"
u7.set = /(start + /busy);       "will use specific internal JK flip-flop"
u8.set = /(start + /busy);
u9.rst = /(start + /busy);
```

Figure 3. (1 of 3)

```
u2.j =  sda;
u2.k = /sda;
u3.j =  u2;
u3.k = /u2;
u4.j =  u3;
u4.k = /u3;
u5.j =  u4;
u5.k = /u4;
u6.j =  u5;
u6.k = /u5;
u7.j =  u6;
u7.k = /u6;
u8.j =  u7;
u8.k = /u7;
u9.j =  u8;
u9.k = /u8;

sr[0..7] = u[2..9];

"-----------------"
" Counter section "
"-----------------"

c2.j = 1;               "3-bit binary counter"
c2.k = 1;               "outputs are connected to state machine"
c1.j = 1;
c1.k = 1;
c0.j = 1;
c0.k = 1;

c0.rst = /(/busy + start);
c1.rst = /(/busy + start);
c2.rst = /(/busy + start);

c0.clk = /(/dsblcnt * /scl);                "counter uses NAND gates in"
c1.clk = /(/dsblcnt * /scl * c0);           "clock NAND array, not in"
c2.clk = /(/dsblcnt * /scl * c0 * c1);      "foldback array"

"-------------------------------------"
" Detect IIC start & stop conditions "
"-------------------------------------"

uuuy     = /scl + uuuy*/sda;                "start condition pulse"
uuustart = /uuuy*/sda*scl;
start    = uuustart;
start.oe = 1;

uuuy0    = uuuy1*/sda + uuuy1*/scl          "bus busy signal is HIGH"
         + /sda*/scl  + /uuuy1*uuuy0*sda    "from start to stop conditions"
         + uuuy0*/scl + /reset;
uuuy1    = uuuy1*uuuy0 + uuuy1*/scl
         + /uuuy0*sda*/scl + uuuy1*/sda + /reset;

uuubusy  = /uuuy1*uuuy0 + /uuuy1*/sda;
busy     = uuubusy;
busy.oe  = 1;

"------------------------"
" State machine portion "
"------------------------"

state2i =  S2*/S1*S0;       "duplication of state definition to be used"
                            "in following equation"

dsblcnt = state2i;          "dsblcnt HIGH when in state2"

s2.set   = /(start + /busy);     "reset state machine upon start"
s1.set   = /(start + /busy);     "condition and whenever bus is"
s0.set   = /(start + /busy);     "not busy (stop detected)."
data.rst = /(start + /busy);
rw.rst   = reset;                "rw displays read or write bus operation"
acka.rst = reset;                "acknowledge address"
ackd.rst = reset;                "acknowledge data"
```

Figure 3. (2 of 3)

```
s2.clk   = scl;                    "clock state machine from I2C bus clock"
s1.clk   = scl;
s0.clk   = scl;
data.clk = scl;
rw.clk   = scl;
acka.clk = scl;
ackd.clk = scl;
                                   "state maachine reads counter to wait"
cntr7 = cx==7h;                    "for specific bits"
cntr6 = cx==6h;

wp2 = dsblcnt * /scl *  data;      "write data to port2"
wp1 = dsblcnt * /scl * /data;      "write address and rw bit to port1"

rwo = rw;                          "pins mirror values of internal flops"
aack=acka;
dack=ackd;

@INPUT VECTORS
@OUTPUT VECTORS
[rw,data,acka,ackd]JKFFR
low = 0000b;

@STATE VECTORS
[s2,s1,s0] JKFFS
state0 = 111b;
state1 = 110b;
state2 = 101b;
state3 = 100b;

@TRANSITIONS
while [state0]                                    "start detected"
 if [cntr6] then [state1]                         "wait for address"

while [state1]                                    "check polarity of rw bit"
 if [sda]  then [state2] with [rw]                "addressed port to read"
 if [/sda] then [state2] with [/rw]               "addressed port to write"

while [state2]                                    "display port data"
 if [ sda*/data] then [state3] with [ acka]       "display address ack"
 if [/sda*/data] then [state3] with [/acka]
 if [ sda* data] then [state3] with [ ackd]       "or data acknowledge"
 if [/sda* data] then [state3] with [/ackd]

while [state3]
 if [cntr7] then [state2] with [data]             "wait for 7 bits"
```

Figure 3. (3 of 3)

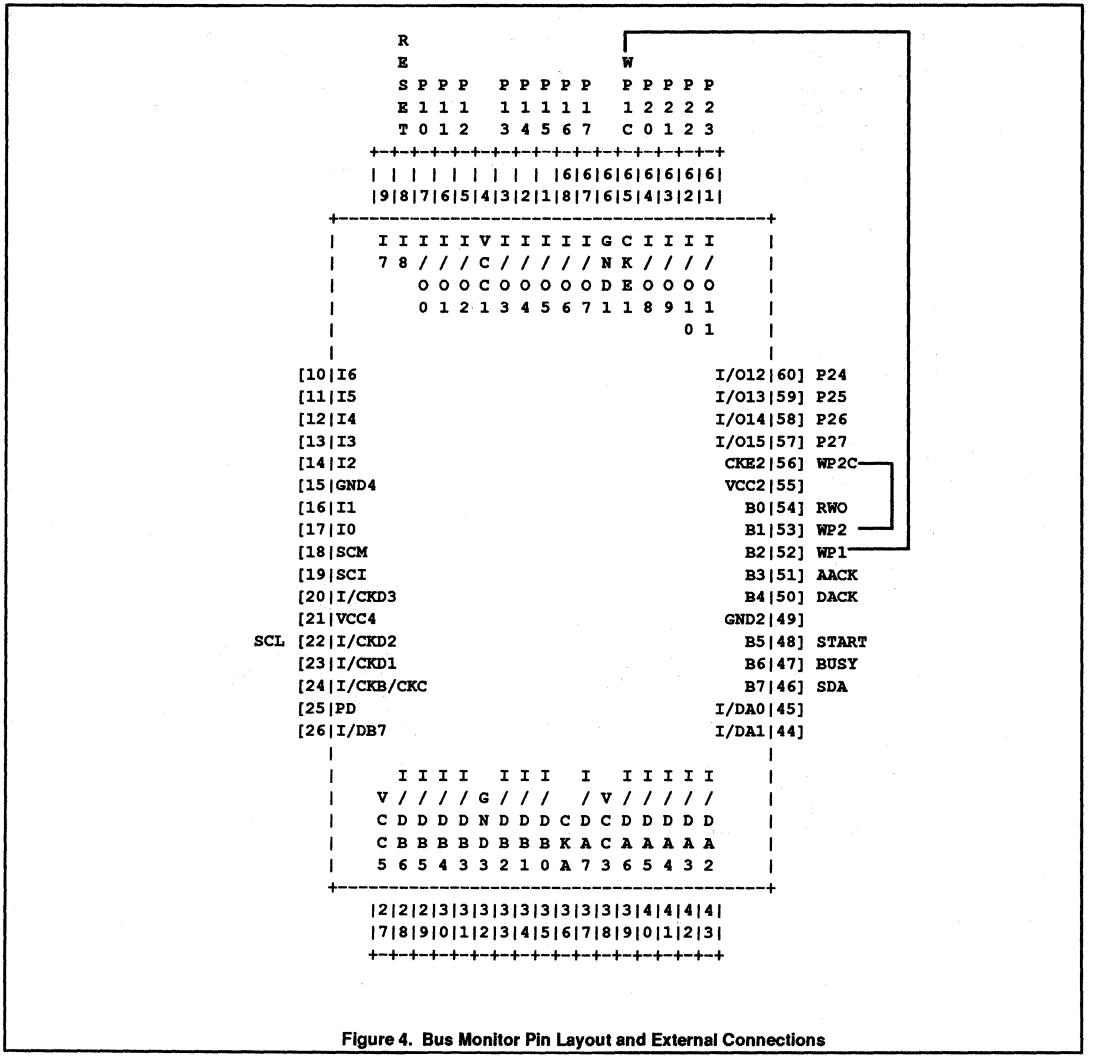


Figure 4. Bus Monitor Pin Layout and External Connections

Switching control unit for data communication via RS232 AN040

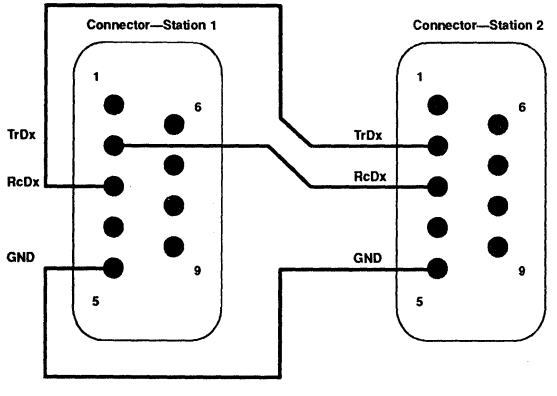
INTRODUCTION

In many laboratories and institutions one can still find many PCs or similar computer equipment which are not connected via a network. There is often a need for occasional communication and data exchange in these cases and this application note describes a inexpensive solution, allowing intercommunications of up to eight stations.

The specific goal of this design was to connect six PCs and two PLD-Programmers in such a way that any two of them could communicate. Each of these devices has a serial interface (RS 232), which may be connected to another device in a null modem configuration by connecting theTrD (Transmit Data) and RcD (Receive Date) lines of the two selected stations. With the addition of readily available software for the PCs that handshakes using a XON/XOFF protocol, the design realizes a digital switching network.

The main part of the developed design represents a MUX-DEMUX-circuit realizing the primary switching network. One Multiplexer for each direction switches the TrD line of an activated station to an internal crosspoint and a Demultiplexer connects it to the RcD wire of the corresponding station. The choice of the stations to be connected is set up by the user via mechanical switches, so that an additional priority encoder has to guarantee the activation of only two stations at a time. In order to display the actual status of the switching network, two decoders are added which can drive 7-Segment-Displays directly and indicate the actual connection.

Figure 1 shows the basic interconnection to be realized by the switching network and the structure of the complete design is shown in Figure 2. Beside the Line Buffers (DS 232) the complete design was to be implemented using just one programmable logic component. A Philips Semiconductor's PLHS501 proved to be an excellent choice for this design.



Nullmode, XON/XOFF protocol

Figure 1. Interconnection via RS232 for a Simple XON/XOFF Protocol

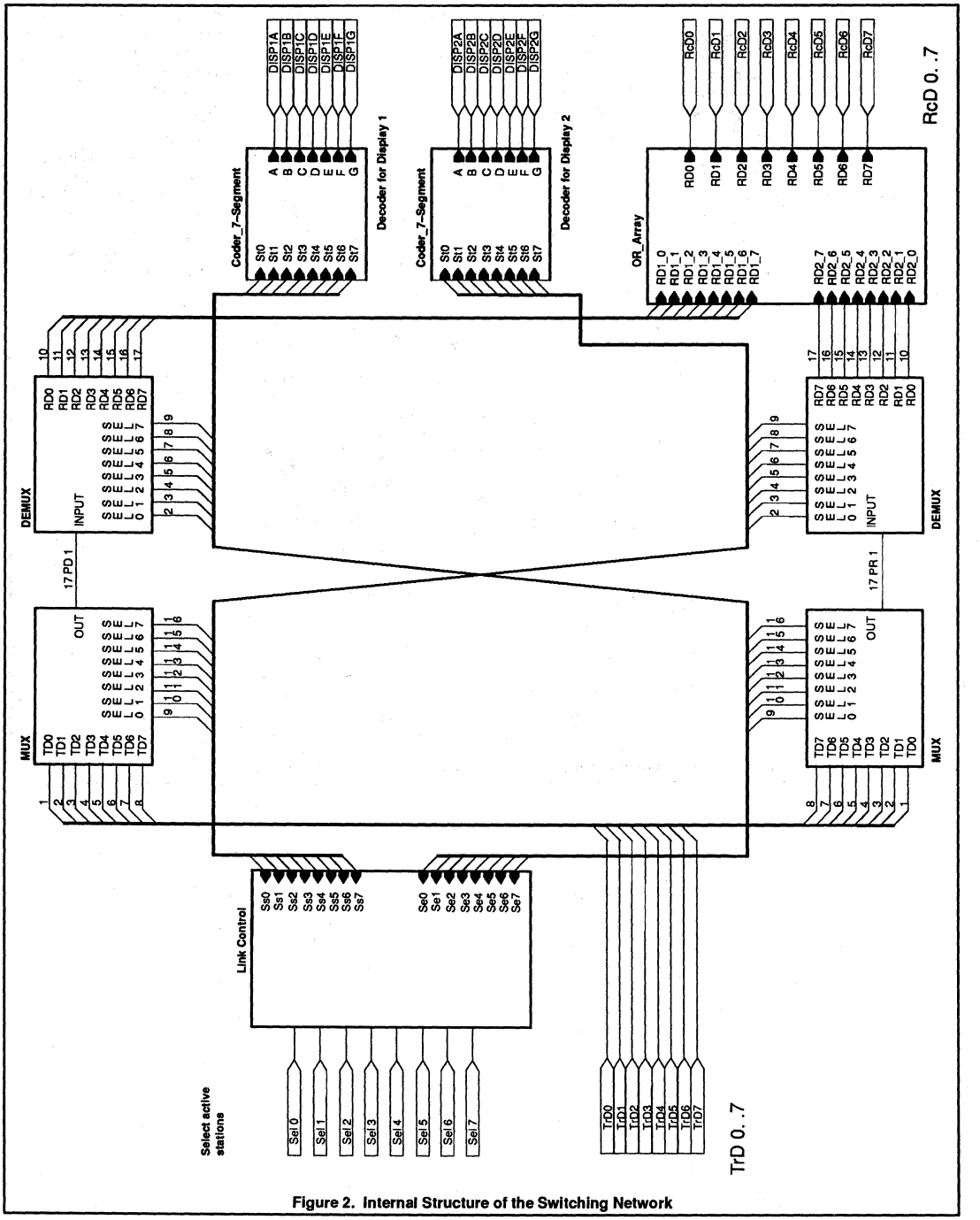


Figure 2. Internal Structure of the Switching Network

Switching control unit for data communication via RS232 AN040

BASIC OPERATION

Link Control and Priority Encoder

The first module of the design is the link control unit. As inputs it has eight selection signals indicating the choice of two stations which are to be connected. This primary selection will be converted to internal control signals for the switching network by the link control. As outputs there are 2x8 signals (8 for each direction of data flow) controlling the multiplexer and demultiplexer directly.

In addition to the basic function, the link control has to ensure that just two stations are connected with each other, even if more are activated by the switches. In order to fulfill this constraint there has been implemented an implicit priority encoder satisfying the following rules:

- if just two stations are chosen by the switches (the normal case) then an interconnect of these stations will be established;
- if only one switch is set active, the transmission signal of this station is directed to the Receive line of the same station (self test);
- if no switch is activated no interconnection is set up at all;
- if more then two switches were set to the 'Active' level just the two stations with the highest and lowest number (Station 0, Station 7 – highest priority) are interconnected with each other.

Realizing such privileges the appropriate function of the switching network is guaranteed in a way, that never more then two stations are linked together.

Multiplexer

Two multiplexers are the first part of the internal switching network. Directly controlled by the output of the link control module the multiplexers have the task to switch the transmission line of a selected station to an internal crosspoint. Since both directions of data flow are to be supported, two multiplexers exist which link the TrD lines to the crosspoints PD and PR respectively.

Demultiplexer

The second part of the digital switching network is formed by two demultiplexers. While the multiplexers have to link the active transmitter to central crosspoints, the demultiplexers connect these internal nodes with the RcD line of the corresponding counter station. The links to be established are as well controlled by the link control module and in the result each set of multiplexer/demultiplexers realizes an interconnection between any TrD and RcD signal lines. Since two of these sets are contained, both directions of data flow can be satisfied and a logical OR-operation of both demultiplexer outputs completes the switching network.

Status Display

A display was added to the basic function to show the actual interconnection. Since the original task was to connect eight stations with each other two numerical displays were used to display just the number of the two active ones. In order to accomplish this function two identical decoders were created. The decoders read directly the eight control signals from the link control module and in correspondence to the active line they drive the displays with numbers 1 to 8. For the case that no station is selected, the displays will blank.

THE PML-ARCHITECTURE

Circuits with a PML-structure represent an architecture that can replace all typical PLDs. In contrast to PAL and PLA circuits which use AND and OR gates, PML is composed of just one programmable logic array using only NAND gates. The outputs of each gate folds back upon itself and all other NAND gates. Inputs, outputs and bi-directional pins are available in the same way as they are in other PLDs.

The general PML-architecture is shown in Figure 3. It is possible to realize any logical function block with this architecture. An example of implementing a D-type Latch is shown in Figure 4. Efficient design implementations require a certain design style and software that includes an appropriate simulator, optimizer and compiler. Such tools are contained in the PLD-Development Software SNAP, which can be used to implement designs within a short period of time for PML as well as for other programmable logic devices.

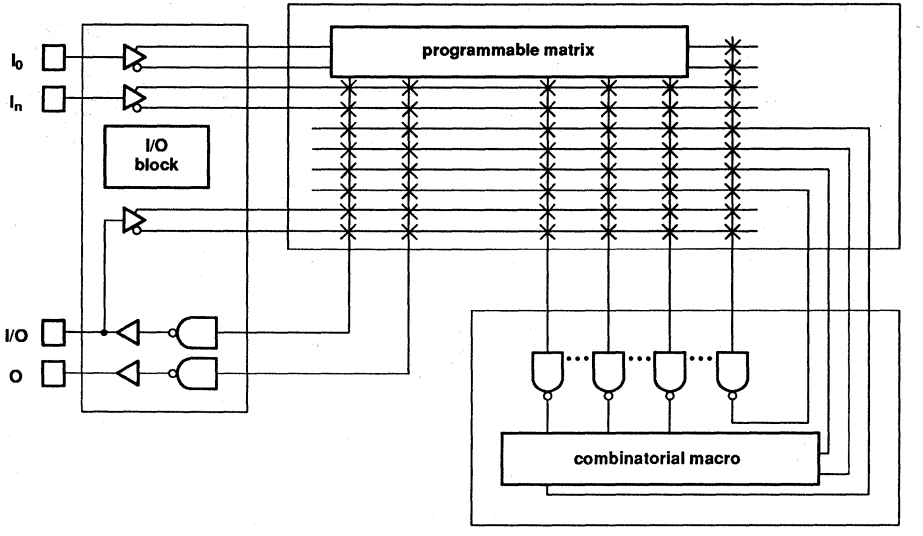


Figure 3. The Basic Architecture of PML-Circuits

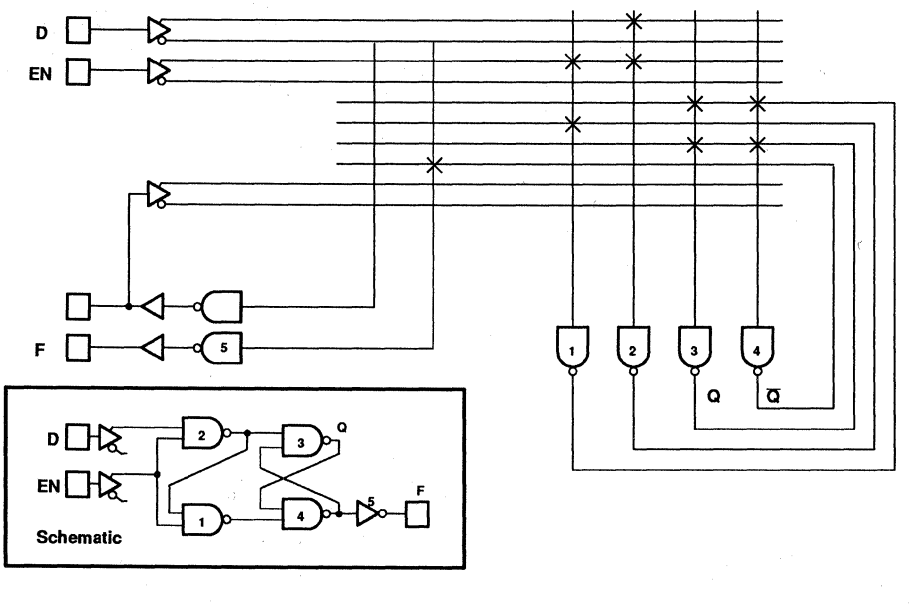


Figure 4. Realization of a D-Latch by a PML-Architecture

Switching control unit for data communication via RS232 AN040

A DESIGN TOOL FOR PML-CIRCUITS (SNAP)

Design implementation for PML as well as of other programmable logic ICs is supported by the Software tool SNAP. This program package, offered by Philips Semiconductors, guides the user through the complete design process beginning with the design description up to the automatic generation of final programming data and a corresponding test vector file. For the initial description of designs several entry tools are available. First there is the opportunity to create a network description via a schematic editor (e.g. OrCAD SDT). Alternatively SNAP defines a specific HDL (Hardware Description Language) and by using this method, a design description may be given in terms of Boolean equations, truth tables or FSM (Finite State Machines) syntax.

Whichever entry method is used becomes converted into an internal network description in a following step. The internal network format is similar to the EDIF-format and it corresponds with the data formats used by Philips Design Station for ASIC development. So it is also possible with SNAP to import designs given as EDIF-network descriptions and moreover the internal format keeps open the choice of a PLD or a Gate Array design implementation. Finally SNAP contains a minimzer module can optimize the Boolean function thereby increasing the quality of the design implementation, sometimes significantly.

After a description has been created, the point of interest is to verify the correct operation of the design. For this purpose SNAP has included an easy to use digital simulator. The simulator, LESIM3, is contained in SNAP as well as in the Philips Design Station which has been used for ASIC-development for many years.

Basically LESIM3 supports the simulation of abstract network descriptions as well as of concrete circuit models. The simulator either assumes a constant, propagation delay value for a given, bare network or it can consider the real timing relations if the model for a certain IC is available. The stimuli for a design simulation can be generated by an interactive, graphical wave form entry or a textual stimuli description. The graphical entry is very easy to handle and leads quickly to appropriate stimuli for smaller designs. The other choice, the textual notation of stimuli, bases on an own simulation control language (SCL) of LESIM3 and it allows the compact description of sophisticated sequences for extensive design verification. This opportunity serves more the needs of experienced designers and in correlation with fault simulator, test pattern generator, model generator and other options it makes possible all kinds of functional simulation and test.

The goal of design development software is the generation of ICs realizing an application specific function. Since the initial design description in SNAP is device independent, one task is to select a specific component. Even the choice of a PLD is made easy by SNAP, since all available devices are listed directly on the screen together with their most relevant data. After the designer has decided which PLD to use, there is only left the task of specifying pinning of the device. An initial pin assignment is suggested by SNAP automatically, which can be revised in an interactive process directly on the graphical presentation of the device.

If all assignments are done, the ultimate design step is the compiler run. Automatically the compiler tries to map the given network on the selected PLD-architecture and in case of success a programmer file is generated (the format of programmer file follows JEDEC No.3A standard). This file can be downloaded onto a device programmer directly from the SNAP-shell, so that the IC-implementation can be completed immediately by programming the physical device. If on the other hand the compiler fails in its run the appeared problems will be listed in a status file and in this way the designer gets hints how to modify the design for a successful compilation. Especially in these cases SNAP has an essential advantage, since the SNAP-shell with its clear structure (Figure 5) allows additional design iterations quickly and furthermore automated.

DEVELOPMENT OF THE SWITCHING CONTROL UNIT WITH SNAP

Design Description

The general functionality of the intended switching network has been outlined already in section two. There, the complete design has been subdivided into basic modules, of which the logical function can easily be described by boolean equations. Therefore it makes sense to use the 'Equations' entry of SNAP for design description and activating this module from the SNAP-shell an editor makes it possible to fill in a script of an empty HDL-file. Using the Equation-entry any design definition has to follow the specific HDL-syntax of SNAP and Figure 6 shows this description for the complete design.

Referring to the listing (Figure 6) the basic structure of HDL-files can be demonstrated. The first section '@PINLIST' serves for the definition of primary inputs, outputs and bi-directional pins of a design. Just the names of the ports and the port types are fixed here. After declaring the I/O-part the section '@LOGIC EQUATIONS' contains the definition of the designs functionality. In this part the output is to specify in terms of boolean equations, in which inputs, intermediate signals and logical operators can be used. The basic operators are AND ('*'), OR ('+'), NOT ('/') and EXOR (':+:'), but more abstract operations are available too.

Within a HDL-file additional sections can be used for describing truth tables and finite state machines, but since they are not necessary for the switching network they are not explained here. Finally it should be mentioned that any project file in SNAP gets a specific file name extension upon completion of each development stage, so that the created Equation-file is marked with '.EQN' and the project file is named as NullMod.Eqn.

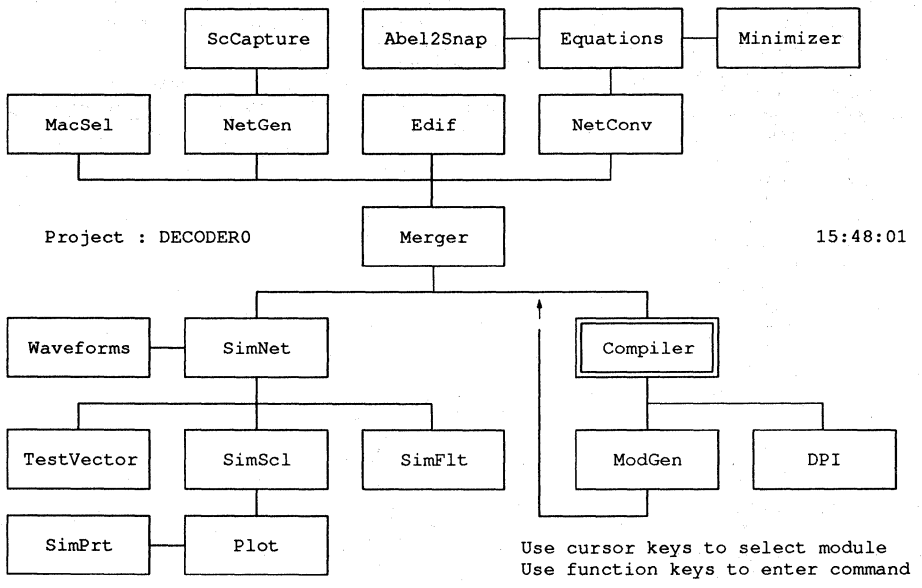


Figure 5. Snap Shell

```
@PINLIST
Sel0        I;
Sel1        I;
Sel2        I;
Sel3        I;
Sel4        I;
Sel5        I;
Sel6        I;
Sel7        I;
TrD0        I;
TrD1        I;
TrD2        I;
TrD3        I;
TrD4        I;
TrD5        I;
TrD6        I;
TrD7        I;
RcD0        O;
RcD1        O;
RcD2        O;
RcD3        O;
RcD4        O;
RcD5        O;
RcD6        O;
RcD7        O;
Disp1A      O;
Disp1B      O;
Disp1C      O;
Disp1D      O;
Disp1E      O;
Disp1F      O;
Disp1G      O;
Disp2A      O;
Disp2B      O;
Disp2C      O;
Disp2D      O;
Disp2E      O;
Disp2F      O;
Disp2G      O;

@LOGIC EQUATIONS

Ss0 = Sel0 ;
Ss1 = /Sel0 * Sel1;
Ss2 = /Sel0 * /Sel1 * Sel2;
Ss3 = /Sel0 * /Sel1 * /Sel2 * Sel3;
Ss4 = /Sel0 * /Sel1 * /Sel2 * /Sel3 * Sel4;
Ss5 = /Sel0 * /Sel1 * /Sel2 * /Sel3 * /Sel4 * Sel5;
Ss6 = /Sel0 * /Sel1 * /Sel2 * /Sel3 * /Sel4 * /Sel5 * Sel6;
Ss7 = /Sel0 * /Sel1 * /Sel2 * /Sel3 * /Sel4 * /Sel5 * /Sel6 * Sel7;

PD =        TrD0 * Ss0 + TrD1 * Ss1 + TrD2 * Ss2 + TrD3 * Ss3 +
            TrD4 * Ss4 + TrD5 * Ss5 + TrD6 * Ss6 + TrD7 * Ss7 ;

Se7 = Sel7 ;
Se6 = /Sel7 * Sel6 ;
Se5 = /Sel7 * /Sel6 * Sel5 ;
Se4 = /Sel7 * /Sel6 * /Sel5 * Sel4 ;
Se3 = /Sel7 * /Sel6 * /Sel5 * /Sel4 * Sel3 ;
Se2 = /Sel7 * /Sel6 * /Sel5 * /Sel4 * /Sel3 * Sel2 ;
Sel = /Sel7 * /Sel6 * /Sel5 * /Sel4 * /Sel3 * /Sel2 * Sel1 ;
Se0 = /Sel7 * /Sel6 * /Sel5 * /Sel4 * /Sel3 * /Sel2 * /Sel1 * Sel0;

PR =        TrD7 * Se7 + TrD6 * Se6 + TrD5 * Se5 + TrD4 * Se4 +
            TrD3 * Se3 + TrD2 * Se2 + TrD1 * Sel + TrD0 * Se0;
```

Figure 6. The Complete Design Description for the Switching Network Circuit (1 of 2)

```
RcD7 = PD * Se7 + PR * Ss7 ;
RcD6 = PD * Se6 + PR * Ss6 ;
RcD5 = PD * Se5 + PR * Ss5 ;
RcD4 = PD * Se4 + PR * Ss4 ;
RcD3 = PD * Se3 + PR * Ss3 ;
RcD2 = PD * Se2 + PR * Ss2 ;
RcD1 = PD * Se1 + PR * Ss1 ;
RcD0 = PD * Se0 + PR * Ss0 ;

Disp1G = /( Se2 + Se3 + Se4 + Se5 + Se6 );
Disp1D = /( Se0 + Se2 + Se3+ Se5 + Se6 );
Disp1E = /( Se0 + Se2+ Se6 );
Disp1F = /( Se0 + Se4 + Se5 + Se6 );
Disp1A = /( Se0+ Se2 + Se3+ Se5 + Se6 + Se7 );
Disp1B = /( Se0 + Se1 + Se2 + Se3 + Se4+ Se7 );
Disp1C = /( Se0 + Se1 + Se3 + Se4 + Se5 + Se6 + Se7 );

Disp2G = /( Ss2 + Ss3 + Ss4 + Ss5 + Ss6 );
Disp2D = /( Ss0 + Ss2 + Ss3 + Ss5 + Ss6 );
Disp2E = /( Ss0 + Ss2 + Ss6 );
Disp2F = /( Ss0+ Ss4 + Ss5 + Ss6 );
Disp2A = /( Ss0 + Ss2 + Ss3+ Ss5 + Ss6 + Ss7 );
Disp2B = /( Ss0 + Ss1 + Ss2 + Ss3 + Ss4 + Ss7 );
Disp2C = /( Ss0 + Ss1 + Ss3 + Ss4 + Ss5 + Ss6 + Ss7);
```

Figure 6. The Complete Design Description for the Switching Network Circuit (2 of 2)

Generation of an Internal Netlist

As already mentioned, each design description within SNAP becomes converted into an internal netlist format first. In general this step is carried out automatically just by activating the corresponding program module and for the actual design this means to start the 'Net Converter' and the 'Merger' in succession. The network converter translates the given HDL-file into an EDIF like netlist description, while the merger has no essential meaning for this design. It is rather important for larger, hierarchical designs, where this module has to merge several function blocks into one, flat netlist.

Regardless of design complexity, the merger has to be executed for each design and the out coming netlist file with the name NullMod.Net (project name with extension '.NET') forms the bases for the following simulation and for the compiler. That's why a part of the resulting netlist is shown in Figure 7 for demonstration of this important, internal file format.

```
*******************************************************
* Output of Merger                Version 1.60    *
* Date: 10/ 1/1993                Time: 14:26:44  *
*******************************************************
* Input File Name:                NULLMOD.MAC   *
* Netlist File Name :             NULLMOD.NET    *
*******************************************************
*
NETSTART
*
SS1 AN2 I(N48_1,Sel1) O(SS1)
B48_1 INV I(Sel0) O(N48_1)
SS2 AN3 I(N49_1,Sel2,N48_1) O(SS2)
B49_1 INV I(Sel1) O(N49_1)
SS3 AN4 I(N49_1,N50_1,Sel3,N48_1) O(SS3)
B50_1 INV I(Sel2) O(N50_1)
                      #
                      #
                      #
Disp2B NO6 I(Sel0,SS1,SS2,SS3,SS4,SS7) O(Disp2B)
Disp2C NO7 I(Sel0,SS1,SS3,SS4,SS5,SS6,SS7) O(Disp2C)
*
NETEND
*
NETIN Sel0,Sel1,Sel2,Sel3,Sel4,Sel5,Sel6,Sel7,TrD0,TrD1,TrD2,
#TrD3,TrD4,TrD5,TrD6,  TrD7
NETOUT RcD0,RcD1,RcD2,RcD3,RcD4,RcD5,RcD6,RcD7,
#Disp1C,Disp1B,Disp1A,Disp1F,Disp1E,Disp1D,Disp1G,
#Disp1C,Disp2B,Disp2A,Disp2F,Disp2E,Disp2D,Disp2G
*
```

Figure 7. The Netlist File NullMod.Net

Switching control unit for data communication via RS232 AN040

Design Simulation

The simulation of a design is an important step in the development of each circuit. Beside the bare functional simulation, the actual device timing relations, the testability, and last but not least an appropriate test pattern needs to be generated. All these functions can be realized by SNAP, even if only the basic procedure is demonstrated here.

Working with SNAP the first step is to convert the netlist into a more compact format easier to read for the simulator. This will be accomplished by starting the program module 'SIMNET' and as a result a binary file with the name NullMod.Bin becomes created. Additionally a stimuli file NullMod.Scl will automatically be generated too. These automatically generated stimuli are quite arbitrary and so it is advisable to modify them by the module 'WAVEFORMS' or by the available text editor. Are the binary design file and the stimuli file present the real simulator 'SIMSCL' can be run and the simulation results are stored in a project file ending with '.RES' (NullMod.Res for the switching network). These results can either be visualized graphically on the screen by 'PLOT' or printed out by the program module 'SIMPRT'.

A complete simulation takes a good deal of work for any design and it can become quite extensive. That's why only a small part of the stimuli file for the switching network shall be shown here for demonstration (Figure 8) and Figure 9 shows the corresponding section of the graphical simulator output.

```
*************************************************************
* Output of Updsirn                Version 1.00             *
* Date: 10/01/93                   Time: 14:27:02           *
*************************************************************
* Input File Name:                 NULLMOD.NET     *
* Output File Name:                NULLMOD.SCL     *
*************************************************************

P Sel0, Sel1, Sel2, Sel3, Sel4, Se0, Se1, Se2, Se3
PC0
S 0  (500) Sel0
S 0 (1000) Sel1
S 0 (1700) Sel2
S 0 (2500) Sel3
S 0 (2800) Sel4
S 0 (3100) Sel5
S 0 (3500) Sel6
S 0 (4000) Sel7

SU time = 5000
```

Figure 8. A Section of the Simulator Stimuli for the Verification of the Design

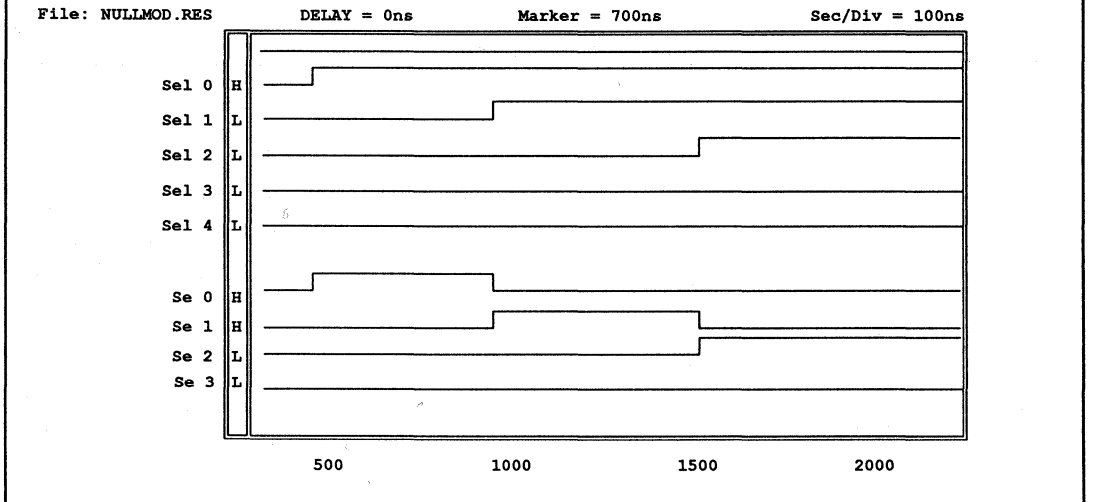


Figure 9. Graphical Presentation of Simulator Output

The Compiler

The final aim of the design process is to generate a data file in order to customize programmable logic devices with the functionality of the developed design. The programming data format generated by SNAP follows the international standard JEDEC No. 3A, which is widely accepted and used in industry and research.

The compiler block of SNAP has the task to map a given, verified netlist onto a certain device architecture and to convert the result into a JEDEC file readable by a device programmer. In order to do so a specific device should be chosen and a pin out selected. For this design, a PLHS501 is chosen within the program module 'DEVSELECT' and the Pinning needs to be entered with 'PINSELECT' in correspondence to that of Figure 10. After running these two programs, the compiler can be started which fits the design into the chosen device. In case of problems during the compilation an error file (with extension '.ERR') will be generated reporting errors, warnings and information about necessary modifications to the initial design. But for the design of the switching network the compiler finishes its work with success resulting in a JEDEC file named NullMod.Jed. This file can now directly be used to program a PLHS501 or it can be modeled for a final timing simulation.

PROGRAMMING OF THE PLHS501

Among other alternatives the UP2000 device programmer represents an interesting, efficient tool for the development of programmable logic especially in laboratories and smaller enterprises. Manufactured for use in PC environments it consists of an expansion board for PCs (8-Bit Slot), an adapter box, several adapters for mounting various package layouts, and corresponding programmer software. This device programmer and associated software supports all of the programmable logic devices that are contained within the SNAP library, so that the UP2000 is the ideal companion for the SNAP development system. The programmer software is mostly self explanatory and with some skill it can directly be included into the SNAP shell. Using the JEDEC file from SNAP and a UP2000 to program a PLHS501 revealed no problems at all. The programming of the design of the switching network into the PLHS501 was finished within one minute.

SUMMARY

The design of a digital switching network as described here was implemented within just one Philips PLHS501. Adding some buffer/drivers and the necessary switches and display elements completed the design to a working application. It was tested within a laboratory of 8 PCs, which were interconnected by the switching network via their standard serial interfaces. By changing the switches, any of the computers could be interconnected and using communication link programs such as LapLink or Norton Commander, a simple data transfer method was realized.

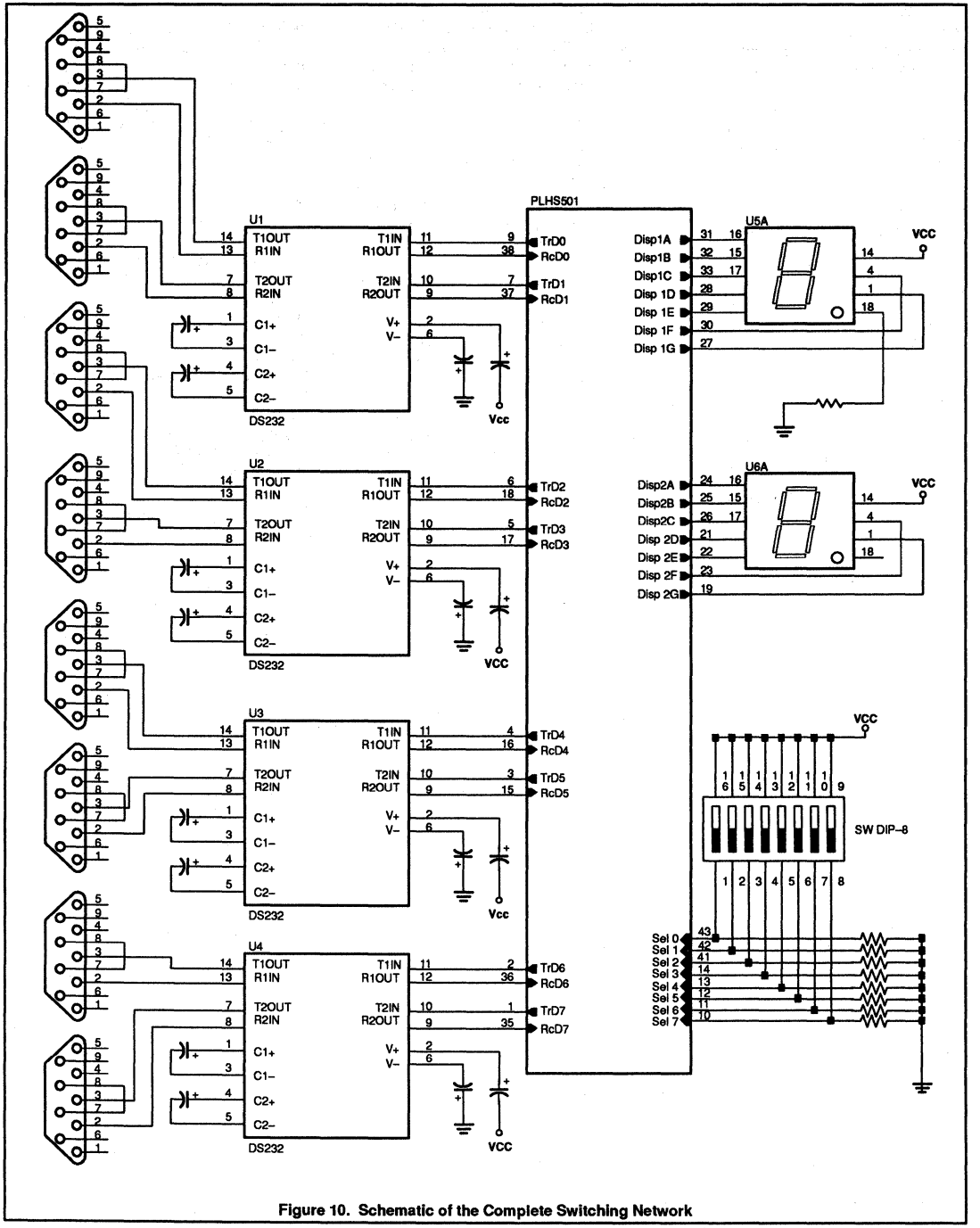


Figure 10. Schematic of the Complete Switching Network

Microcontroller acceleration AN035

INTRODUCTION

This paper presents the design of an accelerator that increases microcontroller performance without resorting to multiple expensive high speed memory chips or a high-cost superscalar controller. The design of a data acquisition unit, similar to a logic analyzer, is presented as an example. The task of the analyzer is to capture and respond to data, sampled at a rate faster than the microcontroller can operate.

Data capture is a hardware-intensive function — it is difficult using software to synchronize controller operations with external data and balance memory requirements. On the other hand, formatting captured data and outputting it for display is best done in software. The accelerator exploits these characteristics by using a programmable logic device with the microcontroller. This partitions data acquisition and display into fast hardware-intensive and slower, software-intensive functions.

What is Microcontroller Acceleration?

Microcontroller acceleration is anything that makes a microcontroller appear to operate faster. Accelerators are special purpose hardware units that perform a time critical task that a microcontroller cannot. In the microprocessor world, accelerators have been hardware arithmetic units, simulation engines or graphics engines. The accelerator works with the microcontroller, either intercepting the time critical activity automatically (coprocessor), or acting when requested by the microcontroller.

To illustrate, an arithmetic coprocessor automatically intercepts multiply and divide instructions, which slow down an integer based microprocessor. The coprocessor does the arithmetic operations with special purpose state machines and floating point hardware. A simulation engine performs high speed logic evaluations and operates directly on a circuit model contained in memory. Graphics engines operate directly with the display memory, performing the basic housekeeping and bit transformations with special purpose hardware. Each acceleration device permits the host processor to execute other code while the accelerator operates at high speed doing its special purpose activity.

SYSTEMATIC APPROACH TO ACCELERATION

Can we systematically design hardware accelerators? The answer to that question is – probably. The standard approach is part science and part art, much like system partitioning. In fact, accelerator definition tracks functional partitioning in many ways.

The microcontroller system must be designed by first identifying microcontroller activities and those that must be done externally. The same approach is used for microprocessors as for microcontrollers, and will be shown by example.

Matching a microcontroller or microprocessor to a target task involves several steps.

1. Identify tasks that must operate at full speed and tasks that can operate slower.
2. Identify operations that are most efficient when handled by microprocessor data paths and those that must be outside, handled by high speed logic.
3. Maximize tasks that can be done by software and minimize tasks that must be done by application specific hardware.
4. Identify which tasks must operate with custom hardware because of their asynchronous nature.

Much of the task partitioning is intuitive, but some may be quantitatively derived. For instance, the cycle time of the microprocessor is a constraint. Any activity that occurs in less time than the cycle time of the processor must by handled by hardware that can respond quickly enough. Any activity that is not critical is a candidate for being handled by the microprocessor.

The key to success in acceleration is correctly identifying the tasks that must be done at speed and those that can wait. This is the first partition.

The next step is to identifying slow and fast datapath operands. For instance, initializing a counter or a comparator may be done slowly, but the counting and comparing operations must occur at full speed. Isolating these kinds of operations is another partition.

Additional steps may or may not be needed. However, synchronizing signals that are asynchronous is very difficult for controllers to do unless the signals are extremely slow and the degree of synchronization is loose.

AN EXAMPLE

Let's look at an example taken from the instrumentation world. Most digital engineers are familiar with logic analyzers, because they use them for digital system debug. The idea is to make an add-on unit for a personal computer, using the PC's abilities to manage and display data. However, the PC cannot do high speed data capturing or triggering (stopping the data capture), unless ridiculously low speeds are the goal. For a 40 MHz data capture rate, these tasks must be done by a second processor, passing data frames to the PC. This unit, shown in Figure 1, could even be inexpensively designed into a system that might benefit from remote access diagnostics. Clearly, it can operate stand-alone, as well.

To handle the 40 MHz data rate, samples must occur every 25 nanoseconds, and be stored in consecutive RAM locations. The RAM will hold a sampled time image of probe voltages captured by the analyzer. Just passing the data into the RAM is no problem, but an address counter must manage the RAM. We'll assume a 20 nanosecond SRAM, giving some time margin.

There are RISC processors today, that could pass data through them and update counters near this rate. However, that is not all that must be done. This is the tough part – each data item must be examined if a pretriggering feature is implemented.

Pre-triggering permits the analyzer to recognize a data item, and when recognized, stop capture. This important feature permits designers to isolate bugs, and find what caused them. An important added feature, is time delayed stopping.

Time delayed stopping permits more data capture after the trigger event, allowing assessment of the results of a bug. Time delayed stopping must be done at full hardware rates to be effective. It gets complicated.

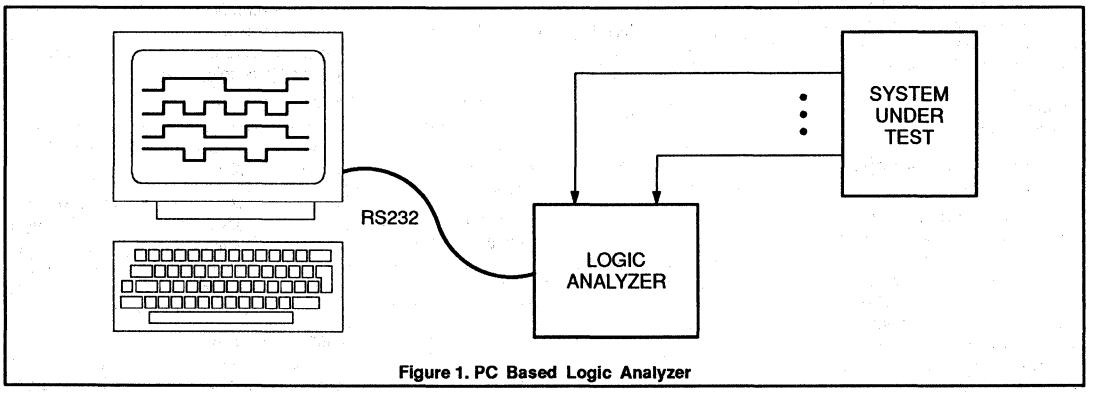


Figure 1. PC Based Logic Analyzer

Now, let's outline the tasks to be done.

1. The logic analyzer must be enabled by the PC.
2. Each sample must be captured, and examined.
3. In parallel, a RAM address must be calculated.
4. The data must be stored at the address.
5. If the data matches a trigger value, a time delay procedure must be started.
6. Samples must continue to be captured and addresses calculated.
7. When the time delay value is reached, sample capture is to cease.
8. The PC must be informed that sample capture has ceased.
9. At this point, the PC assumes control and reads back the RAM for display.

Everything is possible for a fast RISC to perform up to step four, when a trigger event occurs. At this point, the complexity increases to the point that at least four tasks occur simultaneously. Examining the sample, storing it, updating the address and the delay must occur in one 25 nanosecond period, to keep up with arriving samples. This suggests hardware speeds are required, and sets the speed of the events to occur. Since each task can be done within the same 25 nanosecond period, the next sample can arrive and be handled. Even if the RISC could keep up with the tasks, it would be a prohibitive solution (at least one thousand dollars) for processor and memory. Twenty nanosecond SRAMs aren't cheap, and fast 32-bit RISCs need several of them.

Even by restricting the analyzer to an 8 bit one, doesn't help much. RISC processors of this speed are inherently 32 bits wide. This motivates us to find an alternative solution. We'd like to make a unit for less than 100 dollars and fifty is preferable.

First, let's partition the problem into two pieces, on-line tasks and off-line tasks. On-line tasks are those that occur during data capture, and must meet the 25 nanosecond time constraint. Off-line tasks are those that occur when the heat is off, and have no particular time constraints beyond "user friendly" speeds.

Enabling the analyzer, reading back the RAM contents and dealing with the PC are off-line tasks. Sample capture, store and triggering activities are on-line tasks. The design must do the on-line tasks in pure hardware and the off-line tasks in micro-controller software. Figure 2 shows the microcontroller and data capture unit along with an SRAM.

By isolating the high speed operations into application specific hardware, forming an accelerator, a large class of inexpensive microcontrollers can meet the needs of the off-line tasks. First, because the needs of this example are only for 8 bit data, we'll use an 8 bit microcontroller. A microcontroller with enough I/O ports and the ability to do the off-line tasks in a single chip will be the most desirable. Off-line tasks include managing the sample RAM, communicating with the PC and enabling or disabling the accelerator.

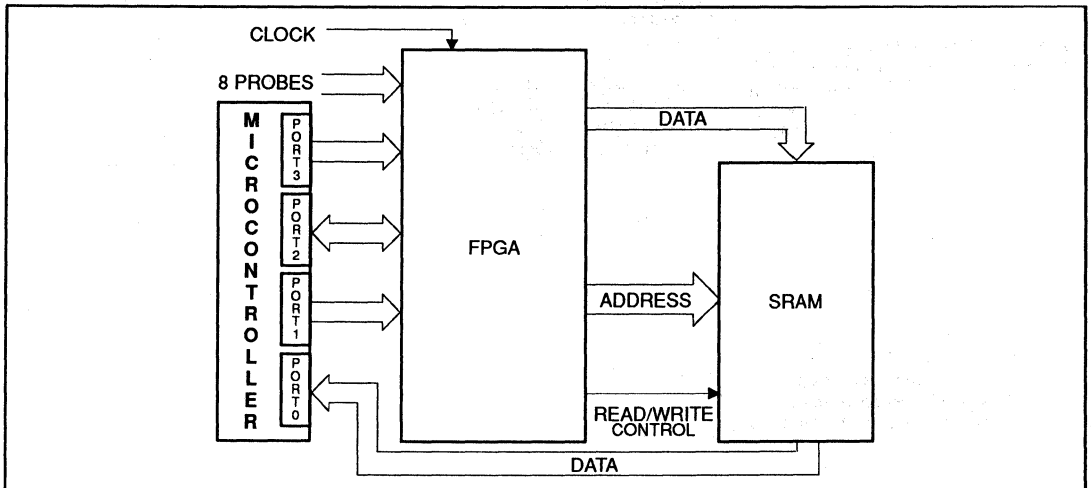


Figure 2. Logic Analyzer Block Diagram

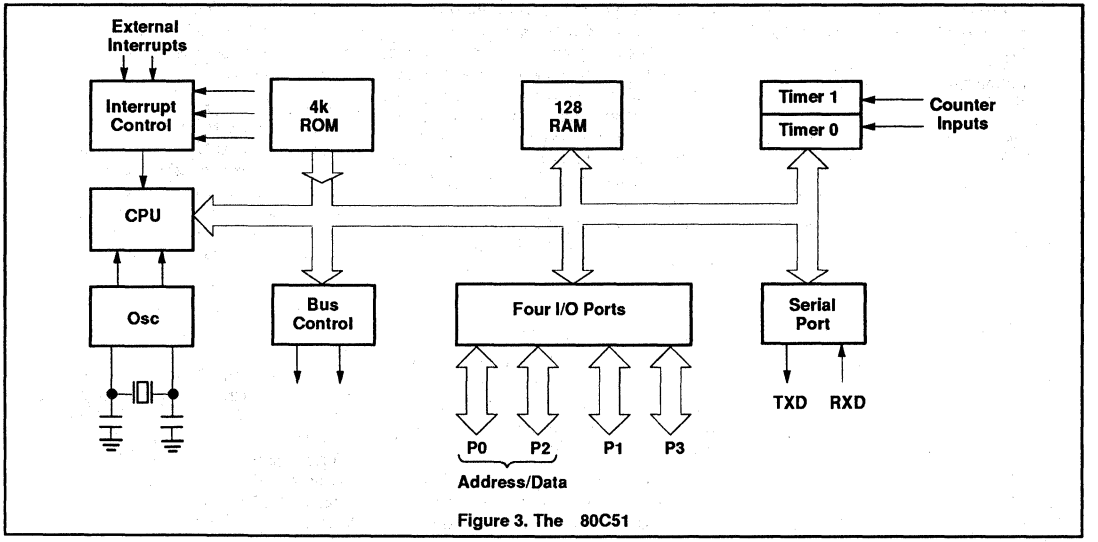


Figure 3. The 80C51

MICROCONTROLLER SELECTION

The design requires serial access to the PC, and at least four ports for handling the accelerator. These requirements are met by a single 8051 type microcontroller. Because 8051s are inexpensive and readily available, let's pick one that has an additional feature – speed – ensuring an enjoyable user interface for off-line activities. The 33 Mhz version of the 8051, offered by Philips Semiconductors meets the basic needs.

An interesting aspect of the 8051 family, is that there are so many derivative parts. The design presented here, can be adapted in many ways to meet other needs. For instance, I picked the 33 MHz version, preserving user frustration. I could have picked the 80CL410 for low power, the 87C751 to minimize the size, or the 87C451 if I needed more I/O ports. Other family members have fewer ports (in smaller packages) and built in A/D converters. For our needs, we'll simply pick the fastest microcontroller, the 80C51. Figure 3 shows the architecture of the 80C51. If the unit was to be portable for a notebook or palmtop PC, the chosen 8051 would probably be different.

ACCELERATOR HARDWARE SELECTION

The accelerator (data capture unit) is another issue. First, it must be able to do the tasks described as on-line, and it must do them in 25 nanoseconds. This should not be confused with a pin to pin propagation time of 25 nanoseconds, but simply at a rate tracking 25 nanosecond samples. As long as the RAM addresses and the captured data arrive at the SRAM together – at a rate equal to 40 MHz – the analyzer tracks the samples. It is important to update the counter at the sample rate, as well as time delay a pretriggered sample at the same rate. This is easily done with an FPGA type part or a complex PLD.

For this task, I picked a Philips Semiconductors PML 2852 (Figure 4). The PML 2852 includes 8 bit registers which operate as input data synchronizers and uses JK flip flops for internal counters. The PML 2852 has 52 JK and D type flip flops, with over a hundred very wide Nand gates inside. The data rate of 40 MHz is well below the PML 2852 system clock rate. Interconnect routing is not a problem because of the fully connectible foldback architecture. In all, the PML 2852 provides enough flip flop and gate resources in an 84 pin package, to form the needed comparators, counters, multiplexers and timers to complete the accelerator.

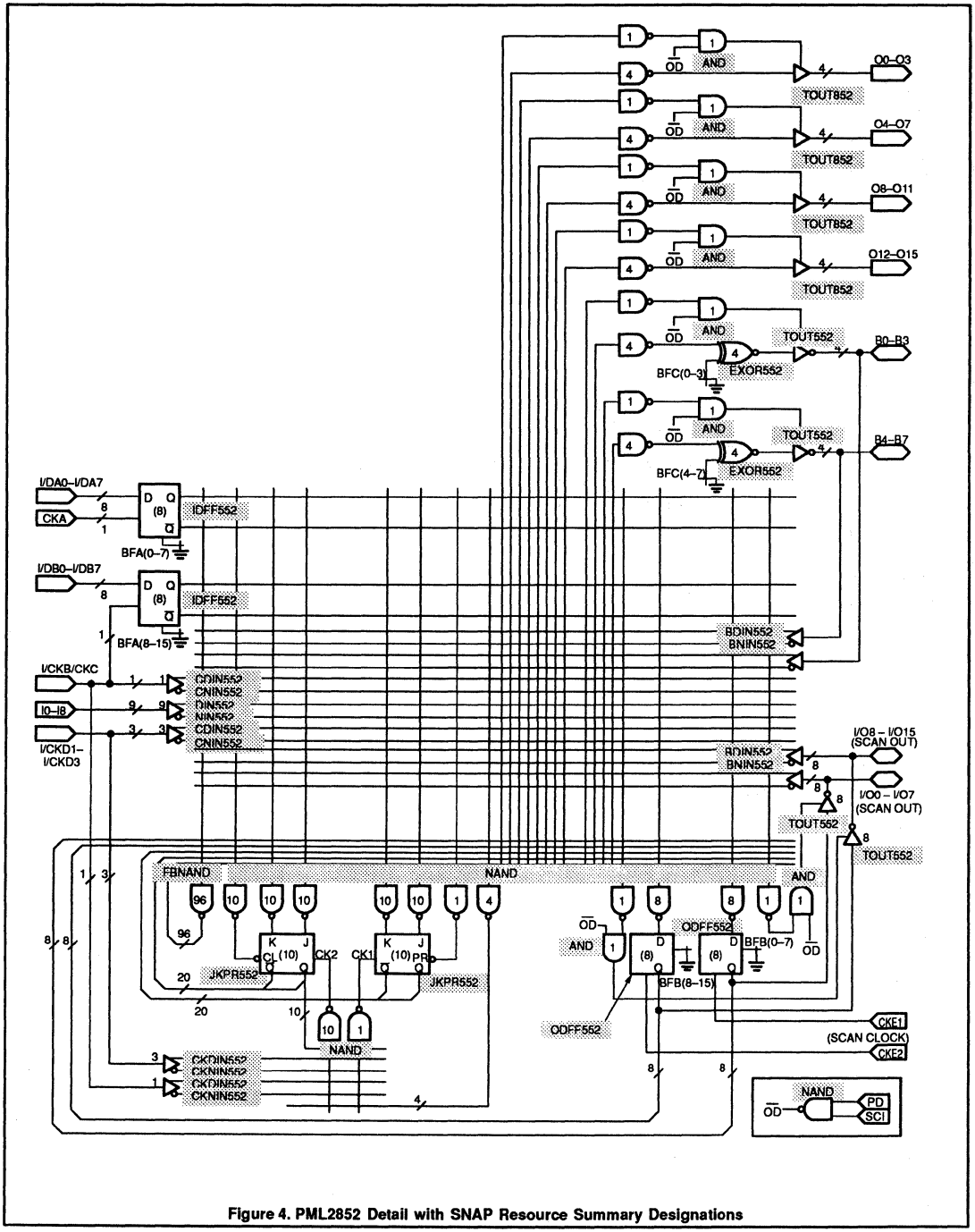


Figure 4. PML2852 Detail with SNAP Resource Summary Designations

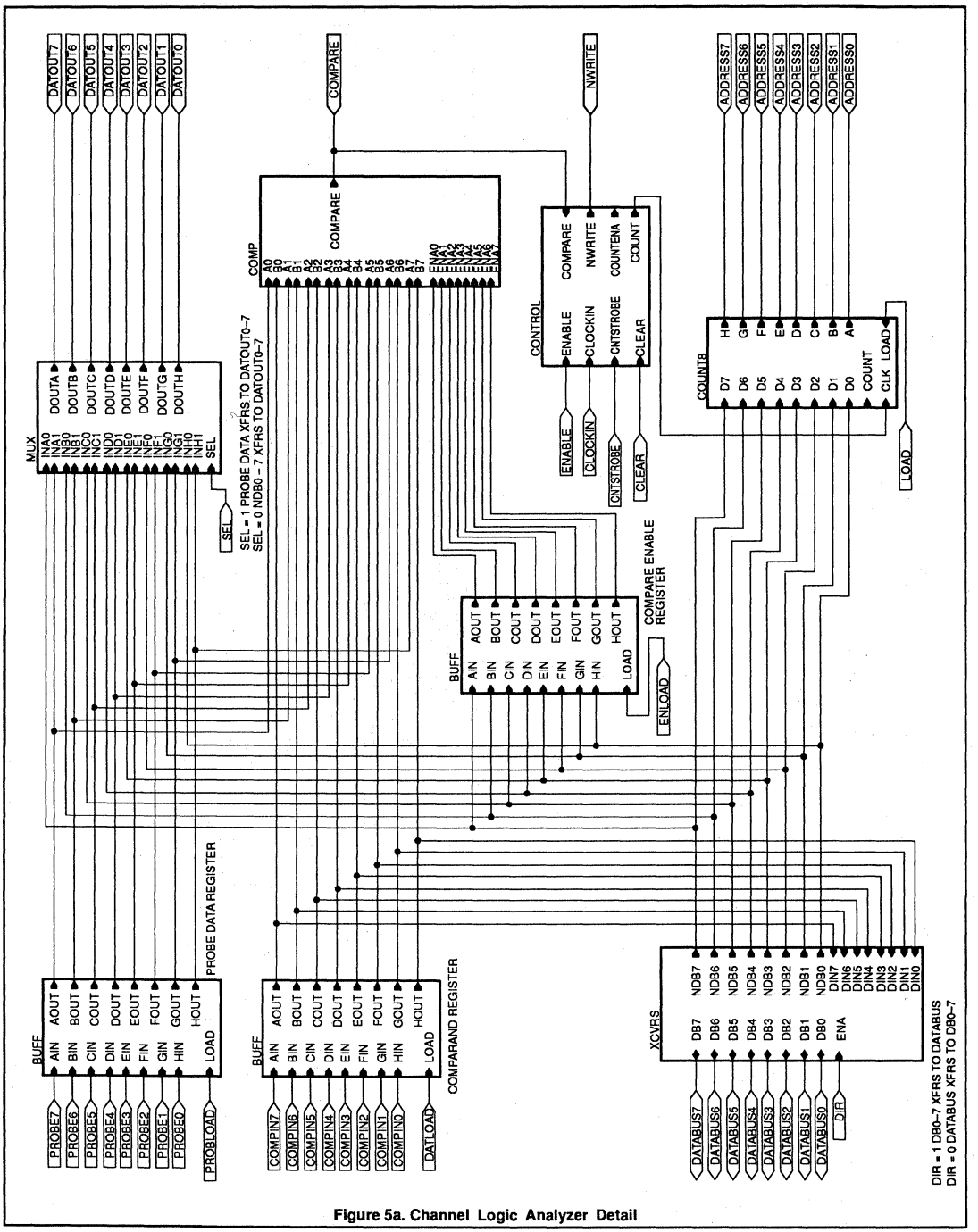


Figure 5a. Channel Logic Analyzer Detail

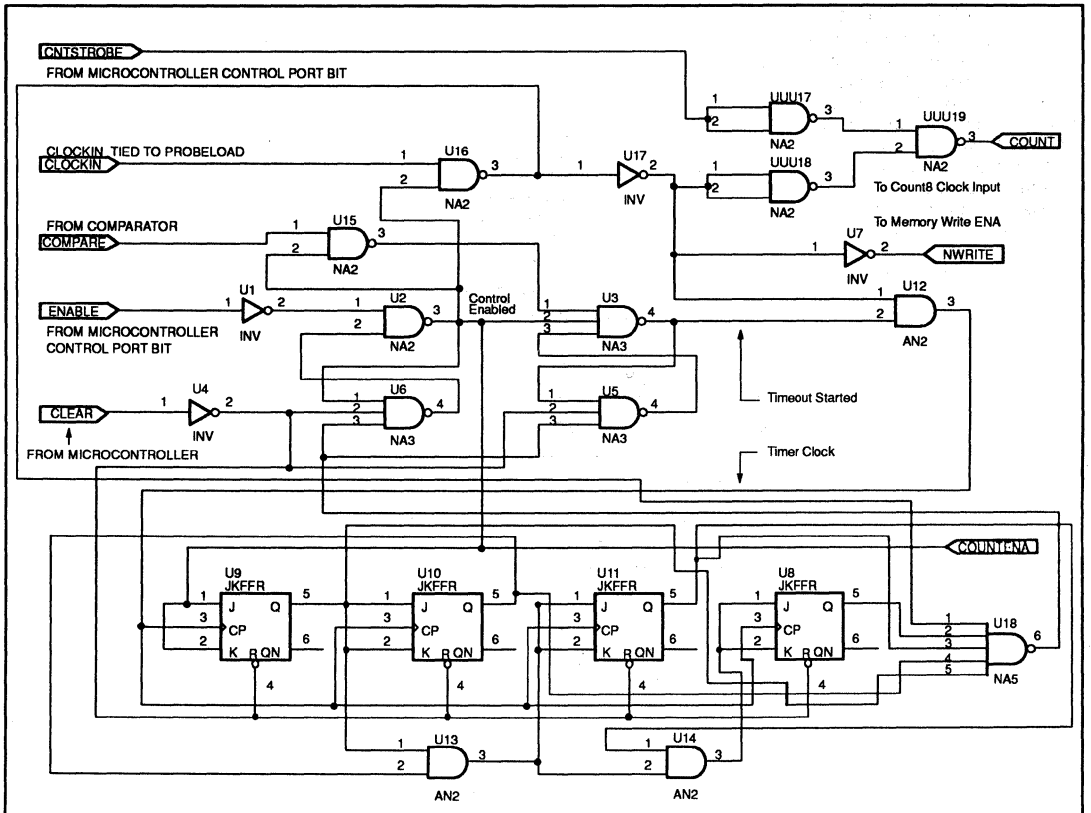


Figure 5b Control Block Schematic

SOME DESIGN DETAILS

Let's look at some of the more critical internal functions (Figure 5a) to understand the accelerator operation. The pretrigger structure is formed with an eight bit register holding a compare pattern, a series of Exclusive-Nor functions and a second register holding bit enables. The bit enables permit "don't care" triggering. The Exclusive-nor gates are made from foldback Nand gates. The entire unit is designed to format 8 bit samples into 256 byte data frames, but the design may be expanded by cascading additional PML 2852s and SRAMs. Cascading can be done using leftover pins, and the fact that the counters and comparators are made in a expandable fashion.

Three of the 8051 ports are used to drive data values into the PML part, with the fourth port makes strobes for the various registers. One of the 8051 ports is designated as bidirectional and is used to read and write data values, compare operands, and counters. The pretrigger in this example is set at a fixed value of 16 time units, but this value can be changed.

The design shown in Figure 5a, was done using Philips Semiconductors PLD design software, SNAP. SNAP (Figure 6) permits a gate array flow for the entire PLD family at Philips Semiconductors, including the PML family. SNAP supports schematic and equation entry, device independent design, optimization, fitting and back annotated logical simulation. All of the blocks in Figure 5a were written using Boolean equations, except for the control block. The control block was described using a schematic shown in Figure 5b. The SNAP equation listings for the other blocks are shown in Figures 9a, b, c, d, and e, with the final PML2852 pin layout in Figure 10.

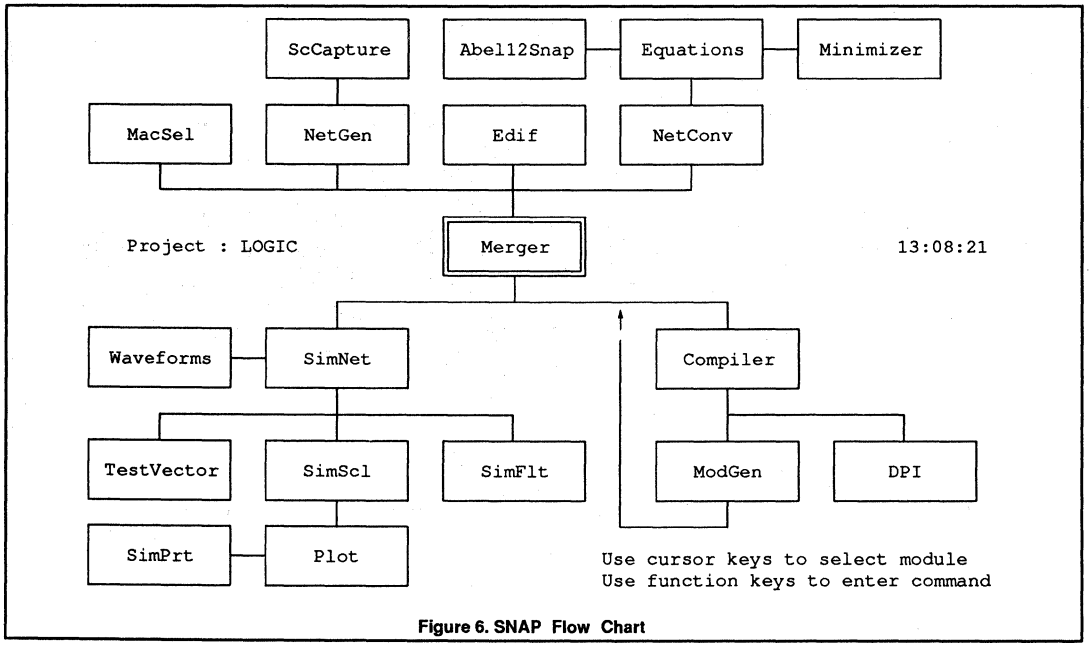


Figure 6. SNAP Flow Chart

ANALYZER OPERATIONS

The order of operations is shown in Figure 7. Initial values are loaded into the counter, the compare and enable registers. When a control strobe is sent from the 8051 to the PML 2852, automatic data capture begins. Data capture continues, loading consecutive values into the SRAM, at that point.

When a data value matches the qualified trigger pattern, a time delay counter is enabled and 16 more samples are taken. At the end of the delay, the SRAM write circuitry is disabled. The 8051 software scans the internal control register and senses the data capture complete condition. The 8051 begins SRAM readback at that point. If a trigger pattern is never encountered, data sampling continues and will load one 256 bit frame after another on top of old values. As mentioned before, the depth of the SRAM can be easily expanded to capture these frames using more parts, if needed.

Readback occurs when the 8051 resets the SRAM address and reads 256 consecutive values from it. The 8051 updates the counter, reads the SRAM and inputs the data. The data is passed on an RS-232 cable to the PC, where a separate display routine takes the data and forms a familiar waveform pattern. Remember that readback is an offline activity. The data capture hardware is stopped, and the RAM dumping occurs at a leisurely rate.

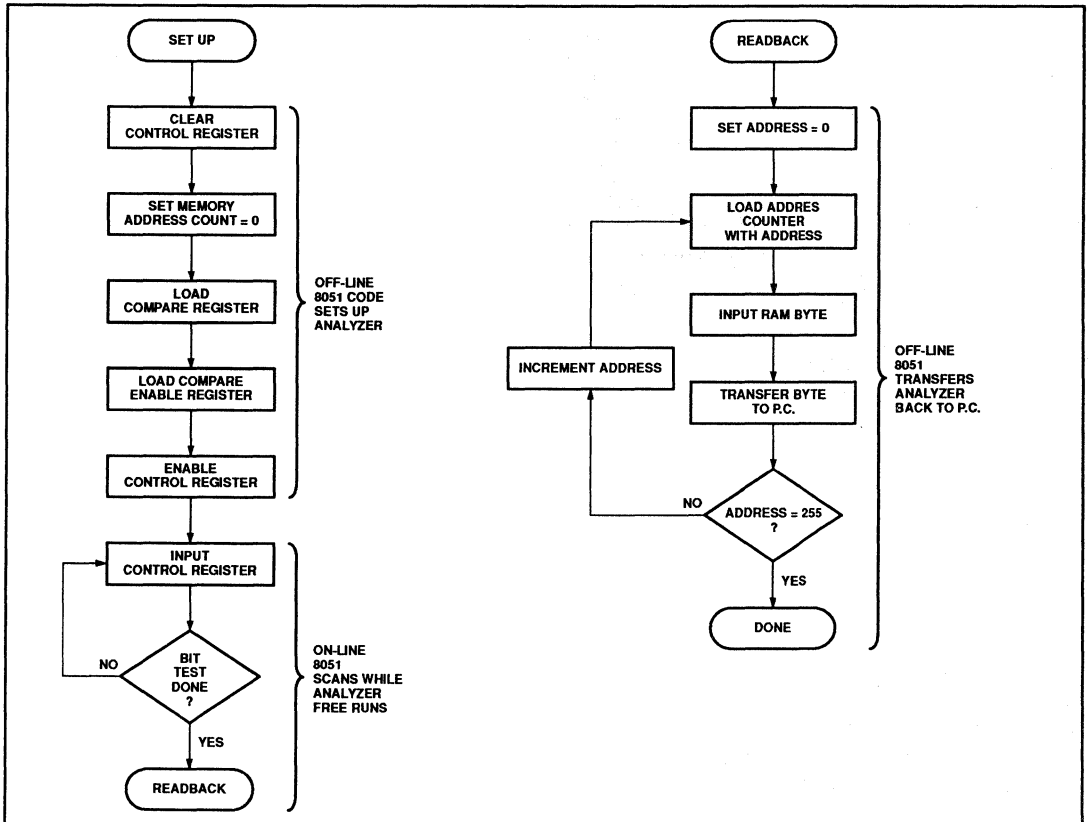


Figure 7. Simplified Analyzer Operation Flow

RESULT DISPLAY

When the data is retrieved by the PC, it can be formatted and displayed in classic logic analyzer style. Separate channels are simply the separate SRAM data lines. Display is simply a matter of finding the frame starting point, and making transition display segments on the screen.

As an example of this type of display, see Figure 8. Figure 8 is actually, a display of data taken from the simulation section of Philips Semiconductors SNAP design software, running on a PC. The simulation software in SNAP mimics the display format of a logic analyzer. By formatting the information taken from the logic analyzer and renaming the file for SNAP's needs, a usable result is obtained. The SNAP simulation display includes features like waveform expansion, cursor movement, overlap and reorder of waveforms, etc.

CONCLUSIONS

At the outset of this paper, a procedure for making accelerators was outlined. By example, a small logic analyzer design was shown, to speed up an inexpensive microcontroller, with a low cost FPGA.

But why is this class of designs so important? First, the logic analyzer shows which activities must be made in hardware and which can be in software. Second, the design is typical of a larger class of designs – parallel data operation designs. Included in this category are other instrumentation applications and many computer control applications. For instance, if the logic analyzer had the triggering hardware removed and only passed the data through it to the SRAM, it becomes a generalized data acquisition unit. If the data acquisition unit was supplemented with a single 8 bit A/D converter, and its analog input was taken as a probe, it becomes a digital oscilloscope. If the data acquisition unit passes data to the SRAM, and adds a word counter, the unit becomes a DMA controller. Other examples exist. Each simply takes a high speed task, offloads the microcontroller and makes the overall operation much faster.

There are other design categories where accelerators can be beneficial. Most notable is the class of serial data operation designs. These designs can also be made with inexpensive microcontrollers and low cost FPGAs. Their primary abilities include capturing and sensing serial data patterns and taking action, as well. The world of high speed data communications uses these types of machines, accelerating lower level data operations. However, that is the topic of another paper!

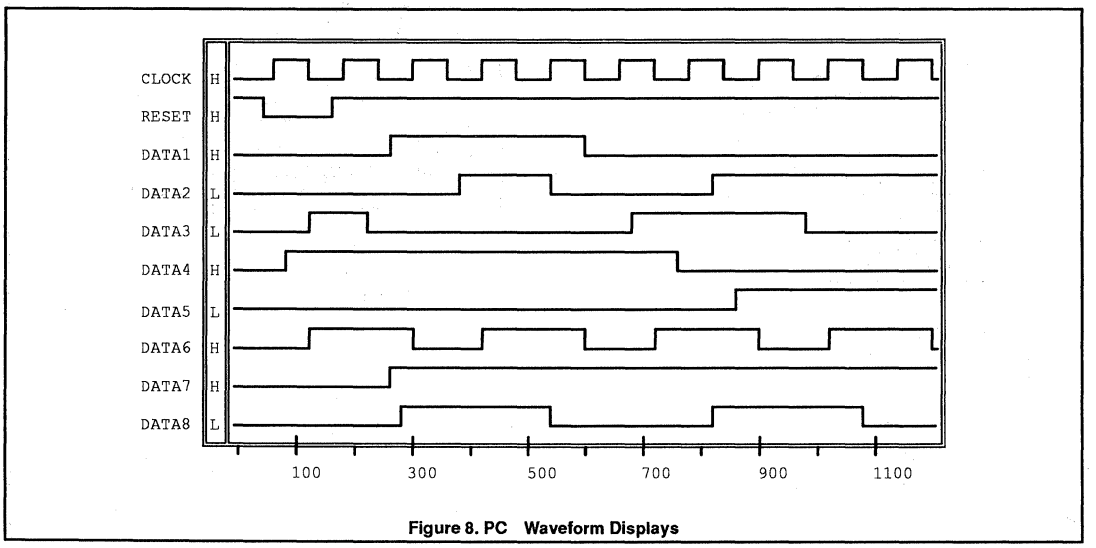


Figure 8. PC Waveform Displays

```
@PINLIST
AIN I;BIN I;CIN I;DIN I;EIN I;FIN I;GIN I;HIN I;LOAD I;
AOUT O;BOUT O;COUT O;DOUT O;EOUT O;FOUT O;GOUT O;HOUT O;
@LOGIC EQUATIONS
AOUT.D = AIN;
BOUT.D = BIN;
COUT.D = CIN;
DOUT.D = DIN;
EOUT.D = EIN;
FOUT.D = FIN;
GOUT.D = GIN;
HOUT.D = HIN;
AOUT.CLK = LOAD;
BOUT.CLK = LOAD;
COUT.CLK = LOAD;
DOUT.CLK = LOAD;
EOUT.CLK = LOAD;
FOUT.CLK = LOAD;
GOUT.CLK = LOAD;
HOUT.CLK = LOAD;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 9a. BUFF Block Equations

```
@PINLIST
INA0 I;INA1 I;INB0 I;INB1 I;INC0 I;INC1 I;IND0 I;IND1 I;
INE0 I;INE1 I;INF0 I;INF1 I;ING0 I;ING1 I;INH0 I;INH1 I;SEL I;
DOUTA O;DOUTB O;DOUTC O;DOUTD O;DOUTE O;DOUTF O;DOUTG O;DOUTH O;
@LOGIC EQUATIONS
DOUTA = /SEL*INA0 + SEL*INA1;
DOUTB = /SEL*INB0 + SEL*INB1;
DOUTC = /SEL*INC0 + SEL*INC1;
DOUTD = /SEL*IND0 + SEL*IND1;
DOUTE = /SEL*INE0 + SEL*INE1;
DOUTF = /SEL*INF0 + SEL*INF1;
DOUTG = /SEL*ING0 + SEL*ING1;
DOUTH = /SEL*INH0 + SEL*INH1;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 9b. MUX Block Equations

```
@PINLIST
DB0 B;DB1 B;DB2 B;DB3 B;DB4 B;DB5 B;DB6 B;DB7 B;
NDB0 O;NDB1 O;NDB2 O;NDB3 O;NDB4 O;NDB5 O;NDB6 O;NDB7 O;
DIN0 I;DIN1 I;DIN2 I;DIN3 I;DIN4 I;DIN5 I;DIN6 I;DIN7 I;
ENA I;
@LOGIC EQUATIONS
DB0 = DIN0;
DB1 = DIN1;
DB2 = DIN2;
DB3 = DIN3;
DB4 = DIN4;
DB5 = DIN5;
DB6 = DIN6;
DB7 = DIN7;
DB0.OE = ENA;
DB1.OE = ENA;
DB2.OE = ENA;
DB3.OE = ENA;
DB4.OE = ENA;
DB5.OE = ENA;
DB6.OE = ENA;
DB7.OE = ENA;
NDB0 = DB0;
NDB1 = DB1;
NDB2 = DB2;
NDB3 = DB3;
NDB4 = DB4;
NDB5 = DB5;
NDB6 = DB6;
NDB7 = DB7;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 9c. XCVRS Block Equations

```
@PINLIST
COUNT I; LOAD I; D0 I;D1 I;D2 I;D3 I;D4 I;D5 I;D6 I;D7 I; CLK I;
A O;B O; C O; D O; E O; F O; G O; H O;
@LOGIC EQUATIONS
H.J = COUNT*A*B*C*D*E*F*G + LOAD*D7;
H.K = COUNT*A*B*C*D*E*F*G + LOAD*/D7;
G.J = COUNT*A*B*C*D*E*F   + LOAD*D6;
G.K = COUNT*A*B*C*D*E*F   + LOAD*/D6;
F.J = COUNT*A*B*C*D*E     + LOAD*D5;
F.K = COUNT*A*B*C*D*E     + LOAD*/D5;
E.J = COUNT*A*B*C*D       + LOAD*D4;
E.K = COUNT*A*B*C*D       + LOAD*/D4;
D.J = COUNT*A*B*C         + LOAD*D3;
D.K = COUNT*A*B*C         + LOAD*/D3;
C.J = COUNT*A*B           + LOAD*D2;
C.K = COUNT*A*B           + LOAD*/D2;
B.J = COUNT*A             + LOAD*D1;
B.K = COUNT*A             + LOAD*/D1;
A.J = COUNT               + LOAD*D0;
A.K = COUNT               + LOAD*/D0;
H.CLK = CLK;G.CLK = CLK;F.CLK = CLK;E.CLK = CLK;
D.CLK = CLK;C.CLK = CLK;B.CLK = CLK;A.CLK = CLK;
A.SET = 1;B.SET=1;C.SET=1;D.SET=1;E.SET=1;F.SET=1;G.SET=1;H.SET=1;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 9d. COUNT8 Block Equations

```
@PINLIST
A0 I:B0 I;A1 I;B1 I;A2 I;B2 I;A3 I;B3 I;
A4 I;B4 I;A5 I;B5 I;A6 I;B6 I;A7 I;B7 I;
ENA0 I;ENA1 I;ENA2 I;ENA3 I;
ENA4 I;ENA5 I;ENA6 I;ENA7 I;
COMPARE O;
@LOGIC EQUATIONS
EQ0 = /(A0:+:B0)+ENA0;
EQ1 = /(A1:+:B1)+ENA1;
EQ2 = /(A2:+:B2)+ENA2;
EQ3 = /(A3:+:B3)+ENA3;
EQ4 = /(A4:+:B4)+ENA4;
EQ5 = /(A5:+:B5)+ENA5;
EQ6 = /(A6:+:B6)+ENA6;
EQ7 = /(A7:+:B7)+ENA7;
COMPARE = EQ0*EQ1*EQ2*EQ3*EQ4*EQ5*EQ6*EQ7;
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANDITIONS
```

Figure 9e. COMP Block Equations

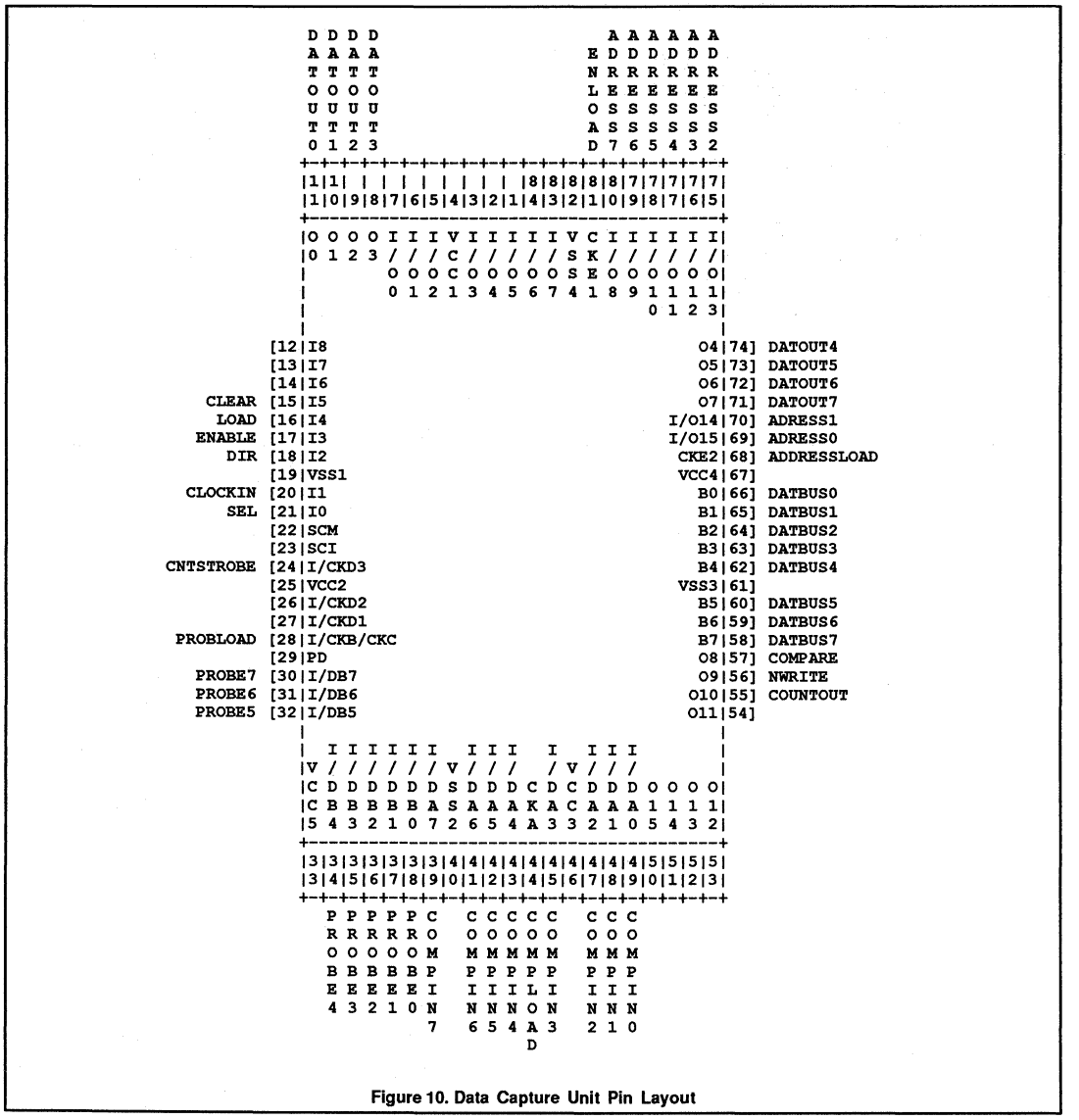


Figure 10. Data Capture Unit Pin Layout

Implementing counters in PML2X52 devices AN042

INTRODUCTION

The PML2552 and PML2852 devices contain internal clocked JK-type flip-flops that may be used to implement counters, shift registers, or synchronous state machines. JK-type flip-flops can generally be used to implement state machines and counters very efficiently. For state machines, JK flip-flop based machines require product terms (AND gates) only to transition between states. Product terms are not required to hold the state registers in specified states as they would be for D-type flip-flop implementations.

For implementing counters, JK-type flip-flops can be easily configured as T-type or toggle flip-flops by connecting the J and K inputs together. Only one product term per counter bit is required to construct a counter in this manner.

The PML2552 and PML2852 each contain two groups of ten clocked JK-type flip-flops. One group features independent clocks and independent reset inputs. The other group of ten features a common clock and a common preset input. These are indicated as JKCL552 and JKPR552 respectively in Figure 1. It is the purpose of this application note to demonstrate how, using Philips Semiconductors SNAP software, counters may be implemented most efficiently in the JKCL552 group of flip-flops. Counters of up to ten bits may be implemented using only one of the 96 foldback (FBNAND) NAND gates. This is accomplished by using the independent clocking feature of the JKCL552 flip-flops.

TYPICAL COUNTER IMPLEMENTATION

The most common way of connecting JK-type flip-flops as a counter is shown in the listing of Figure 2. This arrangement connects all clock inputs to a common clock. The least significant counter bit flip-flop has its J and K inputs tied HIGH. This flip-flop will change state after every clock pulse. The next bit flip-flop has its J and K inputs tied to the output of the least significant bit flip-flop output. If its J and K inputs are tied to the Q output of the previous flip-flop, it will toggle after a clock when the LSB is HIGH, producing an up counter. If its J and K inputs are tied to the QN output, then it will toggle when the LSB is LOW, producing a down counter. The remaining flip-flops in the counter have their J and K inputs tied to the outputs of all previous counter bits. In all Philips Semiconductors PLD devices that contain JK flip-flops, the same product term that connects to the J input may also be connected to the K input. For PML this is also true except it is a NAND gate connecting to the integral NAND gate inputs on the JK flip-flops.

This type of counter will use one foldback NAND gate (FBNAND) for each bit in the counter. It may be implemented in the JKPR552 or the JKCL552 groupings of flip-flops. Which group the counter is implemented in can be controlled in SNAP by defining an equation for the RST or SET inputs of the flip-flops. If a SET equation is defined, SNAP will try to use the JKPRFF devices. If a RST equation is supplied, SNAP will try to use the JKCLFF devices. If neither SET or RST equations are defined or both are defined, SNAP may try to make JK flip-flops totally from the FBNAND gates.

DEVICE SPECIFIC COUNTER

Although using only one FBNAND gate per counter bit is fairly efficient, a ten bit counter would require ten FBNAND gates. A design technique is available that can implement up to a ten bit counter in the JKCL552 group of flip-flops using only one FBNAND gate for the whole counter. Figure 3 contains a schematic representation of a 3-bit version . Figure 4 shows the associated SNAP simulation waveforms. As can be seen in Figure 3, all of the flip-flops JK inputs are tied HIGH. This is were the one FBNAND gate is used. All of its inputs are disconnected so that its output is LOW. The FBNANDs output is then connected to the integral NAND gates on the JK flip-flops inputs.

This counter uses a NAND gate to clock each bit in the counter. As the JKCL552 flip-flops are clocked independently by a special array of NAND gates, this design fits perfectly into the PML2552 or PML2852 devices.

Figure 5 shows SNAP equations for a 10-bit version. Figure 6 shows a SNAP resource summary for this 10-bit implementation to verify that only one FBNAND gate was used. Figure 7 shows the clock NAND array portion of a PML2552 fusetable, demonstrating again how nicely the counter may be implemented in a PML2552 or PML2852. A fusetable representation of any PLD device is available in SNAP by running FUSTABLE.EXE from the DOS command prompt.

SPECIFIC REQUIREMENTS

This counter design may be altered to provide for a count of any length. Figure 8 shows a modification to the 3-bit counter to produce a modulo-5 counter. This counter is reset to count 011. It then counts up to 111 in a binary fashion and then transitions to 011 producing 5 unique states.

Figure 9 contains the SNAP listing of another modification. This 12-bit counter was part of a design that already was using all ten of the JKPR552 flip-flops and two of the JKCL552 resources. For this case 8 of the JKCL552 flip-flops were configured as described above, the remaining four bits used D-type flip-flops realized using only FBNAND gates. An additional design technique was implemented with these D-type flip-flops.

Normally when making a counter using D-type flip-flops, the flip-flops are preceded by some AND-OR logic. In this case, since the flip-flops are constructed using only FBNAND gates, it was possible to merge the AND-OR logic structure directly into the construction of the flip-flop. For additional information on flip-flop merging, please refer to application note AN049.

CONCLUSION

Counters of up to 10-bits may be implemented in PML2552 and PML2852 devices using only one FBNAND gate and the JKCL552 group of flip-flops. The design presented may be altered to provide a count of any value. If more than 10-bits are required, it is possible to add flip-flops from the JKPR552 group, the ODFF group, or construct flip-flops from FBNAND gates.

Implementing counters in PML2X52 devices AN042

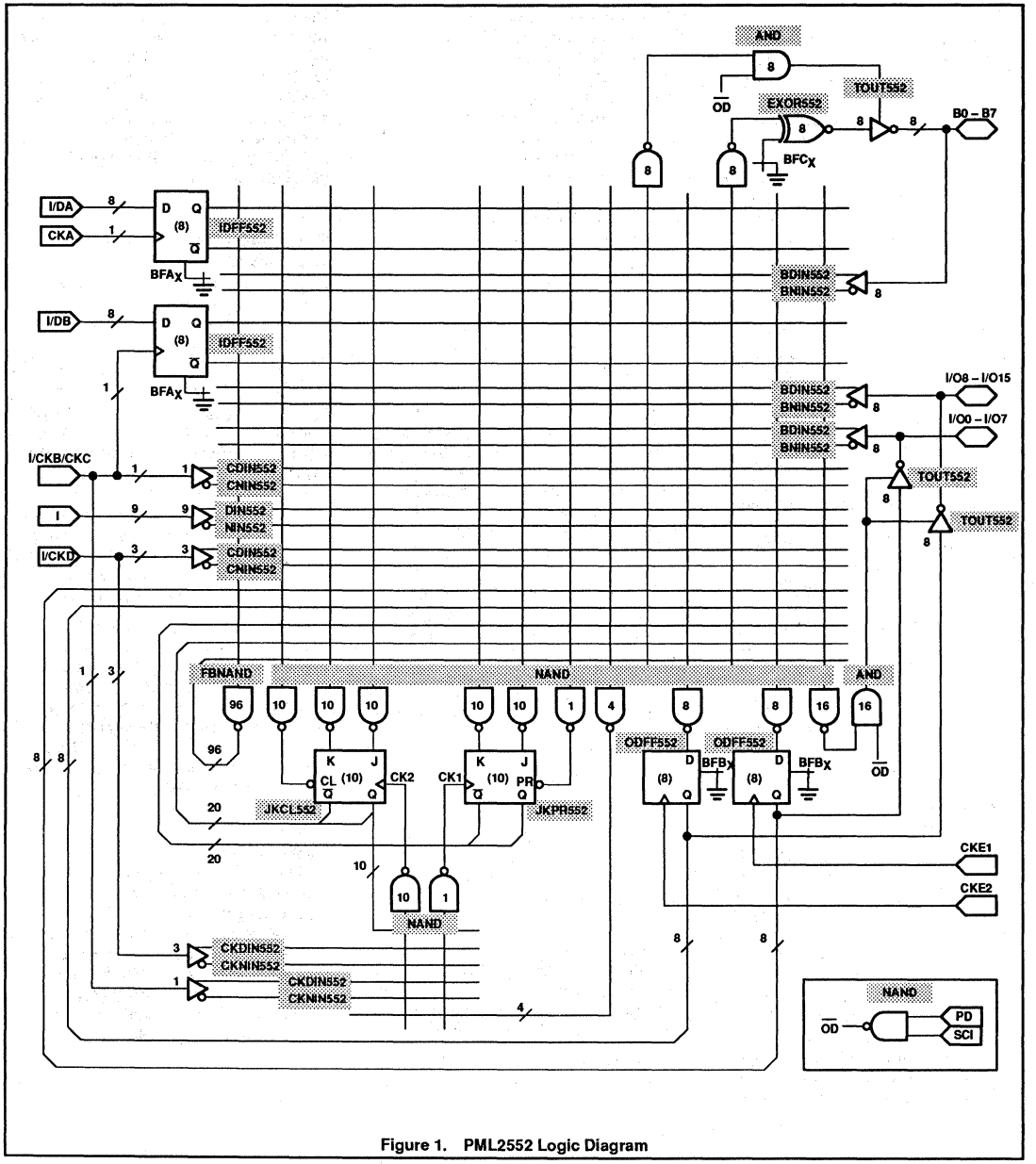
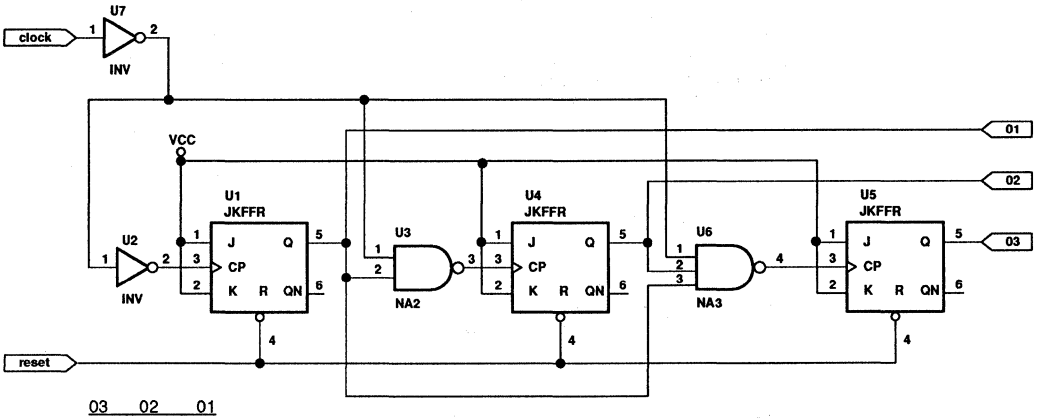


Figure 1. PML2552 Logic Diagram

```
"-----------------"
"8-bit up counter"
"-----------------"

"constructed as a down counter
  with inverted outputs"

@PINLIST
clr i; clk i;
q[7..0]     o;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS

qi[7..0].clk =  clk;          "common clock"

qi[7..0].set = /clr;          "common clear"

q0=/qi0;  q1=/qi1;  q2=/qi2; "invert outputs to"
q3=/qi3;  q4=/qi4;  q5=/qi5; "make an up counter"
q6=/qi6;  q7=/qi7;

qi0.j = 1;
qi0.k = 1;
qi1.j = /qi0;
qi1.k = /qi0;
qi2.j = /qi0*/qi1;
qi2.k = /qi0*/qi1;
qi3.j = /qi0*/qi1*/qi2;
qi3.k = /qi0*/qi1*/qi2;
qi4.j = /qi0*/qi1*/qi2*/qi3;
qi4.k = /qi0*/qi1*/qi2*/qi3;
qi5.j = /qi0*/qi1*/qi2*/qi3*/qi4;
qi5.k = /qi0*/qi1*/qi2*/qi3*/qi4;
qi6.j = /qi0*/qi1*/qi2*/qi3*/qi4*/qi5;
qi6.k = /qi0*/qi1*/qi2*/qi3*/qi4*/qi5;
qi7.j = /qi0*/qi1*/qi2*/qi3*/qi4*/qi5*/qi6;
qi7.k = /qi0*/qi1*/qi2*/qi3*/qi4*/qi5*/qi6;

@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 2. Typical 8-Bit Binary Counter

Implementing counters in PML2X52 devices AN042



03	02	01
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Figure 3. 3-Bit Binary Counter (modulo-8)

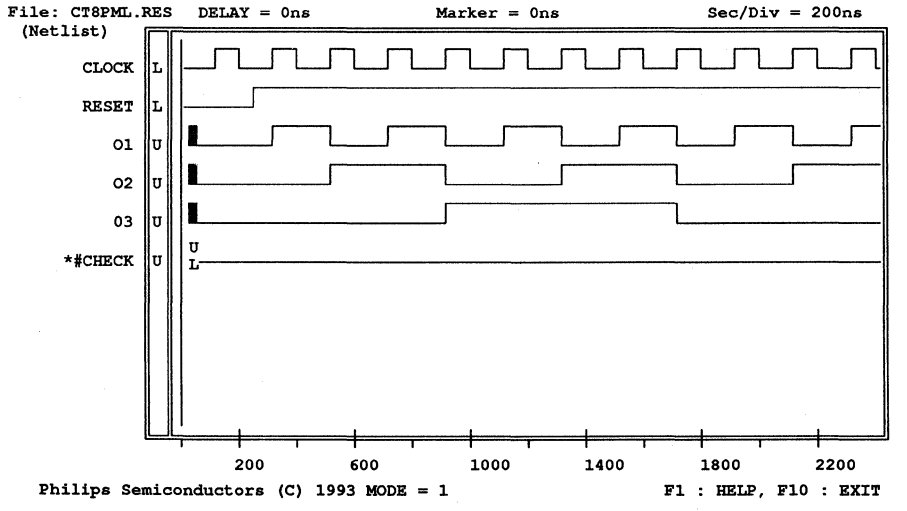


Figure 4. SNAP Simulation Waveforms of 3-Bit Binary Counter

```
"Ten bit counter using only NAND gates in clock array
 of JK flip-flops with reset in PML2552 or PML2852"

@PINLIST
clr       i;
clk       i;
q[9..0]   o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
qi0.clk  = clk;                                    "JKCL552 flip-flops have"
qi1.clk  = /(qi0*/clk);                            "an independent clock    "
qi2.clk  = /(qi0*qi1*/clk);                        "NAND array              "
qi3.clk  = /(qi0*qi1*qi2*/clk);
qi4.clk  = /(qi0*qi1*qi2*qi3*/clk);          "clock on rising edge of clk"
qi5.clk  = /(qi0*qi1*qi2*qi3*qi4*/clk);
qi6.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*/clk);
qi7.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*qi6*/clk);
qi8.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*qi6*qi7*/clk);
qi9.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*qi6*qi7*qi8*/clk);

qi0.rst =  /clr; qi1.rst =  /clr; qi2.rst =  /clr;  "define reset so SNAP"
qi3.rst =  /clr; qi4.rst =  /clr; qi5.rst =  /clr;  "will use JKCL552 devices"
qi6.rst =  /clr; qi7.rst =  /clr; qi8.rst =  /clr;
qi9.rst =  /clr;

q0=qi0; q1=qi1; q2=qi2; q3=qi3; q4=qi4;
q5=qi5; q6=qi6; q7=qi7; q8=qi8; q9=qi9;

qi[9..0].j = 1;                                     "tie J and K inputs HIGH"
qi[9..0].k = 1;
```

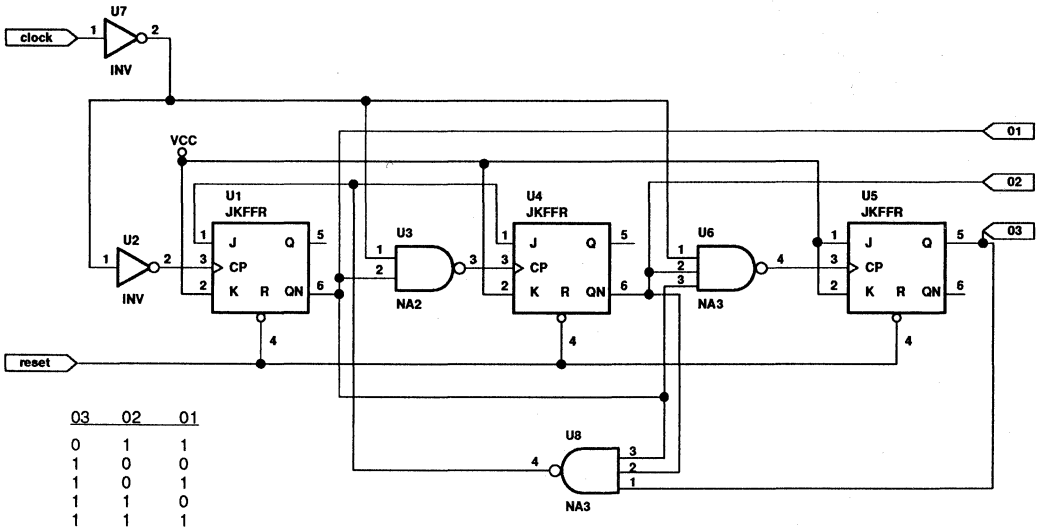
Figure 5. SNAP Equations for a 10-Bit Counter

```
Design from  CT256PML.N2 - for device PML2552
   Cell name    used/total      %
=========================================
    CKDIN552      0 /    4      0%
    CKNIN552      1 /    4     25%
      FBNAND      1 /   96      1%
        NAND     51 /  104     49%
      DIN552      1 /   25      4%
      NIN552      0 /   25      0%
     CDIN552      0 /    4      0%
     CNIN552      0 /    4      0%
       CK552      0 /    4      0%
     IDFF552      0 /   16      0%
     BDIN552      0 /   24      0%
     BNIN552      0 /   24      0%
     JKCL552     10 /   10    100%
     JKPR552      0 /   10      0%
     EXOR552      8 /    8    100%
     TOUT552     10 /   24     41%
     ODFF552      2 /   16     12%
```

Figure 6. SNAP Resource Summary

```
>> Print out of fusemap starting from col 1 to col 18 <<

                IIIIMMMMJJJJJJJJJJ
                ////AAAAKKKKKKKKKK
                CCCCIIIICCCCCCCCCC
                KKKKNNNNLLLLLLLLLL
                B(((____0123456789
                /123CCCC__________
                C)))KKKKQQQQQQQQQQ
                ____((((||||||||||
                22220123||||||||||
                4320))))||||||||||
                ||||||||||||||||||
                ||||||||||||||||||
                VVVVVVVVVVVVVVVVVV
   JKPR_CK-> ----..............
  JKCL0_CK-> ---L.....AAAAAAAAA
  JKCL1_CK-> ---L......AAAAAAAA
  JKCL2_CK-> ---L.......AAAAAAA
  JKCL3_CK-> ---L........AAAAAA
  JKCL4_CK-> ---L.........AAAAA
  JKCL5_CK-> ---L..........AAAA
  JKCL6_CK-> ---L...........AAA
  JKCL7_CK-> ---L............AA
  JKCL8_CK-> ---L.............A
  JKCL9_CK-> ---L..............
```

Figure 7. Portion of PML2552 Fuse Table Showing Clock Array Connections



03	02	01
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Figure 8a. Modulo-5 Counter

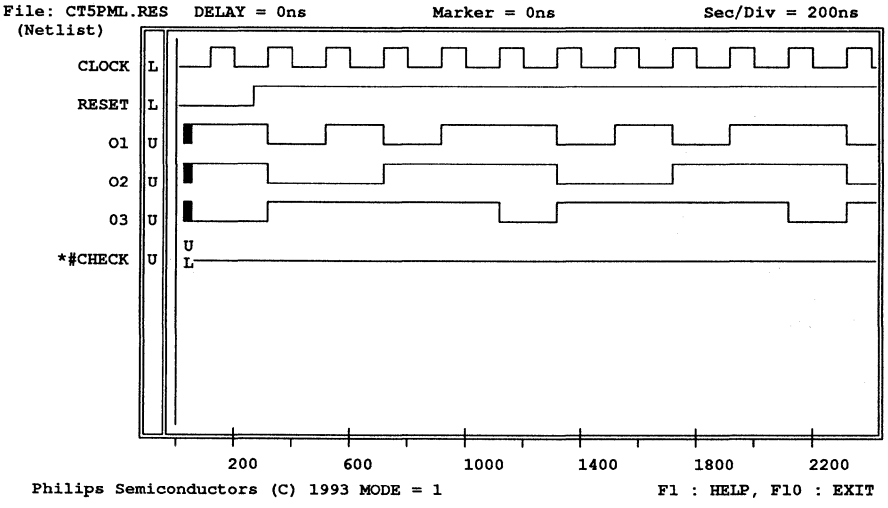


Figure 8b. SNAP Simulation Waveforms of Modulo-5 Counter

```
" This example is a 12-bit binary up counter. It is
  constructed using 8 JKCL552 flip-flops and 4 D-type
  flip-flops constructed from FBNAND gates.
"
@PINLIST
clr       i;
clkn      i;
q[11..0]  o;
@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
qi0.clk  = /clkn;
qi1.clk  = /(qi0*clkn);              "clock on falling edge of clkn"
qi2.clk  = /(qi0*qi1*clkn);
qi3.clk  = /(qi0*qi1*qi2*clkn);
qi4.clk  = /(qi0*qi1*qi2*qi3*clkn);
qi5.clk  = /(qi0*qi1*qi2*qi3*qi4*clkn);
qi6.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*clkn);
qi7.clk  = /(qi0*qi1*qi2*qi3*qi4*qi5*qi6*clkn);
qi0.rst =  /clr; qi1.rst =  /clr; qi2.rst =  /clr;     "define reset inputs"
qi3.rst =  /clr; qi4.rst =  /clr; qi5.rst =  /clr;
qi6.rst =  /clr; qi7.rst =  /clr; qi8.rst =  /clr;
qi9.rst =  /clr; qi10.rst = /clr; qi11.rst = /clr;
q0=qi0; q1=qi1; q2=qi2; q3=qi3; q4=qi4;
q5=qi5; q6=qi6; q7=qi7;
q8=qi8;
q9=qi9;
q10=qi10;
q11=qi11;
qi[7..0].j = 1;                      "all J and K inputs are tied HIGH"
qi[7..0].k = 1;
"Four bit D-type flop counter clocked from output of qi7"
qi8  =/(qi8n*f8sn);           "LSB"
qi8n =/(qi8*f8rn*/clr);
f8sn =/(f8snn*/qi7*/clr);
f8snn=/(f8sn*f8rnn);
f8rn =/(f8rnn*f8sn*/qi7);
f8rnn=/(f8rn*qi8n*/clr);      "qi8.d = /qi8;"
qi9  =/(qi9n*f9sn);           "LMSB"
qi9n =/(qi9*f9rn*/clr);
f9sn =/(f9snn*/qi7*/clr);
f9snn=/(f9sn*f9rnna*f9rnnb);
f9rn =/(f9rnna*f9rnnb*f9sn*/qi7); "merge AND/OR into flop"
f9rnna=/(f9rn*/clr*qi8n*qi9);     "qi9.d = /qi8*qi9+qi8*/qi9;"
f9rnnb=/(f9rn*/clr*qi8*qi9n);
qi10  =/(qi10n*f10sn);        "UMSB"
qi10n =/(qi10*f10rn*/clr);
f10sn =/(f10snn*/qi7*/clr);
f10snn=/(f10sn*f10rnna*f10rnnb*f10rnnc);
f10rn =/(f10rnna*f10rnnb*f10rnnc*f10sn*/qi7); "merge AND/OR into flop"
f10rnna=/(f10rn*/clr*qi8n*qi10);  "qi10.d=/qi8*qi10+/qi9*qi10+qi8*qi9*/qi10;"
f10rnnb=/(f10rn*/clr*qi9n*qi10);
f10rnnc=/(f10rn*/clr*qi8*qi9*qi10n);
qi11  =/(qi11n*f11sn);        "MSB"
qi11n =/(qi11*f11rn*/clr);
f11sn =/(f11snn*/qi7*/clr);
f11snn=/(f11sn*f11rnna*f11rnnb*f11rnnc*f11rnnd);
f11rn =/(f11rnna*f11rnnb*f11rnnc*f11rnnd*f11sn*/qi7);
      "merge AND/OR into flop"
      "qi11.d=/qi8*qi11+/qi9*qi11+/q10*qi11+qi8*qi9*qi10*/qi11;"
f11rnna=/(f11rn*/clr*qi8n*qi11);
f11rnnb=/(f11rn*/clr*qi9n*qi11);
f11rnnc=/(f11rn*/clr*qi10n*qi11);
f11rnnd=/(f11rn*/clr*qi8*qi9*qi10*qi11n);
@INPUT VECTORS
@OUTPUT VECTORS
@STATE VECTORS
@TRANSITIONS
```

Figure 9. 12-Bit Counter

Serial data encoder and decoder AN041

Authors: Uwe Krüger and Jürgen Meixner

INTRODUCTION

Serial digital data encoding and decoding protocols are often used to increase the reliability of data transmission and storage. Transmitting digital data permits reliable error detection and correction techniques to be applied. Some encoding formats allow the transmitting clock to be extracted from the data stream and reproduced at the receiver. Long sequences of binary 0s or 1s may be transparently limited using formats such as HDBn or Manchester.

Encoded serial signals are used, for instance, on the S0–bus in ISDN user interfaces (inverse AMI) and in PCM systems (HDB3 encoding).

This application note describes circuits that can encoder and decode eight of the most frequently used formats. A programmable logic device, the Philips PML2552, was used to implement the encoder. Another PML2552 was used to implement the decoder function. Three inputs on both devices permit selection of the desired encoding or decoding method.

The decoder will flag transmitting errors based upon the specific coding rules of each format.

This design is part of a testing device for cables and encoded serial data signals.

Serial Encoding Formats

This is a short explanation of the eight implemented codes:

AMI – Code:
(alternate mark inversion) A binary 1 is transmitted alternating as a high mark (+1, positive voltage level) and as a low mark (–1, negative voltage level), a binary 0 appears as space (0 Volt).

HDBn – Code:
(high density bipolar n) after n spaces will be included a code violation, i.e. the following "0" is transmitted as high mark or low mark depending on previous "1": Was it sent as a high mark, the violation will be also a high mark. So a binary 0 is turned into a "wrong" mark that can be recognized and corrected by the receiver. A binary 1 after these n spaces will be sent as the alternating mark.

IAMI – Code:
(inverse AMI) same as AMI code, but binary 0 and 1 are swapped.

RZ – Code:
(return–to–zero), a binary 1 is transmitted as high mark returning to space after the half step time.

NRZ – Code:
(non–return–to–zero), it's the same as the input of the coder, here will be amplified the signal amplitude only.

Manchester – Code:
binary 1 is transmitted as transition between high mark and low mark, binary 0 switches from low mark to high mark. This code is free of any direct voltage, but requires the double bandwidth.

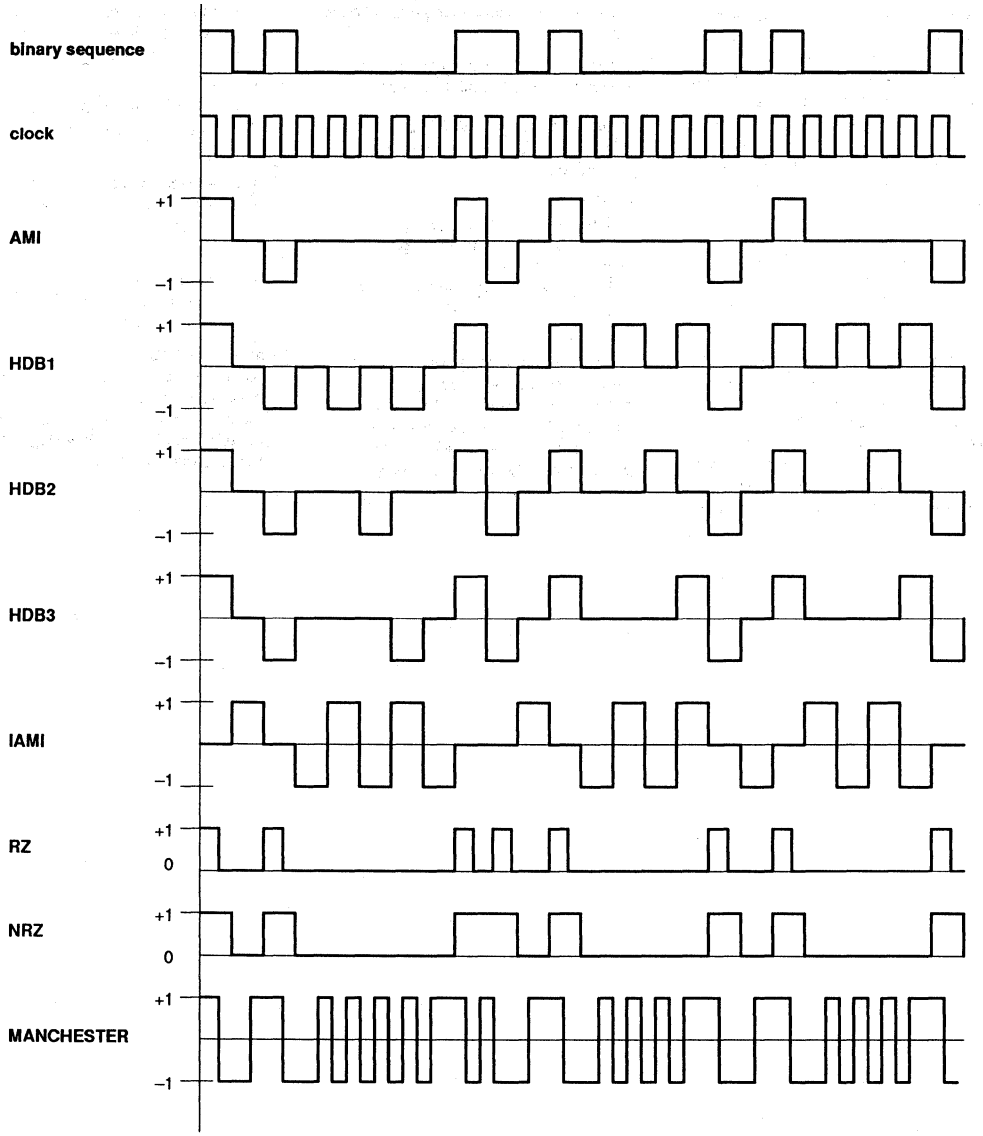


Figure 1. Timing Diagrams of Implemented Serial Encoding Methods

Serial data encoder and decoder AN041

ENCODER

Realizing the Encoder Using a PML2552

Description of Encoder Facilities
The encoder is designed as a Moore–type finite state machine. This structure of state machine ensures the output signals transition synchronously with the clock. Using a clock generated from a quartz oscillator, the data link can be provided on only one line, however, in this case the receiver must have extra circuitry such as a phase locked loop to extract the clock from the incoming data stream. The receiver must also be able to synchronize itself to the transmitter when the transmitter sends a special data pattern.

Three additional pins are implemented in this design: Single and double bit clock (CP und CP2) and a signal for synchronization (/SYNC) of external devices. These pins are used in the testing device to trigger an oscilloscope and a pattern generator.

To make a bipolar signal, the outputs Y[2,1] of the encoder must be connected with a subtractor realized with an OP–AMP. A schematic of the complete encoder circuit is shown in Figure 15. The following table shows dependencies between the encoder output and the output of the subtractor:

	Lowmark	Space	Highmark
Y[2,1]	10	00	01

To run the encoder, pin CP4 should be connected to an oscillator running at quadruple the bit clock. After resetting by a LOW pulse on RESET, the encoder generates the synchronizing signal (SYNC). Outputs Y[2,1] are valid upon the falling edges of the clock (CP) after the /SYNC pulse. Refer to Figure 6 for timing information.

Code will be chosen by a binary combination on pins S[2..0]:

Code	AMI	HDB1	HDB2	HDB3	IAMI	RZ	NRZ	Manch.
S[2..0]	000	001	010	011	100	101	110	111

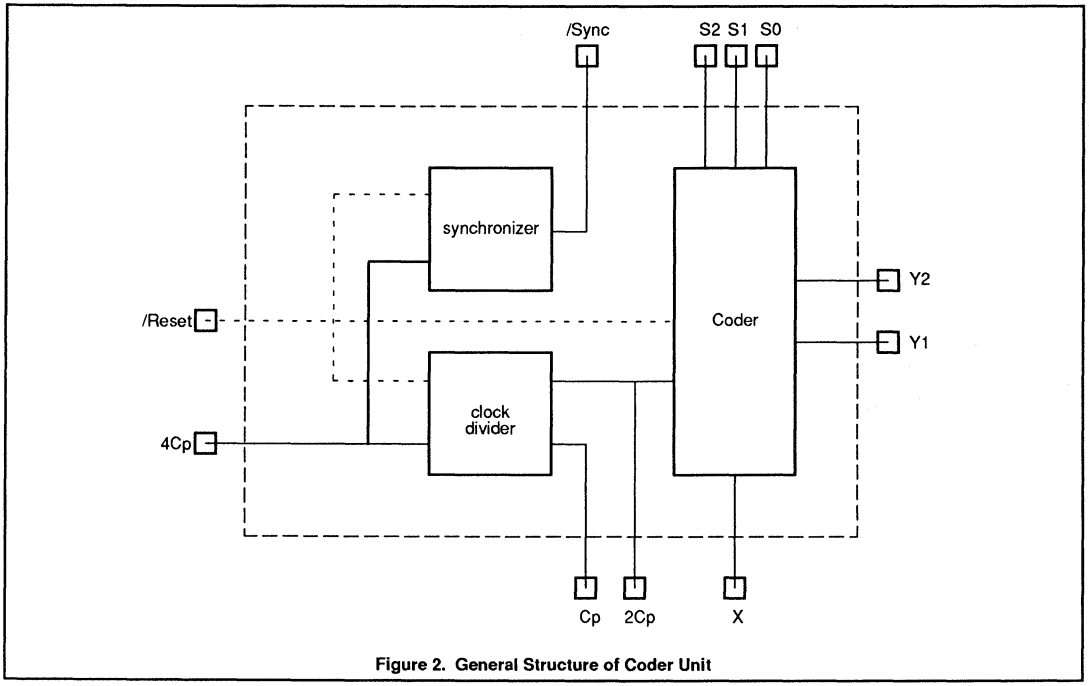


Figure 2. General Structure of Coder Unit

Structure of Encoder State Machine
The finite state machine implementing the encoder has a total of nineteen states. Due to the complexity of a complete state diagram, and the size of this page, only a partial state diagram is shown in Figure 3. The state diagram of Figure 3 shows the encoding technique for the Alternate Mark Inversion (AMI) format. Operation of the complete state machine may be analyzed by studying the @TRANSITIONS section in the CODER.EQN–file shown in Figure 5.

Structure of Synchronizer State Machine
The synchronizer was also realized as a Moore machine. A state diagram is shown in Figure 4. RESET initializes the state machine to State 0. After four clock pulses, the machine will be in State 4 and output SYNC is forced LOW. SYNC stays LOW for one more clock pulse, while in State 5, and returns HIGH when in State 6. The state machine will stay in State 6 until another RESET occurs.

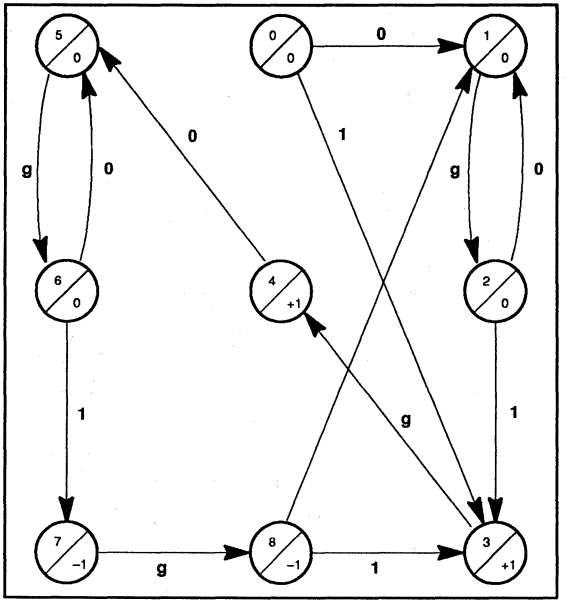


Figure 3. State Diagram of AMI Encoding Technique

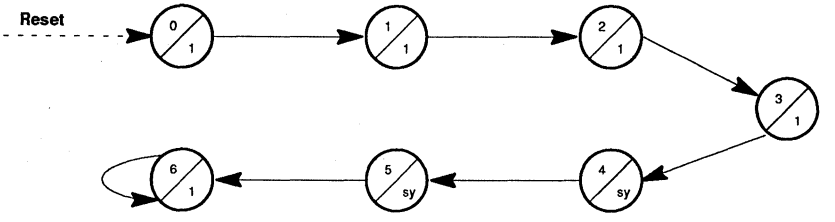


Figure 4. State Diagram of Synchronizer

Listing of Descriptive File CODER.EQN

```
@PINLIST
s0      i; s1      i; s2      i;
x       i;
cp4     i;
reset   i;

y1      o; y2      o;
sync    o;
cp2     o; cp      o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
sy_r = /(/reset * cp4 * zt1);

zt1.clk = cp4; zt1.j = /zt1; zt1.k = zt1; zt1.rst = /((zt1+zt2)*/reset);
zt2.clk = zt1; zt2.j = /zt2; zt2.k = zt2; zt2.rst = /((zt1+zt2)*/reset);

cp2 = zt1; cp = zt2;

zs1.clk = cp4; zs2.clk = cp4; zs3.clk = cp4;
zs1.rst = sy_r; zs2.rst = sy_r; zs3.rst = sy_r;

zz4.clk = zt1; zz3.clk = zt1; zz2.clk = zt1; zz1.clk = zt1; zz0.clk = zt1;
zz4.rst = sy_r; zz3.rst = sy_r; zz2.rst = sy_r; zz1.rst = sy_r; zz0.rst = sy_r;

@INPUT VECTORS
[s2, s1, s0]
ami  = 000 B; hdb1 = 001 B; hdb2 = 010 B; hdb3 = 011 B;
iami = 100 B; rz   = 101 B; nrz  = 110 B; man  = 111 B;

[x]
x0 = 0 B; x1 = 1 B;

@OUTPUT VECTORS
[y2, y1]
highm = 01 B; null = 00 B; lowm = 10B;

[sync]
sy = 0 B; nsy = 1 B;

@STATE VECTORS
[zz4, zz3, zz2, zz1, zz0] jkffr
z0  = 00000 B; z1  = 00001 B; z2  = 00010 B; z3  = 00011 B;
z4  = 00100 B; z5  = 00101 B; z6  = 00110 B; z7  = 00111 B;
z8  = 01000 B; z9  = 01001 B; z10 = 01010 B; z11 = 01011 B;
z12 = 01100 B; z13 = 01101 B; z14 = 01110 B; z15 = 01111 B;
z16 = 10000 B; z17 = 10001 B; z18 = 10010 B;

[zs3, zs2, zs1] jkffr
sy0 = 000 B; sy1 = 001 B; sy2 = 010 B; sy3 = 011 B;
sy4 = 100 B; sy5 = 101 B; sy6 = 110 B;

@TRANSITIONS
while[z0] with[null]
          if[] then[z1]
while[z1] with[null]
          if[] then[z2]
while[z2] with[null]
          if[(iami*x1)+((ami+hdb1+hdb2+hdb3)*x0)]          then[z5]
          if[man*x0]                                       then[z4]
          if[(man+nrz)*x1]                                 then[z3]
          if[(iami*x0)+((ami+rz+hdb1+hdb2+hdb3)*x1)]       then[z12]
          if[(nrz+rz)*x0]                                  then[z13]
while[z3] with[highm]
          if[((man+iami)*x0)+((ami+hdb1+hdb2+hdb3)*x1)]    then[z4]
          if[man*x1]                                       then[z11]
          if[nrz]                                          then[z12]
          if[((ami+hdb1+hdb2+hdb3)*x0)+(iami*x1)]          then[z13]
while[z4] with[lowm]
          if[ami+hdb1+hdb2+hdb3+iami]                      then[z11]
          if[man*x0]                                       then[z12]
```

Figure 5. Description File CODER.EQN (1 of 2)

```
while[z5] with[null]
          if[ami+hdb1+hdb2+hdb3+iami]                    then[z6]
while[z6] with[null]
          if[(hdb2+hdb3)*x0]                             then[z7]
          if[hdb1*x0]                                    then[z4]
          if[(ami*x0)+(iami*x1)]                         then[z5]
          if[((ami+hdb1+hdb2+hdb3)*x1)+(iami*x0)]        then[z12]
while[z7] with[null]
          if[]                                           then[z8]
while[z8] with[null]
          if[hdb3*x0]                                    then[z9]
          if[hdb2*x0]                                    then[z4]
          if[(hdb2+hdb3)*x1]                             then[z12]
while[z9] with[null]
          if[]                                           then[z10]
while[z10] with[null]
          if[hdb3*x0]                                    then[z4]
          if[hdb3*x1]                                    then[z12]
while[z11] with[lowm]
          if[((ami+hdb1+hdb2+hdb3)*x0)+(iami*x1)]        then[z5]
          if[man*x0]                                     then[z4]
          if[man*x1]                                     then[z3]
          if[((ami+hdb1+hdb2+hdb3)*x1)+(iami*x0)]        then[z12]
while[z12] with[highm]
          if[man*x0]                                     then[z4]
          if[rz+(nrz*x0)]                                then[z13]
          if[ami+hdb1+hdb2+hdb3+iami+((nrz+man)*x1)]     then[z3]
while[z13] with[null]
          if[rz*x0]                                      then[z13]
          if[rz*x1]                                      then[z12]
          if[ami+hdb1+hdb2+hdb3+iami+nrz]                then[z14]
while[z14] with[null]
          if[((ami+hdb1+hdb2+hdb3)*x1)+(iami*x0)]        then[z4]
          if[nrz*x1]                                     then[z3]
          if[hdb1*x0]                                    then[z12]
          if[(hdb2+hdb3)*x0]                             then[z15]
while[z15] with[null]
          if[]                                           then[z16]
while[z16] with[null]
          if[hdb2*x0]                                    then[z12]
          if[(hdb2+hdb3)*x1]                             then[z4]
          if[hdb3*x0]                                    then[z17]
while[z17] with[null]
          if[]                                           then[z18]
while[z18] with[null]
          if[hdb3*x0]                                    then[z12]
          if[hdb3*x1]                                    then[z4]

while[sy0] with[nsy] if[] then[sy1]
while[sy1] with[nsy] if[] then[sy2]
while[sy2] with[nsy] if[] then[sy3]
while[sy3] with[nsy] if[] then[sy4]
while[sy4] with[sy]  if[] then[sy5]
while[sy5] with[sy]  if[] then[sy6]
while[sy6] with[nsy] if[] then[sy6]
```

Figure 5. Description File CODER.EQN (2 of 2)

Timing simulation

Timing Diagram of Synchronizer

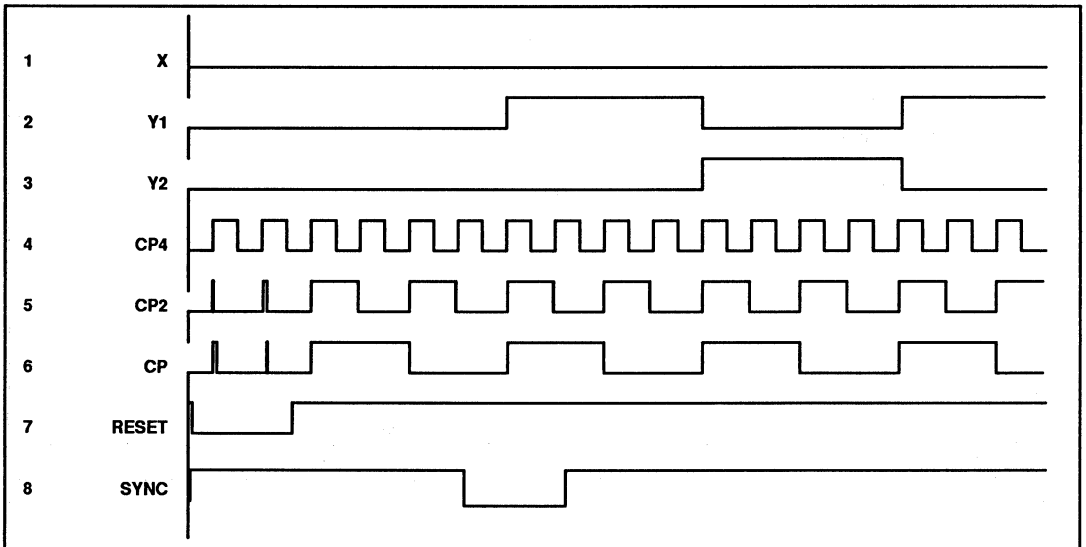


Figure 6. Timing Diagram of Generating the /SYNC Signal

Listings of Stimuli Files for Timing Simulation Using LESIM

```
*************************************************
*     Output of Waveform Version 1.90           *
* Date: 07/19/93                  Time: 09:56:50 *
*************************************************
*                                               *
* Input File Name   : AMI.SCL                   *
* Rule File Name    : Scl Rule                  *
* Output File Name  : AMI.SCL                   *
*                                               *
*************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 0 (200000) S0
 S 0 (200000) S1
 S 0 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 7. Stimulus File AMI.SCL, Control Word On Inputs S[2..0]= 000B

```
*************************************************
*     Output of Waveform Version 1.90           *
* Date: 07/19/93                  Time: 09:56:50 *
*************************************************
*                                               *
* Input File Name   : HDB1.SCL                  *
* Rule File Name    : Scl Rule                  *
* Output File Name  : HDB1.SCL                  *
*                                               *
*************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 1 (200000) S0
 S 0 (200000) S1
 S 0 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 8. Stimulus File HDB1.SCL, Control Word On Inputs S[2..0]= 001B

```
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                  Time: 09:56:50 *
**************************************************
*                                                *
* Input File Name   : HDB2.SCL                   *
* Rule File Name    : Scl Rule                   *
* Output File Name  : HDB2.SCL                   *
*                                                *
**************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 0 (200000) S0
 S 1 (200000) S1
 S 0 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 9. Stimulus File HDB2.SCL, Control Word On Inputs S[2..0] = 010 B

```
**************************************************
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                  Time: 09:56:50 *
**************************************************
*                                                *
* Input File Name   : HDB3.SCL                   *
* Rule File Name    : Scl Rule                   *
* Output File Name  : HDB3.SCL                   *
*                                                *
**************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 1 (200000) S0
 S 1 (200000) S1
 S 0 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 10. Stimulus File HDB3.SCL, Control Word On Inputs S[2..0] = 011 B

```
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                  Time: 09:56:50 *
**************************************************
*                                                *
* Input File Name   : INV_AMI.SCL                *
* Rule File Name    : Scl Rule                   *
* Output File Name : INV_AMI.SCL                 *
*                                                *
**************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 0 (200000) S0
 S 0 (200000) S1
 S 1 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 11. Stimulus File INV_AMI.SCL, Control Word On Inputs S[2..0] = 100 B

```
**************************************************
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                  Time: 09:56:50 *
**************************************************
*                                                *
* Input File Name   : RZ.SCL                     *
* Rule File Name    : Scl Rule                   *
* Output File Name : RZ.SCL                      *
*                                                *
**************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 1 (200000) S0
 S 0 (200000) S1
 S 1 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 12. Stimulus File RZ.SCL, Control Word On Inputs S[2..0] = 101 B

```
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                    Time: 09:56:50 *
*************************************************
*                                                 *
* Input File Name   : NRZ.SCL                     *
* Rule File Name    : Scl Rule                    *
* Output File Name  : NRZ.SCL                     *
*                                                 *
*************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 0 (200000) S0
 S 1 (200000) S1
 S 1 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 13. Stimulus File NRZ.SCL, Control Word On Inputs S[2..0] = 110 B

```
*************************************************
*     Output of Waveform Version 1.90           *
* Date: 07/19/93                    Time: 09:56:50 *
*************************************************
*                                                 *
* Input File Name   : MANCHEST.SCL                *
* Rule File Name    : Scl Rule                    *
* Output File Name  : MANCHEST.SCL                *
*                                                 *
*************************************************
 P S[2..0], X, CK2, Y1, Y2, ZZ[4..0], RESET
 PCO
 S 1 (200000) S0
 S 1 (200000) S1
 S 1 (200000) S2
 S 0 (20500, 24500, 28500, 36500, 52500, 64500, 104500, 112500, 116500,
# 120500, 124500, 128500) X
 S 0 (1000, 2000, ETC) CK2
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 14. Stimulus File MANCHEST.SCL, Control Word On Inputs S[2..0] = 111 B

Wiring of the Encoder

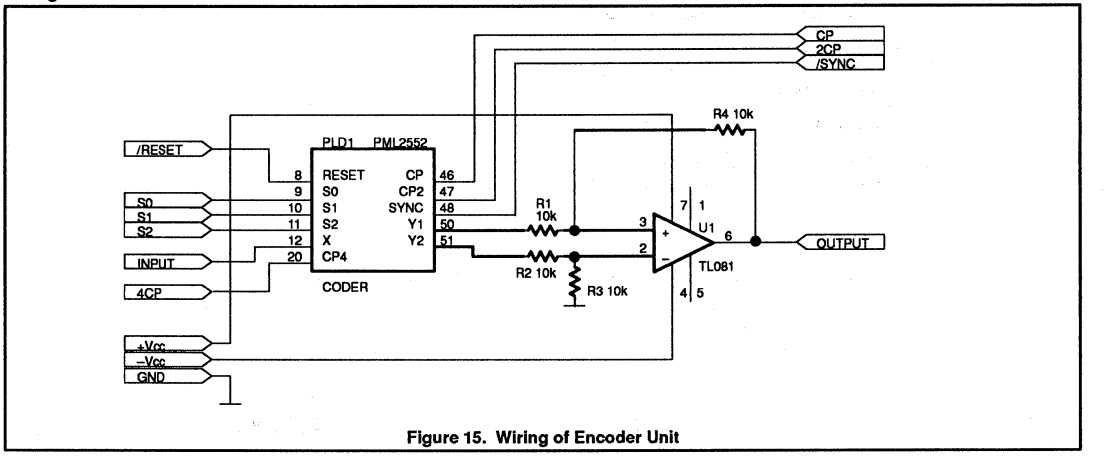


Figure 15. Wiring of Encoder Unit

DECODER

Realizing the Decoder Using a PML2552

Description of Decoder

The decoding unit is also realized as a Moore machine. The capability to recognize transmission errors can also be implemented as a state machine. So, actually, the decoding and error detection functions are combined and written as one state machine.

The SNAP state equations for the decoder and error detection functions are shown in Figure 18.

Pin 2CP of the decoder must be connected to a double bit clock. Resetting the decoder brings it in the initial state. It is necessary to convert the incoming bipolar sequence into two digital signals to be applied to the two decoder inputs. This splitter is made in the testing device by two voltage comparators as shown in Figure 18. Synchronized by a LOW pulse on the SYNC input, the decoder converts the serially encoded sequence into binary information on output Y.

	Lowmark	Space	Highmark
Y[2,1]	10	00	01

Code will be chosen by a binary combination on pins S[2..0]:

Code	AMI	HDB1	HDB2	HDB3	IAMI	RZ	NRZ	Manch.
S[2..0]	000	001	010	011	100	101	110	111

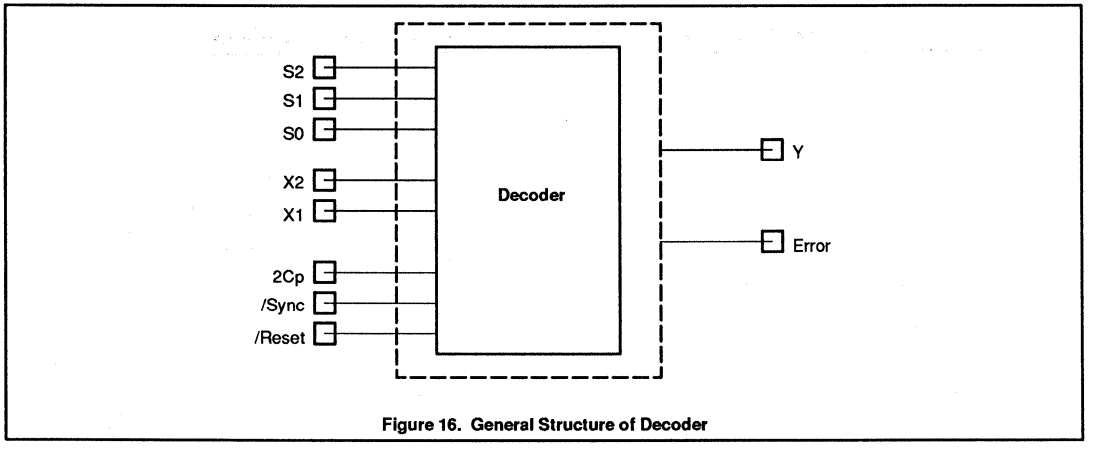


Figure 16. General Structure of Decoder

If an error should occur, the decoder will set output ERROR to a HIGH. Output Y will remain LOW until a reset and synchronization cycle begins. Using this facility, a layer II of the OSI stack can be controlled by the decoder unit and the decoder can get its RESET and SYNC inputs from layer II.

Structure of Decoders State Machine

The state diagram shown in Figure 17 shows only the AMI portion of the decoder. For detailed information on the operation of the decoder for other formats, refer to the @TRANSISTIONS section of the SNAP state equation listing shown in Figure 18.

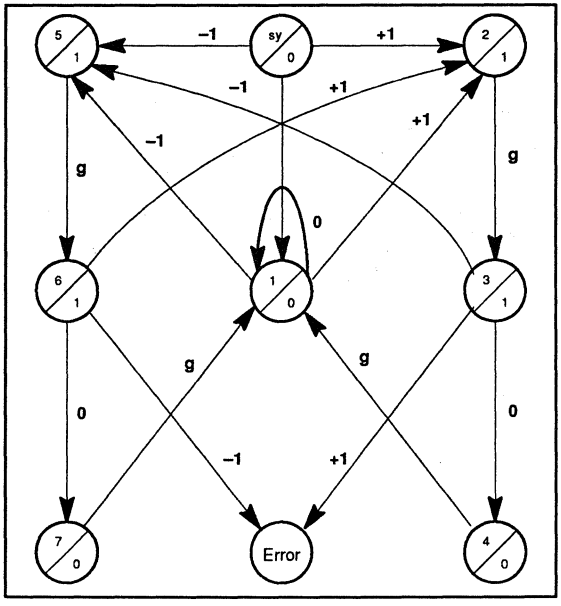


Figure 17. State Diagram of AMI Decoder

Listing of Description File DECODER.EQN

```
@PINLIST
s0      i; s1      i; s2      i;
x1      i; x2      i;
ck2     i;
reset   i;
sync    i;
y       o;
f       o;

@GROUPS
@TRUTHTABLE
@LOGIC EQUATIONS
zz5.clk = ck2; zz4.clk = ck2; zz3.clk = ck2;
zz2.clk = ck2; zz1.clk = ck2; zz0.clk = ck2;
zz5.rst = reset; zz4.set = reset; zz3.rst = reset;
zz2.rst = reset; zz1.rst = reset; zz0.rst = reset;

@INPUT VECTORS
[s2, s1, s0]
ami  = 000 B; hdb1 = 001 B; hdb2 = 010 B; hdb3 = 011 B;
iami = 100 B; rz   = 101 B; nrz  = 110 B; man  = 111 B;

[x2, x1]
highm = 11 B; lowm  = 00 B; spc   = 10 B;

[sync]
sy = 1 B; nsy = 0 B;

@OUTPUT VECTORS
[y, f]
high = 10 B; low = 00 B; err = 01 B;

@STATE VECTORS
[zz5, zz4, zz3, zz2, zz1, zz0] jkffsr
z0  = 010000 B; z1  = 000010 B; z2  = 100000 B; z3  = 000011 B;
z4  = 000100 B; z5  = 000101 B; z6  = 001011 B; z7  = 001001 B;
z8  = 010010 B; z9  = 010001 B; z10 = 000001 B; z11 = 110000 B;
z12 = 001000 B; z13 = 101000 B; z14 = 100101 B; z15 = 100100 B;
z16 = 100011 B; z17 = 100010 B; z18 = 000000 B; z19 = 101001 B;
z20 = 100001 B; z21 = 000110 B; z22 = 000111 B;

@TRANSITIONS
while[z0] with[low]
          if[sy]                                                    then[z0]
          if[nsy*((lowm*(ami+hdb1+hdb2+hdb3))+(highm*(rz+nrz+man)))] then[z10]
          if[nsy*((spc*(hdb1+hdb2+hdb3))+(highm*iami)+(lowm*man))]  then[z12]
          if[nsy*(spc*(ami+rz+nrz))]                                then[z1]
          if[nsy*(highm*(ami+hdb1+hdb2+hdb3))]                      then[z2]
          if[nsy*lowm*iami]                                         then[z4]
          if[nsy*((lowm*(rz+nrz))+(spc*man))]                       then[z18]
while[z1] with[low]
          if[spc*(ami+rz+nrz)]                                      then[z1]
          if[(lowm*ami)+(highm*(rz+nrz+man))]                       then[z10]
          if[lowm*man]                                              then[z12]
          if[highm*ami]                                             then[z2]
          if[(lowm*(rz+nrz))+(spc*man)]                             then[z18]
while[z2] with[high]
          if[ami+hdb1+hdb2+hdb3+iami]                               then[z3]
while[z3] with[high]
          if[lowm*(ami+hdb1+hdb2+hdb3)]                             then[z10]
          if[(spc*(ami+hdb1+hdb2+hdb3))+(iami*lowm)]                then[z4]
          if[highm*(ami+hdb1+hdb2+hdb3+iami)]                       then[z18]
          if[spc*iami]                                              then[z2]
while[z4] with[low]
          if[hdb1+hdb2+hdb3+iami]                                   then[z5]
          if[ami]                                                   then[z1]
```

Figure 18. Description File DECODER.EQN (1 of 2)

```
while[z5] with[low]
          if[spc*(hdb2+hdb3)]                                    then[z6]
          if[(lowm*(hdb1+hdb2+hdb3))+(spc*iami)]                 then[z10]
          if[highm*iami]                                         then[z12]
          if[highm*hdb1]                                         then[z21]
          if[(highm*(hdb2+hdb3))+(lowm*iami)+(spc*hdb1)]         then[z18]
while[z6] with[low]
          if[hdb2+hdb3]                                          then[z7]
while[z7] with[low]
          if[highm*hdb2]                                         then[z21]
          if[lowm*(hdb2+hdb3)]                                   then[z10]
          if[(highm*hdb3)+(spc*hdb2)]                            then[z18]
          if[spc*hdb3]                                           then[z8]
while[z8] with[low]
          if[hdb3]                                               then[z9]
while[z9] with[low]
          if[spc*hdb3]                                           then[z18]
          if[highm*hdb3]                                         then[z21]
          if[lowm*hdb3]                                          then[z10]
while[z10] with[high]
          if[man*(spc+highm)]                                    then[z18]
          if[iami+ami+hdb1+hdb2+hdb3+rz+nrz+(man*lowm)]          then[z11]
while[z11] with[high]
          if[highm*(ami+hdb1+hdb2+hdb3)]                         then[z2]
          if[(highm*(rz+nrz+man))+(spc*iami)]                    then[z10]
          if[(spc*(ami+hdb1+hdb2+hdb3))+(highm*iami)+(lowm*man)] then[z12]
          if[spc*(rz+nrz)]                                       then[z1]
          if[(lowm*(iami+ami+hdb1+hdb2+hdb3+rz+nrz))+(spc*man)]  then[z18]
while[z12] with[low]
          if[man*(spc+lowm)]                                     then[z18]
          if[(highm*man)+ami]                                    then[z1]
          if[iami+hdb1+hdb2+hdb3]                                then[z13]
while[z13] with[low]
          if[lowm*hdb1]                                          then[z19]
          if[spc*iami]                                           then[z2]
          if[highm*(hdb1+hdb2+hdb3)]                             then[z2]
          if[(highm*iami)+(lowm*(hdb2+hdb3))+(hdb1*spc)]         then[z18]
          if[lowm*iami]                                          then[z4]
          if[spc*(hdb2+hdb3)]                                    then[z14]
while[z14] with[low]
          if[hdb2+hdb3]                                          then[z15]
while[z15] with[low]
          if[spc*hdb3]                                           then[z16]
          if[highm*(hdb2+hdb3)]                                  then[z2]
          if[(lowm*hdb3)+(spc*hdb2)]                             then[z18]
          if[lowm*hdb2]                                          then[z19]
while[z16] with[low]
          if[hdb3]                                               then[z17]
while[z17] with[low]
          if[spc*hdb3]                                           then[z18]
          if[highm*hdb3]                                         then[z2]
          if[lowm*hdb3]                                          then[z19]
while[z18] with[err]
          if[]                                                   then[z18]
while[z19] with[low]
          if[hdb1+hdb2+hdb3]                                     then[z20]
while[z20] with[low]
          if[lowm*(hdb1+hdb2+hdb3)]                              then[z18]
          if[highm*(hdb1+hdb2+hdb3)]                             then[z2]
          if[spc*(hdb1+hdb2+hdb3)]                               then[z12]
while[z21] with[low]
          if[hdb1+hdb2+hdb3]                                     then[z22]
while[z22] with[low]
          if[lowm*(hdb1+hdb2+hdb3)]                              then[z10]
          if[highm*(hdb1+hdb2+hdb3)]                             then[z18]
          if[spc*(hdb1+hdb2+hdb3)]                               then[z4]
```

Figure 18. Description File DECODER.EQN (2 of 2)

Timing simulation

Timing Diagram of Synchronizing Cycle

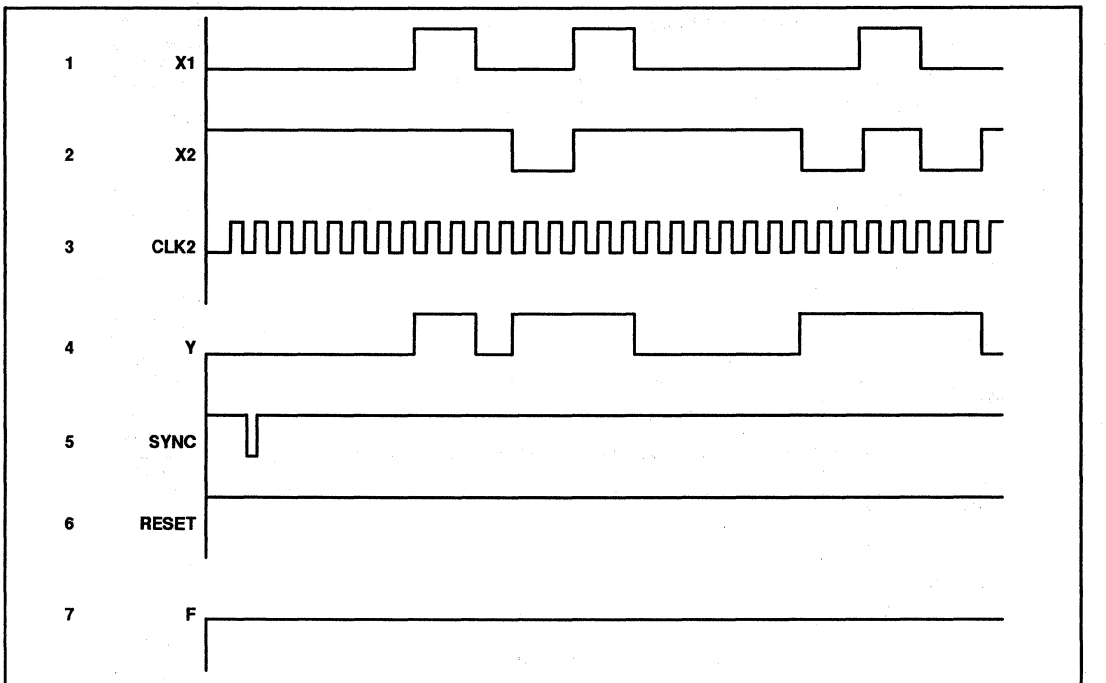


Figure 19. Timing Diagram of Synchronizing Cycle

Listings of Stimulus Files for Timing Simulation Using LESIM

```
************************************************
*     Output of Waveform Version 1.90          *
* Date: 06/08/93                Time: 19:25:46 *
************************************************
*                                              *
* Input File Name   : AMI.SCL                  *
* Rule File Name    : Scl Rule                 *
* Output File Name  : AMI.SCL                  *
*                                              *
************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 0 (16500, 20500, 28500, 32500, 52500, 56500, 100500, 104500,
# 112500, 116500) X1
 S 1 (24500, 28500, 48500, 52500, 56500, 60500, 104500, 108500,
# 120500, 124500) X2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 S 0 (1000, 2000, ETC) CK2
 S 0 (200000) S2
 S 0 (200000) S1
 S 0 (200000) S0
 SU time = 200000
 F
```

Figure 20. Stimulus File AMI.SCL, Control Word On Inputs S[2..0] = 000 B

```
************************************************
*     Output of Waveform Version 1.90          *
* Date: 06/08/93                Time: 19:25:46 *
************************************************
*                                              *
* Input File Name   : HDB1.SCL                 *
* Rule File Name    : Scl Rule                 *
* Output File Name  : HDB1.SCL                 *
*                                              *
************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 0 (16500, 20500, 28500, 32500, 36500, 40500, 44500, 48500, 52500, 56500,
# 100500, 104500, 112500, 116500) X1
 S 1 (4500, 8500, 12500, 16500, 24500, 28500, 48500, 52500, 56500, 60500,
# 64500, 68500, 72500, 76500, 80500, 84500, 88500, 92500, 96500, 100500,
# 104500, 108500, 120500, 124500) X2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 S 0 (1000, 2000, ETC) CK2
 S 0 (200000) S2
 S 0 (200000) S1
 S 1 (200000) S0
 SU time = 200000
 F
```

Figure 21. Stimulus File HDB1.SCL, Control Word On Inputs S[2..0] = 001 B

```
*     Output of Waveform Version 1.90            *
* Date: 06/08/93                    Time: 19:25:46 *
*************************************************
*                                               *
* Input File Name   : HDB2.SCL                  *
* Rule File Name    : Scl Rule                  *
* Output File Name  : HDB2.SCL                  *
*                                               *
*************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 0 (16500, 20500, 28500, 32500, 40500, 44500, 52500, 56500,
# 100500, 104500, 112500, 116500) X1
 S 1 (8500, 12500, 24500, 28500, 48500, 52500, 56500, 60500, 68500,
# 72500, 80500, 84500, 92500, 96500, 104500, 108500, 120500, 124500) X2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 S 0 (1000, 2000, ETC) CK2
 S 0 (200000) S2
 S 1 (200000) S1
 S 0 (200000) S0
 SU time = 200000
 F
```

Figure 22. Stimulus File HDB2.SCL, Control Word On Inputs S[2..0] = 010 B

```
*************************************************
*     Output of Waveform Version 1.90           *
* Date: 06/08/93                   Time: 19:25:46 *
*************************************************
*                                               *
* Input File Name   : HDB3.SCL                  *
* Rule File Name    : Scl Rule                  *
* Output File Name  : HDB3.SCL                  *
*                                               *
*************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 0 (16500, 20500, 28500, 32500, 44500, 48500, 52500, 56500,
# 100500, 104500, 112500, 116500) X1
 S 1 (12500, 16500, 24500, 28500, 48500, 52500, 56500, 60500, 72500,
# 76500, 88500, 92500, 104500, 108500, 120500, 124500) X2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 S 0 (1000, 2000, ETC) CK2
 S 0 (200000) S2
 S 1 (200000) S1
 S 1 (200000) S0
 SU time = 200000
 F
```

Figure 23. Stimulus File HDB3.SCL, Control Word On Inputs S[2..0] = 011 B

```
*     Output of Waveform Version 1.90            *
* Date: 06/08/93                  Time: 19:25:46 *
**************************************************
*                                                *
* Input File Name   : INV_AMI.SCL                *
* Rule File Name    : Scl Rule                   *
* Output File Name  : INV_AMI.SCL                *
*                                                *
**************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 0 (500, 4500, 8500, 12500, 20500, 24500, 36500, 40500, 44500,
# 48500, 64500, 68500, 72500, 76500, 80500, 84500, 88500, 92500,
# 96500, 100500, 116500, 120500) X1
 S 1 (4500, 8500, 12500, 16500, 32500, 36500, 40500, 44500,
# 60500, 64500, 68500, 72500, 76500, 80500, 84500, 88500, 92500,
# 96500, 108500, 112500, 124500, 128500) X2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 S 0 (1000, 2000, ETC) CK2
 S 1 (200000) S2
 S 0 (200000) S1
 S 0 (200000) S0
 SU time = 200000
 F
```

Figure 24. Stimulus File INV_AMI.SCL, Control Word On Inputs S[2..0] = 100 B

```
**************************************************
*     Output of Waveform Version 1.90            *
* Date: 07/19/93                  Time: 16:01:57 *
**************************************************
*                                                *
* Input File Name   : RZ.SCL                     *
* Rule File Name    : Scl Rule                   *
* Output File Name  : RZ.SCL                     *
*                                                *
**************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 1 (200000) S2
 S 0 (200000) S1
 S 1 (200000) S0
 S 0 (16500, 18500, 24500, 26500, 28500, 30500, 48500, 50500, 52500,
#   54500, 56500, 58500, 100500, 102500, 104500, 106500, 112500, 114500,
#   120500, 122500) X1
 S 1 (200010) X2
 S 0 (1000, 2000, ETC) CK2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 25. Stimulus File RZ.SCL, Control Word On Inputs S[2..0] = 101 B

```
*     Output of Waveform Version 1.90              *
* Date: 07/19/93                    Time: 16:01:57 *
*************************************************
* Input File Name   : NRZ.SCL                     *
* Rule File Name    : Scl Rule                    *
* Output File Name  : NRZ.SCL                     *
*************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 1 (200000) S2
 S 1 (200000) S1
 S 0 (200000) S0
 S 0 (16500, 20500, 26500, 32500, 48500, 60500, 100500, 108500,
# 112500, 116500, 120500, 124500) X1
 S 1 (200010) X2
 S 0 (1000, 2000, ETC) CK2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 26. Stimulus File NRZ.SCL, Control Word On Inputs S[2..0] = 110 B

```
*************************************************
*     Output of Waveform Version 1.90             *
* Date: 07/19/93                    Time: 16:01:57 *
*************************************************
* Input File Name   : MANCHEST.SCL                *
* Rule File Name    : Scl Rule                    *
* Output File Name  : MANCHEST.SCL                *
*************************************************
 P S[2..0], X1, X2, CK2, Y, ZZ[4..0], SYNC, RESET, F
 PCO
 S 1 (200000) S2
 S 1 (200000) S1
 S 1 (200000) S0
 S 0 (2500, 4500, 6500, 8500, 10500, 12500, 14500, 18500, 22500,
# 26500, 28500, 30500, 34500, 36500, 38500, 40500, 42500, 44500,
# 46500, 50500, 52500, 54500, 56500, 58500, 62500, 64500, 66500,
# 68500, 70500, 72500, 74500, 76500, 78500, 80500, 82500, 84500,
# 86500, 88500, 90500, 92500, 94500, 96500, 98500, 102500, 104500,
# 106500, 110500, 114500, 118500, 122500, 126500, 128500) X1
 S 0 (2500, 4500, 6500, 8500, 10500, 12500, 14500, 18500, 22500,
# 26500, 28500, 30500, 34500, 36500, 38500, 40500, 42500, 44500,
# 46500, 50500, 52500, 54500, 56500, 58500, 62500, 64500, 66500,
# 68500, 70500, 72500, 74500, 76500, 78500, 80500, 82500, 84500,
# 86500, 88500, 90500, 92500, 94500, 96500, 98500, 102500, 104500,
# 106500, 110500, 114500, 118500, 122500, 126500, 128500) X2
 S 0 (1000, 2000, ETC) CK2
 S 0 (300, 400) SYNC
 S 1 (100, 200) RESET
 SU time = 200000
 F
```

Figure 27. Stimulus File MANCHEST.SCL, Control Word On Inputs S[2..0] = 111 B

Wiring of the Decoder Unit

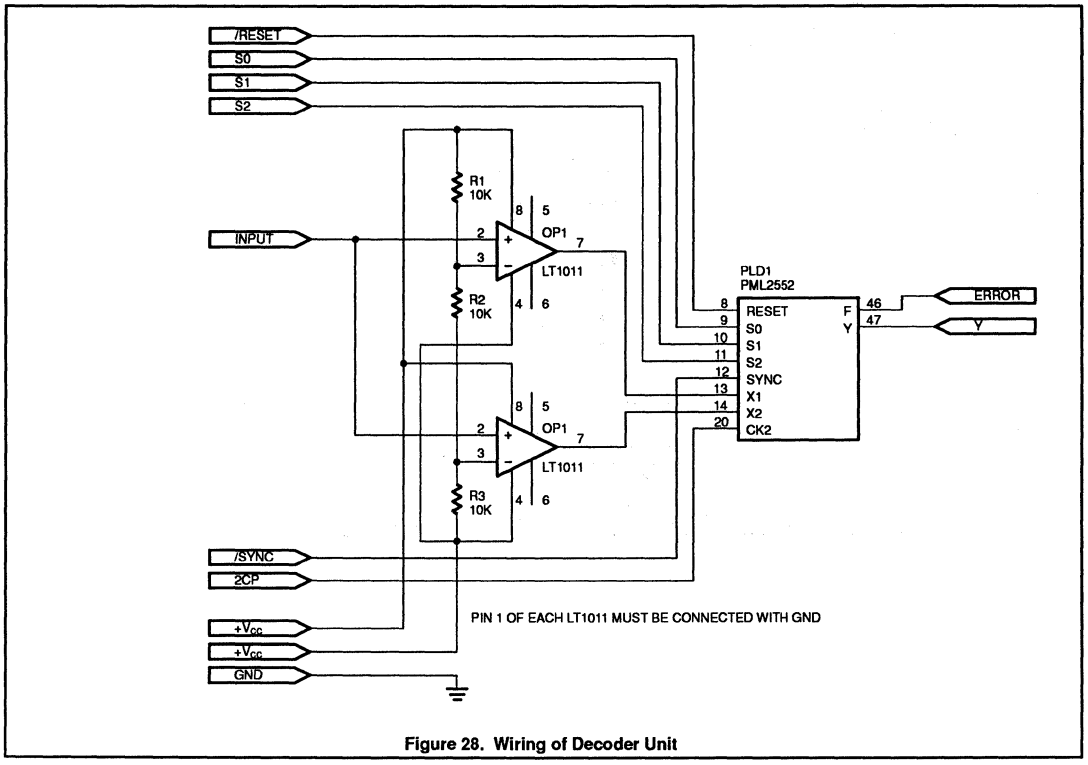


Figure 28. Wiring of Decoder Unit

WIRING OF THE ENCODER AND DECODER UNITS FOR TESTING

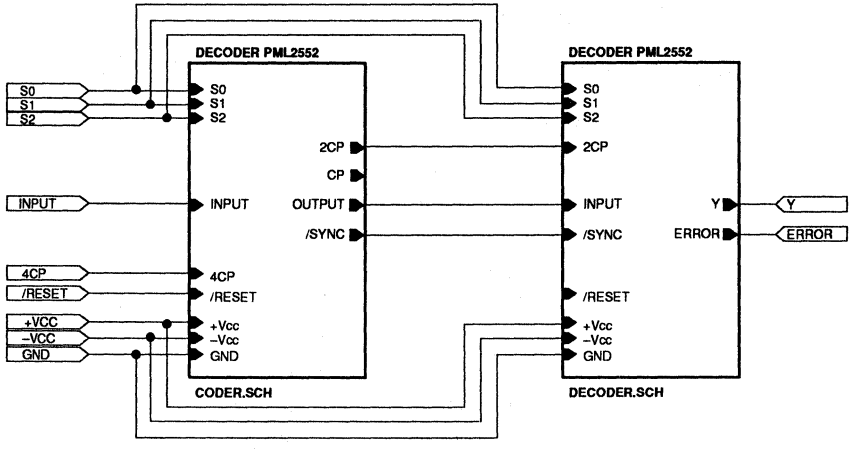
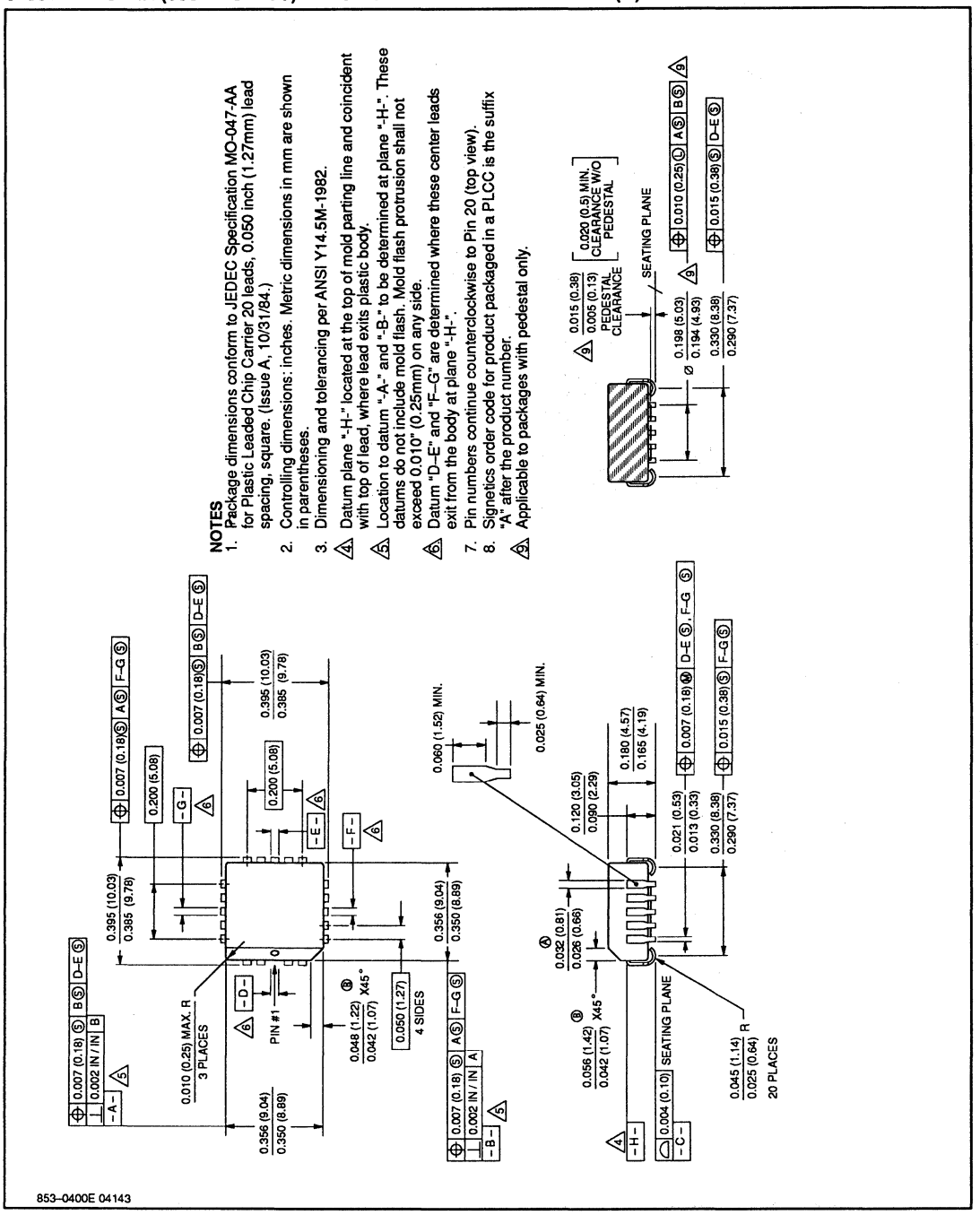


Figure 29. Wiring of the Encoder and Decoder Units for Testing

Section 12
Package Outlines

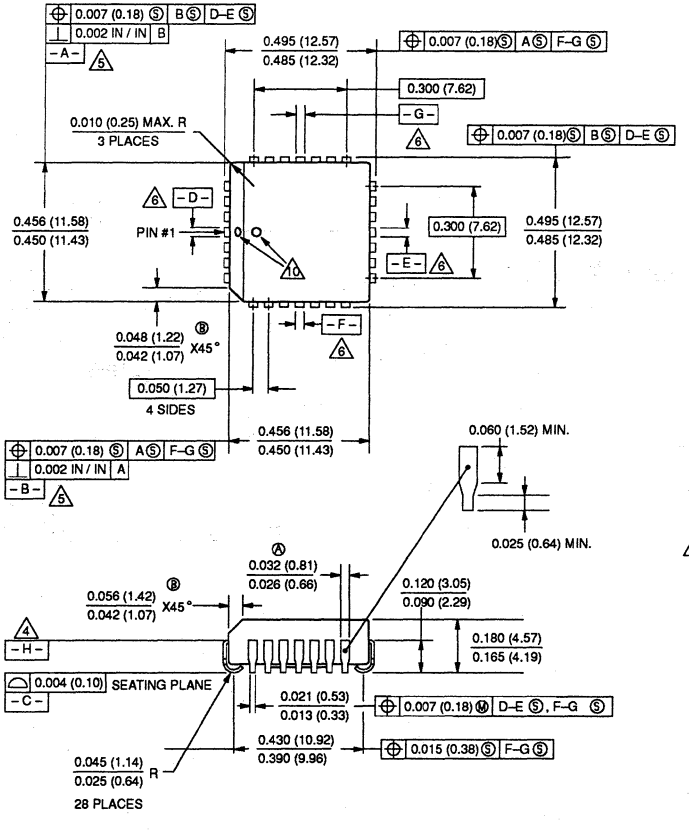
Package outlines

0400E 20-PIN (350 mils wide) PLASTIC LEADED CHIP CARRIER (A) PACKAGE



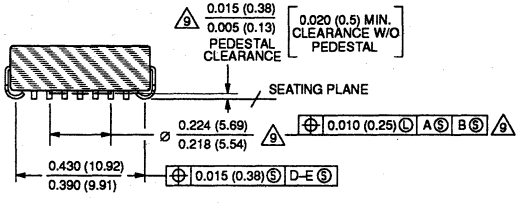
Package outlines

0401F 28-PIN (300 mils wide) PLASTIC LEADED CHIP CARRIER (A) PACKAGE



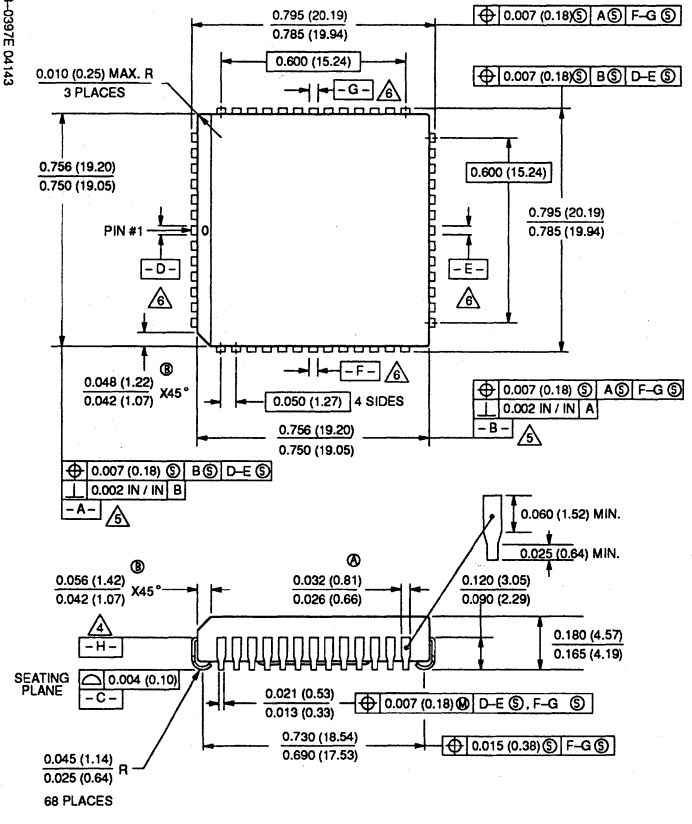
NOTES

1. Package dimensions conform to JEDEC Specification MO-047-AB for Plastic Leaded Chip Carrier 28 leads, 0.050 inch (1.27mm) lead spacing, square. (Issue A, 10/31/84.)
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "-H-" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "-A-" and "-B-" to be determined at plane "-H-". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
6. Datum "D–E" and "F–G" are determined where these center leads exit from the body at plane "-H-".
7. Pin numbers continue counterclockwise to Pin 28 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.
10. Location of Pin #1 mark is optional. Mark on chamfered side is preferred.



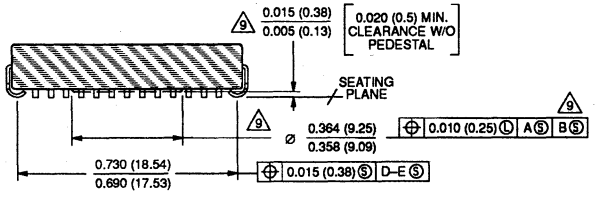
Package outlines

0397E 52-PIN PLASTIC LEADED CHIP CARRIER (A) PACKAGE



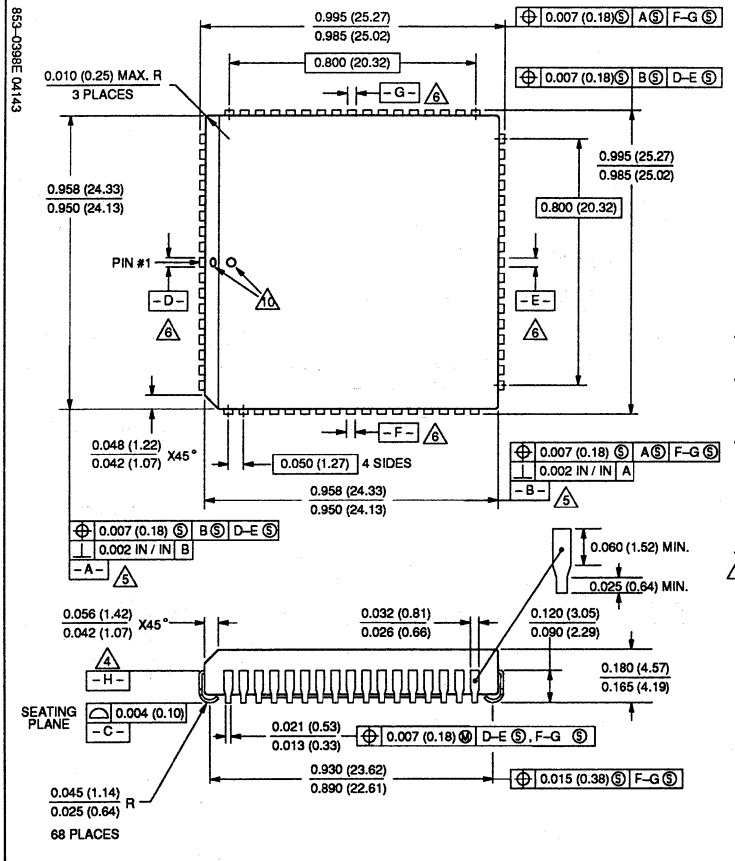
NOTES

1. Package dimensions conform to JEDEC Specification MO-047-AD for Plastic Leaded Chip Carrier 52 leads, 0.050 inch (1.27mm) lead spacing, square. (Issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "-H-" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "-A-" and "-B-" to be determined at plane "-H-". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
6. Datum "D–E" and "F–G" are determined where these center leads exit from the body at plane "-H-".
7. Pin numbers continue counterclockwise to Pin 52 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.



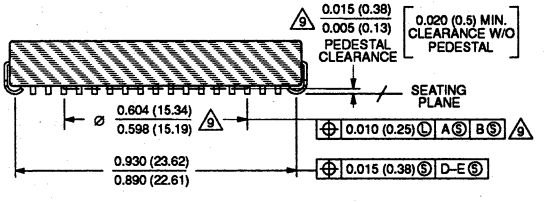
Package outlines

0398E 68-PIN PLASTIC LEADED CHIP CARRIER (A) PACKAGE



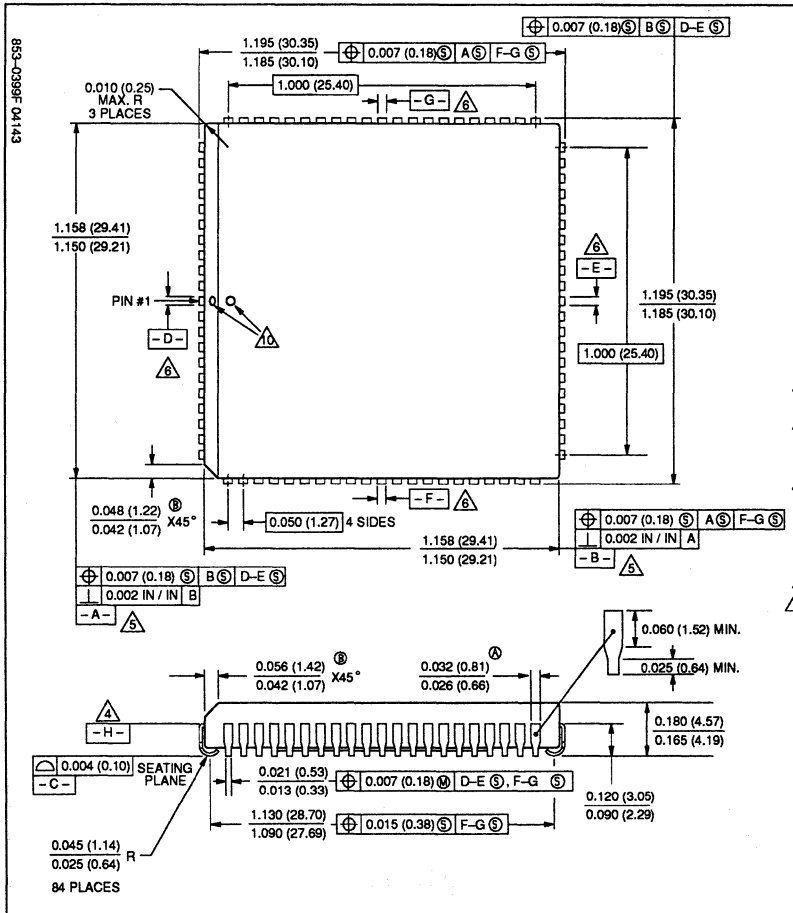
NOTES

1. Package dimensions conform to JEDEC Specification MO-047-AE for Plastic Leaded Chip Carrier 68 leads, 0.050 inch lead spacing, square. (Issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "-H-" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "-A-" and "-B-" to be determined at plane "-H-". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
6. Datum "D–E" and "F–G" are determined where these center leads exit from the body at plane "-H-".
7. Pin numbers continue counterclockwise to Pin 68 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.
10. Location of Pin #1 mark is optional. Mark on chamfered side is preferred.



Package outlines

0399F 84-PIN PLASTIC LEADED CHIP CARRIER (A) PACKAGE

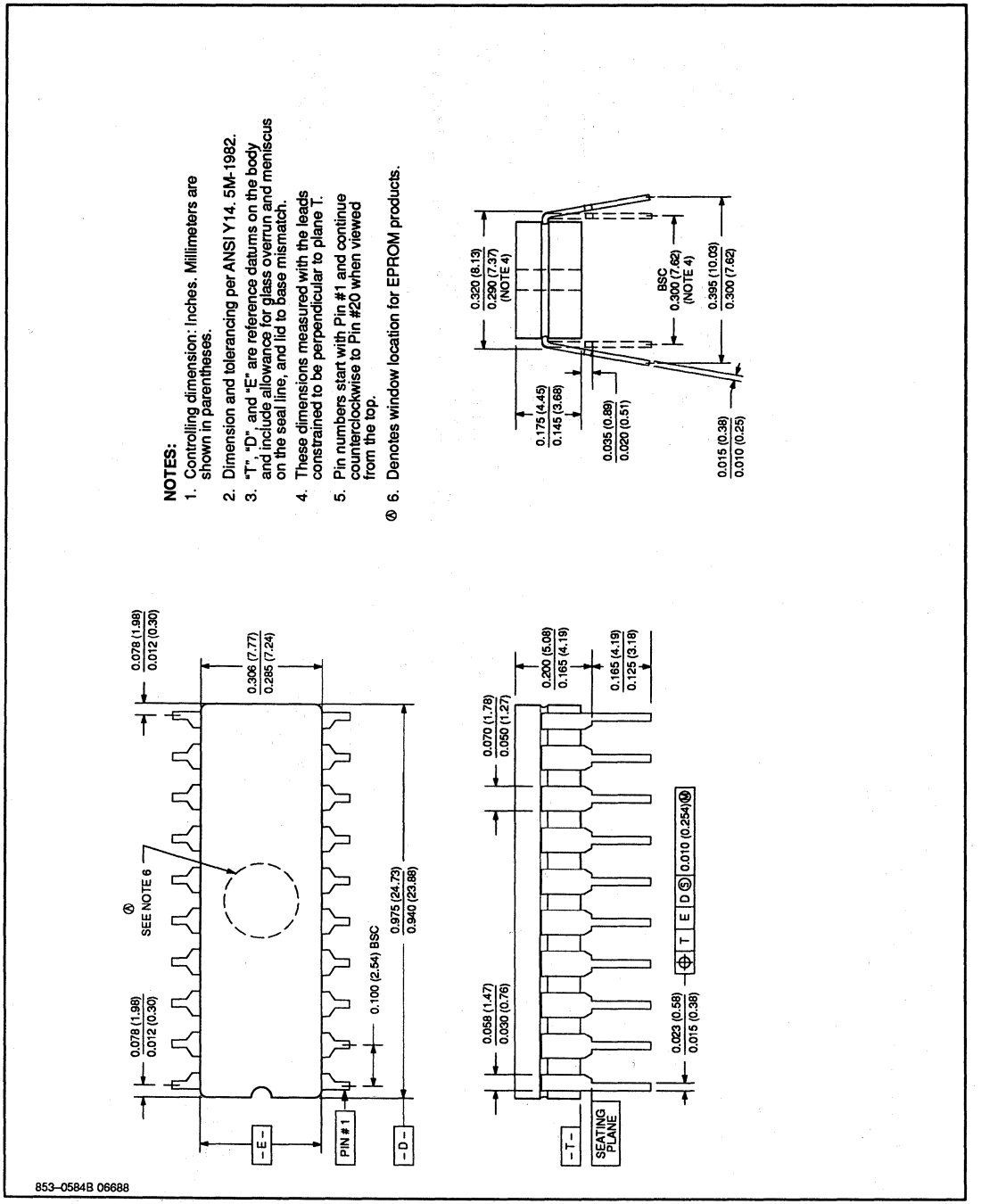


NOTES

1. Package dimensions conform to JEDEC Specification MO-047-AF for Plastic Leaded Chip Carrier 84 leads, 0.050 inch lead spacing, square. (Issue A, 10/31/84).
2. Controlling dimensions: inches. Metric dimensions in mm are shown in parentheses.
3. Dimensioning and tolerancing per ANSI Y14.5M-1982.
4. Datum plane "-H-" located at the top of mold parting line and coincident with top of lead, where lead exits plastic body.
5. Location to datum "-A-" and "-B-" to be determined at plane "-H-". These datums do not include mold flash. Mold flash protrusion shall not exceed 0.010" (0.25mm) on any side.
6. Datum "D–E" and "F–G" are determined where these center leads exit from the body at plane "-H-".
7. Pin numbers continue counterclockwise to Pin 84 (top view).
8. Signetics order code for product packaged in a PLCC is the suffix "A" after the product number.
9. Applicable to packages with pedestal only.
10. Location of Pin #1 mark is optional. Mark on chamfered side is preferred.

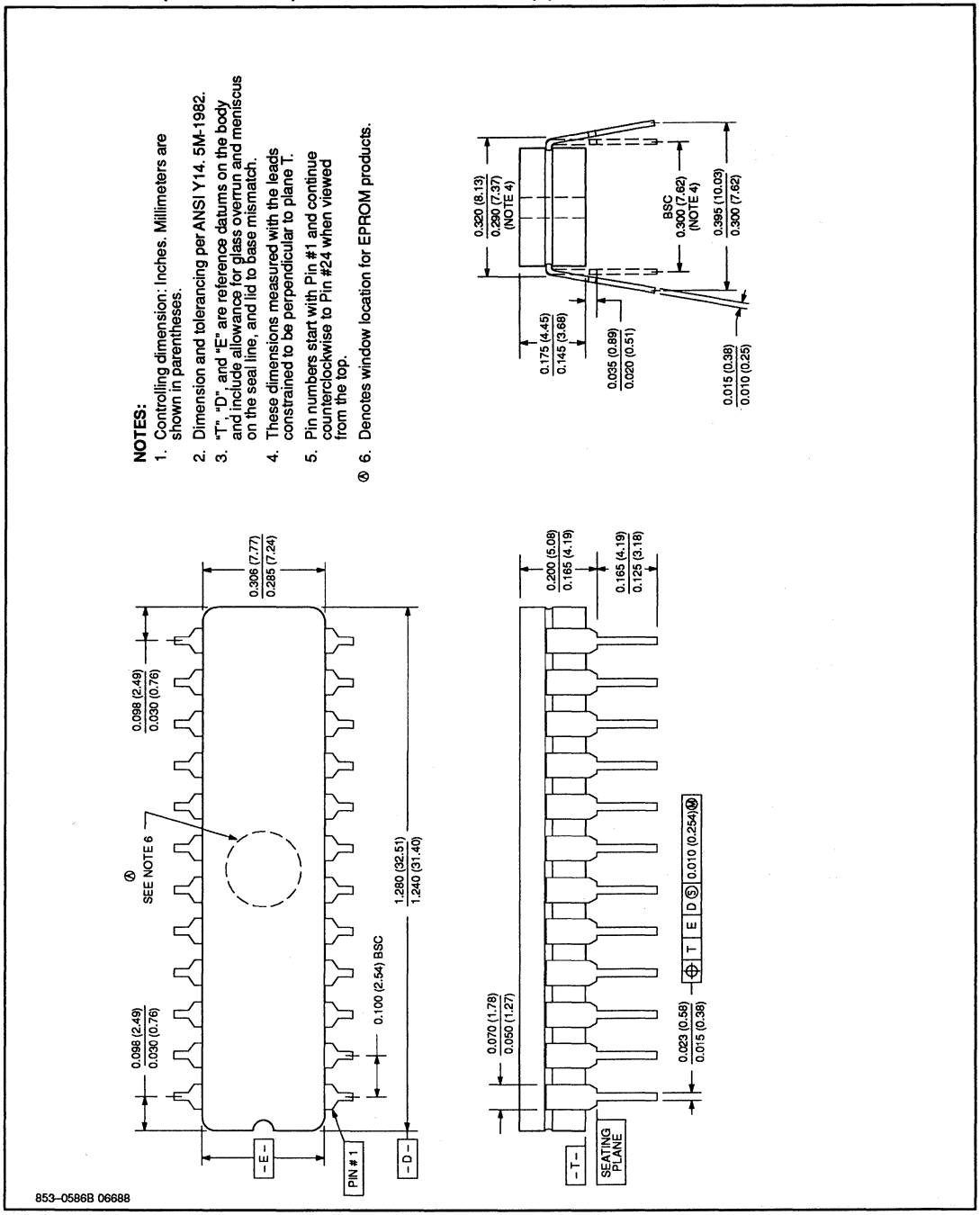
Package outlines

0584B 20-PIN (300 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)



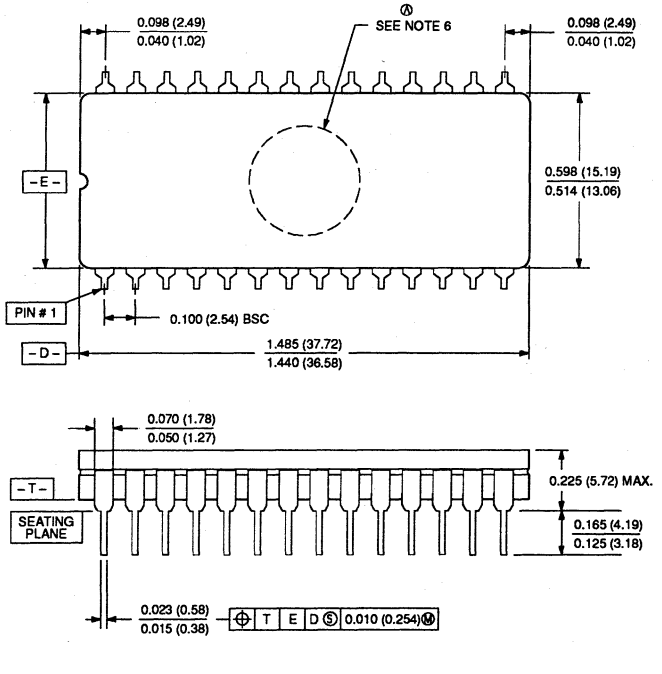
Package outlines

0586B 24-PIN (300 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)



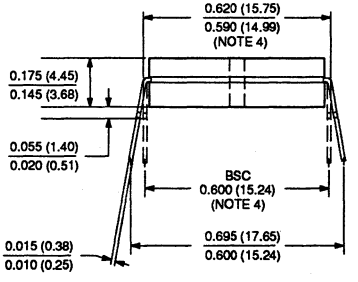
Package outlines

0589B 28-PIN (600 mils wide) CERAMIC DUAL IN-LINE (F) PACKAGE (WITH WINDOW (FA) PACKAGE)



NOTES:

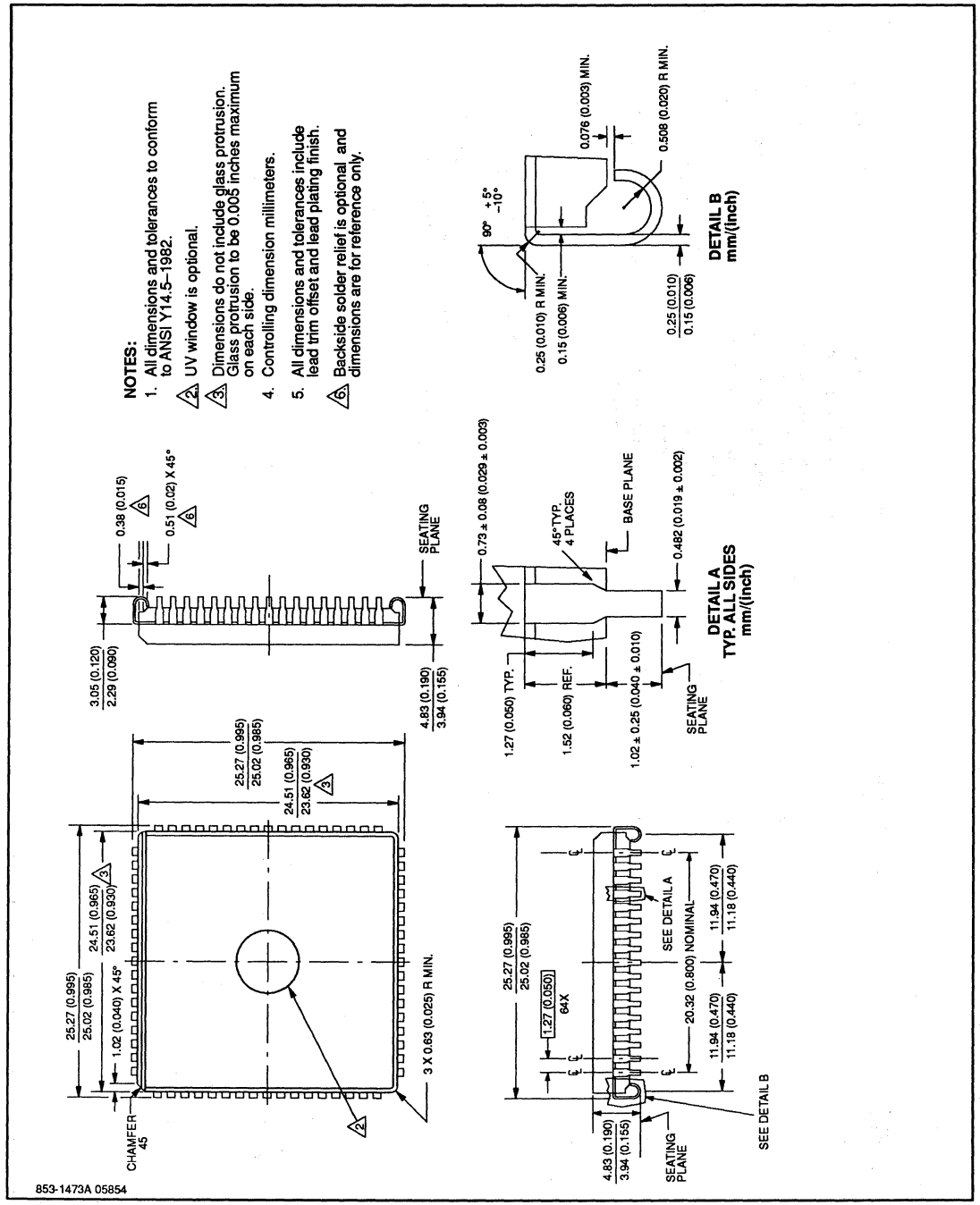
1. Controlling dimension: Inches. Millimeters are shown in parentheses.
2. Dimension and tolerancing per ANSI Y14. 5M-1982.
3. "T", "D", and "E" are reference datums on the body and include allowance for glass overrun and meniscus on the seal line, and lid to base mismatch.
4. These dimensions measured with the leads constrained to be perpendicular to plane T.
5. Pin numbers start with Pin #1 and continue counterclockwise to Pin #28 when viewed from the top.
6. Ⓐ Denotes window location for EPROM products.



853-0589B 08898

Package outlines

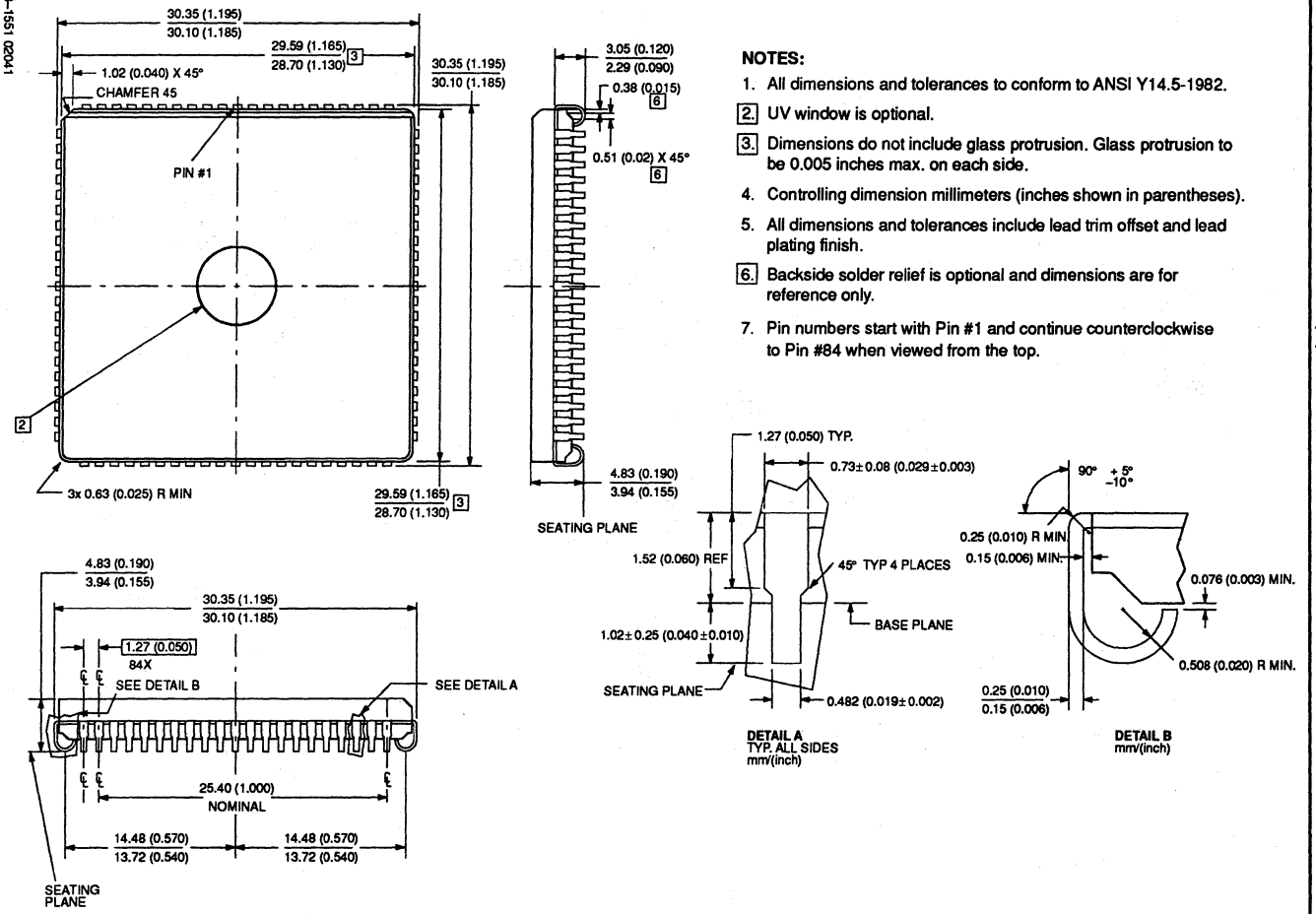
1473A 68-PIN CERQUAD J-BEND (K) PACKAGE



Package outlines

1551 84-PIN CERQUAD J-BEND (K) PACKAGE

853-1551 02041

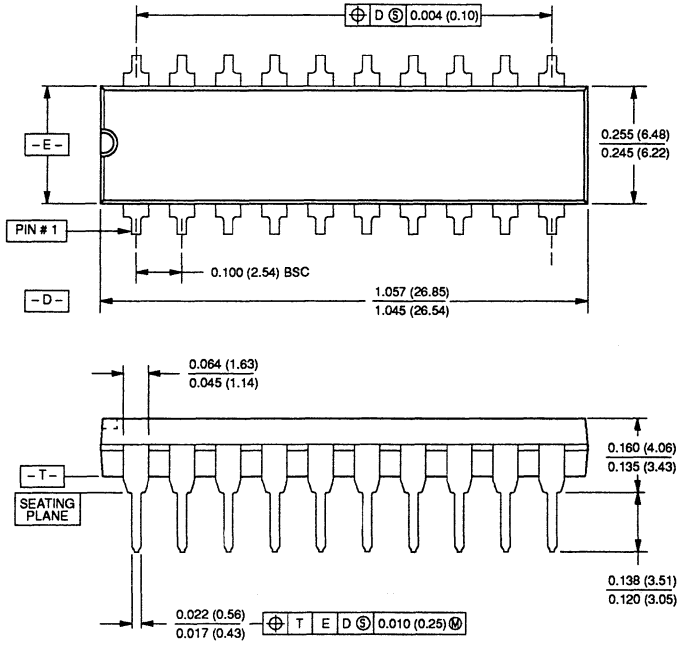


NOTES:

1. All dimensions and tolerances to conform to ANSI Y14.5-1982.
2. UV window is optional.
3. Dimensions do not include glass protrusion. Glass protrusion to be 0.005 inches max. on each side.
4. Controlling dimension millimeters (inches shown in parentheses).
5. All dimensions and tolerances include lead trim offset and lead plating finish.
6. Backside solder relief is optional and dimensions are for reference only.
7. Pin numbers start with Pin #1 and continue counterclockwise to Pin #84 when viewed from the top.

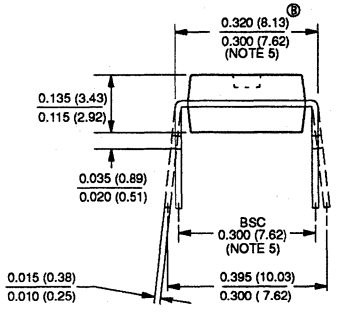
Package outlines

0408B 20-PIN (300 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE



NOTES

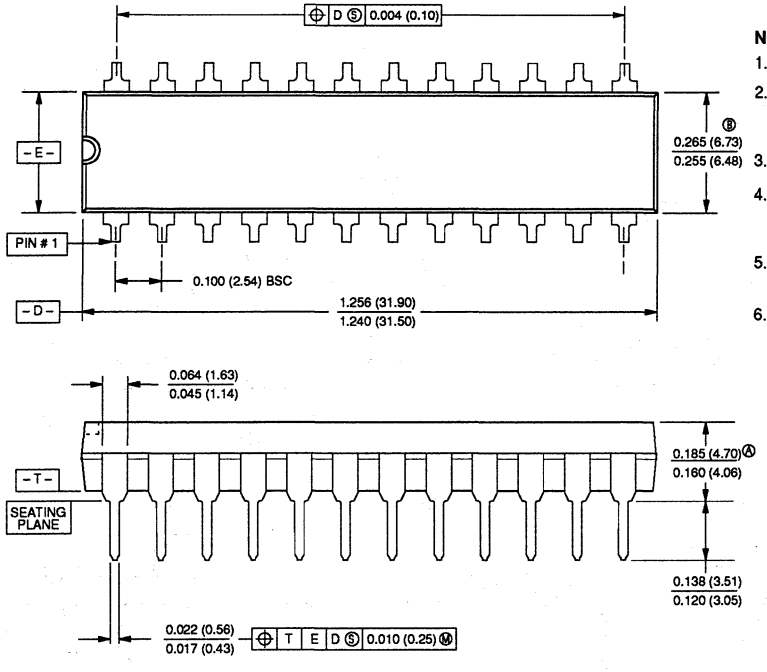
1. Controlling dimension: Inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-001-AE for standard Dual In-Line (DIP) package 0.300 inch row spacing (plastic) 20 leads (Issue B, 7/85).
3. Dimension and tolerancing per ANSI Y14, 5M – 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #20 when viewed from the top.



853-0408B 02880

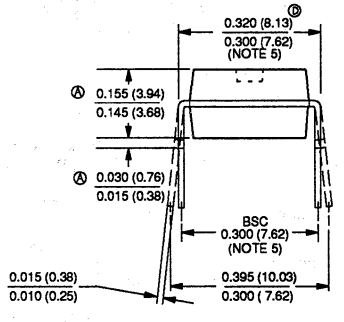
Package outlines

0410D 24-PIN (300 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE



NOTES:

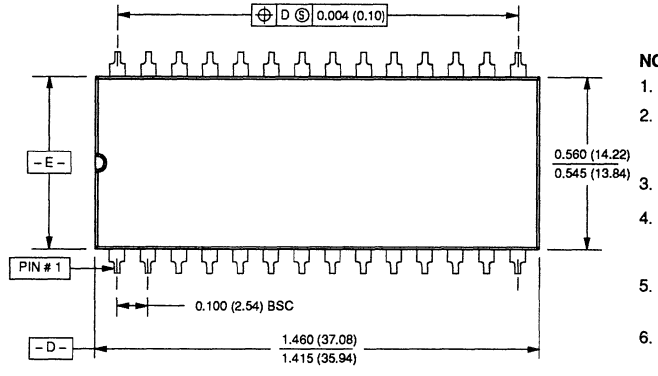
1. Controlling dimension: Inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-001-AF for standard Dual In-Line (DIP) package 0.300 inch row spacing (plastic) 24 leads (Issue B, 7/85).
3. Dimension and tolerancing per ANSI Y14, 5M – 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #24 when viewed from the top.



853-0410D 02880

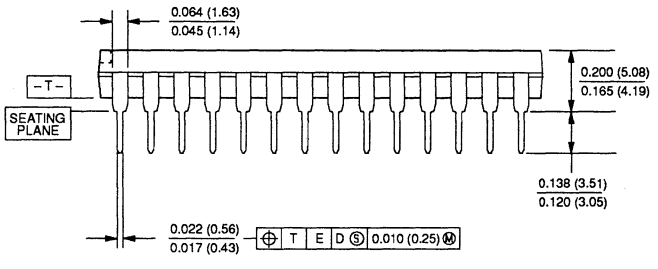
Package outlines

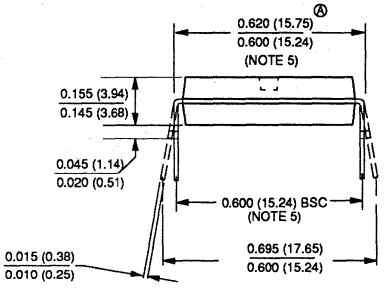
0413B 28-PIN (600 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE



NOTES:

1. Controlling dimension: Inches. Metric are shown in parentheses.
2. Package dimensions conform to JEDEC Specification MS-011-AB for standard Dual In-Line (DIP) package 0.600 inch row spacing (plastic) 28 leads (Issue B, 7/84).
3. Dimension and tolerancing per ANSI Y14, 5M – 1982.
4. "T", "D", and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010 inch (0.25mm) on any side.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Pin numbers start with Pin #1 and continue counterclockwise to Pin #28 when viewed from the top.

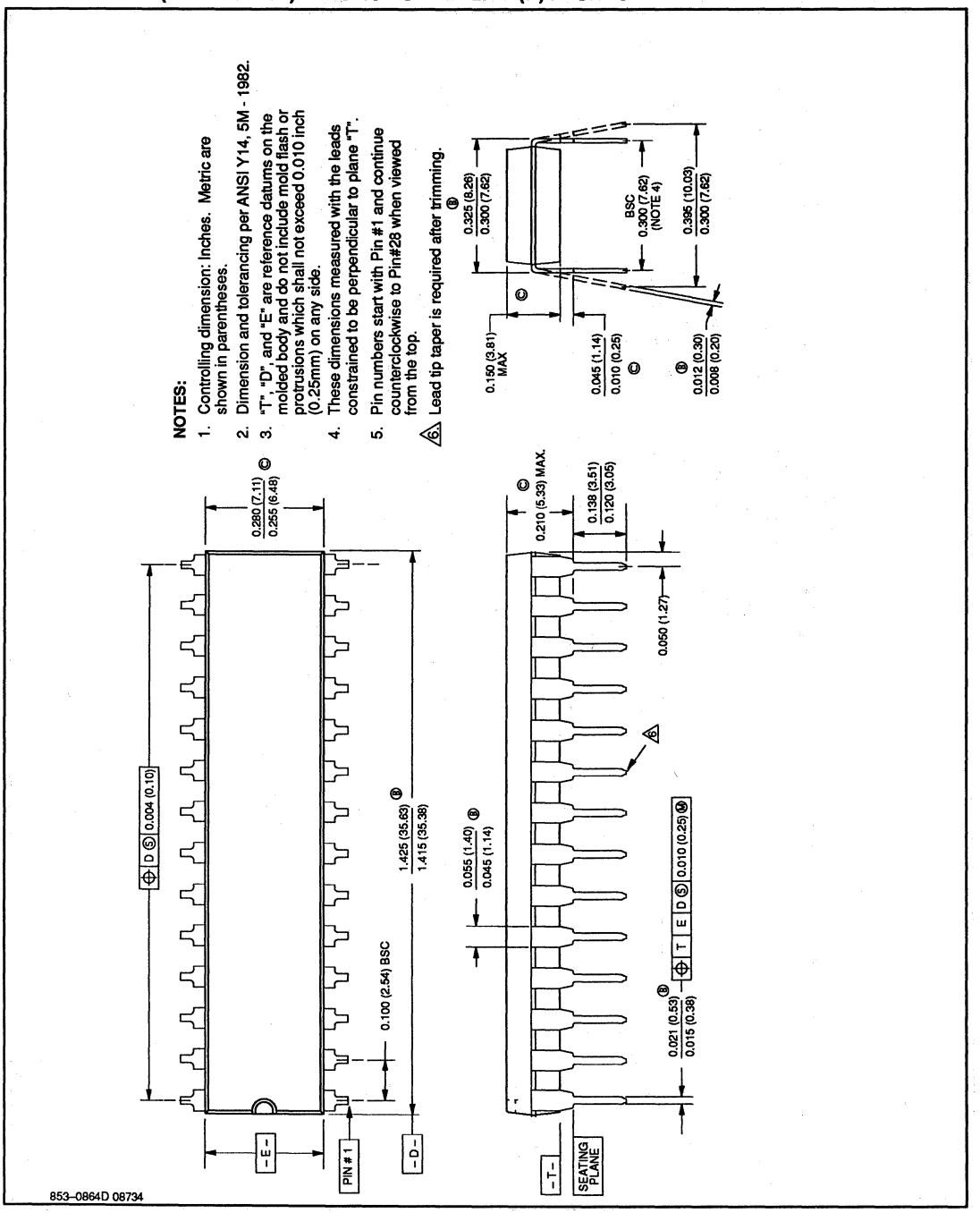




853-0413B 01988

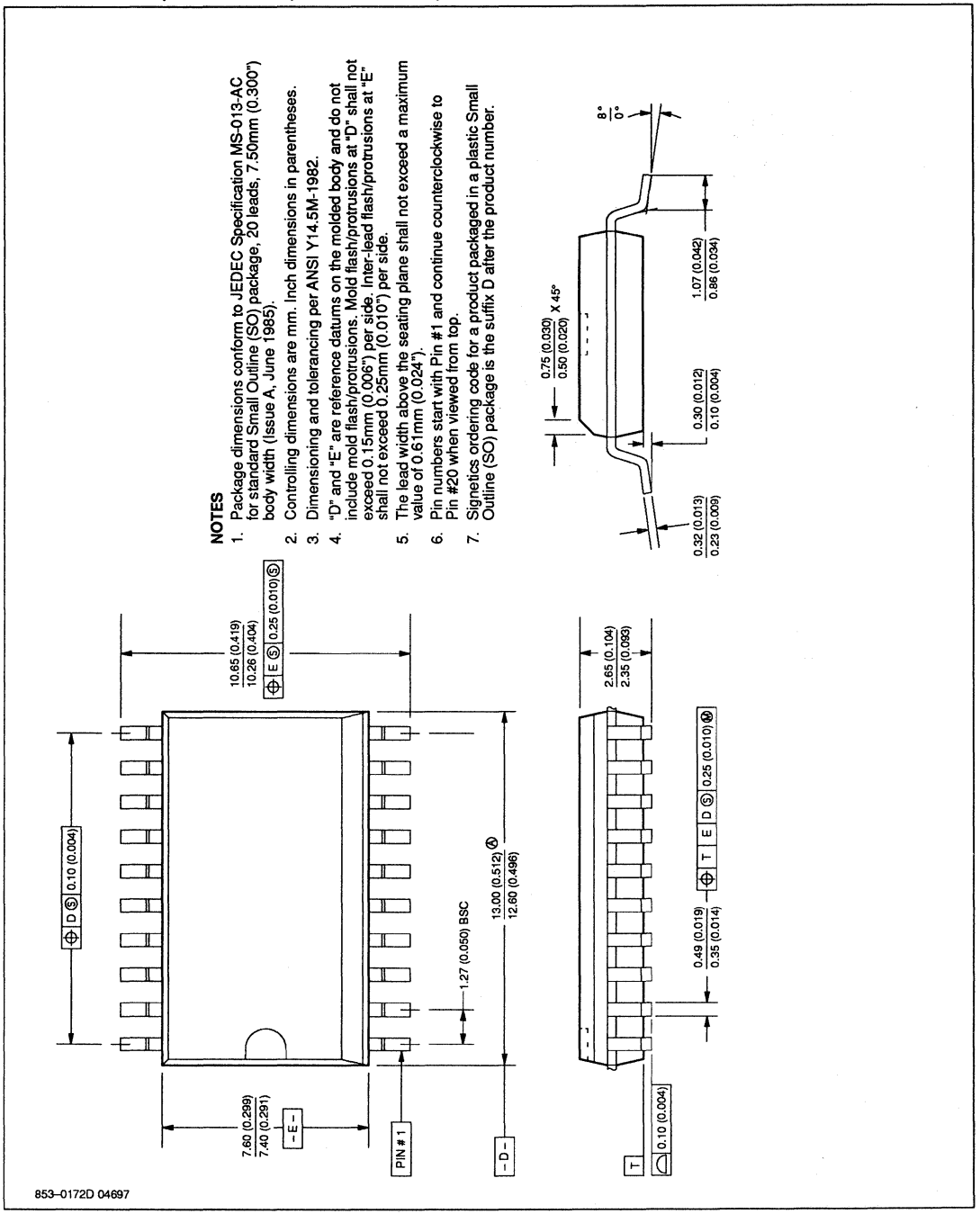
Package outlines

0864D 28-PIN (300 mils wide) PLASTIC DUAL IN-LINE (N) PACKAGE



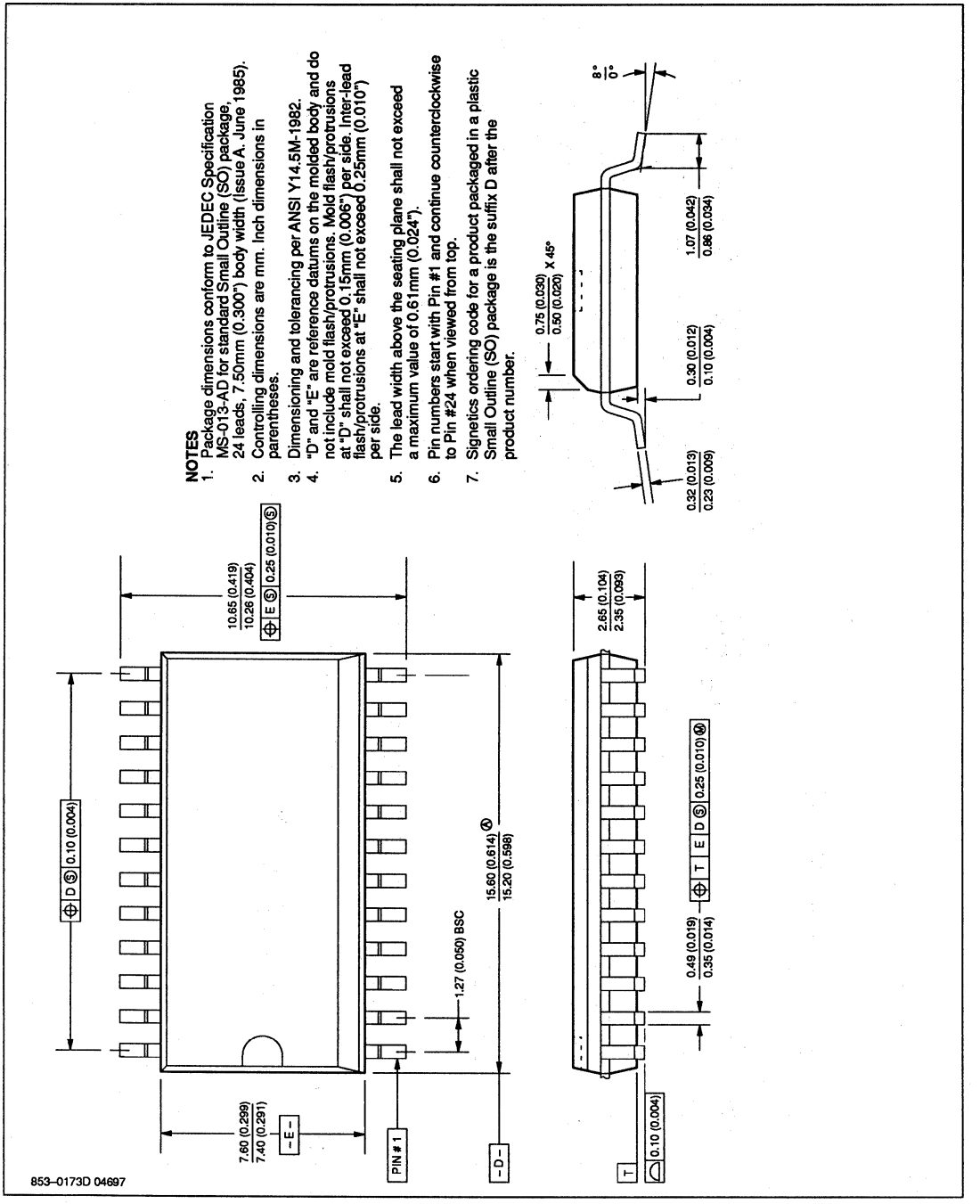
Package outlines

0172D 20-PIN (300 mils wide) PLASTIC SOL (SMALL OUTLINE LARGE) DUAL IN-LINE (D) PACKAGE



Package outlines

0173D 24-PIN (300 mils wide) PLASTIC SOL (SMALL OUTLINE LARGE) DUAL IN-LINE (D) PACKAGE



Section 13
North American Regional Sales Organization Sales Offices

North American Sales Offices, Representatives and Distributors

PHILIPS SEMICONDUCTORS
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, CA 94088-3409

ALABAMA
Huntsville
Philips Semiconductors
Phone: (205) 464-0111

Elcom, Inc.
Phone: (205) 830-4001

ARIZONA
Scottsdale
Thom Luke Sales, Inc.
Phone: (602) 451-5400

CALIFORNIA
Calabasas
Philips Semiconductors
Phone: (818) 880-6304

Irvine
Philips Semiconductors
Phone: (714) 833-8980
(714) 752-2780

Orangevale
Webster Associates
Phone: (916) 989-0843

San Diego
Philips Semiconductors
Phone: (619) 560-0242

San Jose
B.A.E. Sales, Inc.
Phone: (408) 452-8133

Sunnyvale
Philips Semiconductors
Phone: (408) 991-3737

COLORADO
Englewood
Philips Semiconductors
Phone: (303) 792-9011

Thom Luke Sales, Inc.
Phone: (303) 649-9717

CONNECTICUT
Wallingford
JEBCO
Phone: (203) 265-1318

FLORIDA
Oviedo
Conley and Assoc., Inc.
Phone: (407) 365-3283

GEORGIA
Atlanta
Philips Semiconductors
Phone: (404) 594-1392

Norcross
Elcom, Inc.
Phone: (404) 447-8200

ILLINOIS
Hoffman Estates
Micro-Tex, Inc.
Phone: (708) 765-3000

Itasca
Philips Semiconductors
Phone: (708) 250-0050

INDIANA
Indianapolis
Mohrfield Marketing, Inc.
Phone: (317) 546-6969

Kokomo
Philips Semiconductors
Phone: (317) 459-5355

MARYLAND
Columbia
Third Wave Solutions, Inc.
Phone: (410) 290-5990

MASSACHUSETTS
Chelmsford
JEBCO
Phone: (508) 256-5800

Westford
Philips Semiconductors
Phone: (508) 692-6211

MICHIGAN
Monroe
S-J Associates
Phone: (313) 242-0450

Novi
Philips Semiconductors
Phone: (313) 347-1700

MINNESOTA
Bloomington
High Technology Sales
Phone: (612) 844-9933

MISSOURI
Bridgeton
Centech, Inc.
Phone: (314) 291-4230

Raytown
Centech, Inc.
Phone: (816) 358-8100

NEW JERSEY
Toms River
Philips Semiconductors
Phone: (908) 505-1200

NEW YORK
Ithaca
Bob Dean, Inc.
Phone: (607) 257-1111

Rockville Centre
S-J Associates
Phone: (516) 536-4242

Wappingers Falls
Philips Semiconductors
Phone: (914) 297-4074

Bob Dean, Inc.
Phone: (914) 297-6406

NORTH CAROLINA
Greensboro
ADI, Inc.
Phone: (919) 273-8887

Matthews
ADI, Inc.
Phone: (704) 847-4323

Smithfield
ADI, Inc.
Phone: (919) 934-8136

OHIO
Aurora
S-J Associates, Inc.
Phone: (216) 562-2050

Columbus
S-J Associates, Inc.
Phone: (614) 885-6700

Kettering
S-J Associates, Inc.
Phone: (513) 298-7322

Parma
S-J Associates, Inc.
Phone: (216) 888-7004

Toledo
S-J Associates, Inc.
Phone: (313) 242-0450

West Carrollton
S-J Associates, Inc.
Phone: (513) 438-1700

OREGON
Beaverton
Philips Semiconductors
Phone: (503) 627-0110

Western Technical Sales
Phone: (503) 644-8860

PENNSYLVANIA
Erie
S-J Associates, Inc.
Phone: (216) 888-7004

Hatboro
Delta Technical Sales, Inc.
Phone: (215) 957-0600

Pittsburgh
S-J Associates, Inc.
Phone: (216) 888-7004

Plymouth Meeting
Philips Semiconductors
Phone: (215) 825-4404

TENNESSEE
Greeneville
Philips Semiconductors
Phone: (615) 639-0251

TEXAS
Austin
Philips Semiconductors
Phone: (512) 339-9945

Austin
Synergistic Sales, Inc.
Phone: (512) 346-2122

Houston
Synergistic Sales, Inc.
Phone: (713) 937-1990

Richardson
Philips Semiconductors
Phone: (214) 644-1610

Richardson
Synergistic Sales, Inc.
Phone: (214) 644-3500

UTAH
Salt Lake City
Electrodyne
Phone: (801) 264-8050

WASHINGTON
Bellevue
Western Technical Sales
Phone: (206) 641-3900

Spokane
Western Technical Sales
Phone: (509) 922-7600

WISCONSIN
Waukesha
Micro-Tex, Inc.
Phone: (414) 542-5352

CANADA
PHILIPS SEMICONDUCTORS CANADA, LTD.

Calgary, Alberta
Tech-Trek, Ltd.
Phone: (403) 241-1719

Kanata, Ontario
Philips Semiconductors
Phone: (613) 599-8720

Tech-Trek, Ltd.
Phone: (613) 599-8787

Mississauga, Ontario
Tech-Trek, Ltd.
Phone: (416) 238-0366

Richmond, B.C.
Tech-Trek, Ltd.
Phone: (604) 276-8735

Ville St. Laurent, Quebec
Tech-Trek, Ltd.
Phone: (514) 337-7540

MEXICO
Anzures Section
Philips Components
Phone: 52-5-533-3858

El Paso, TX
Philips Components
Phone: (915) 775-4020

PUERTO RICO
Santurce
Mectron Group
Phone: (809) 723-6165

DISTRIBUTORS

Contact one of our local distributors:
Almac/Arrow Electronics
Anthem Electronics
Arrow/Schweber Electronics
Future Electronics
Gerber Electronics
Hamilton Hallmark
Marshall Industries
Wyle Laboratories, EMG
Zeus Electronics

10/26/93

Data handbook system

Appendix A

DATA HANDBOOK SYSTEM

Philips Semiconductors data handbooks contain all pertinent data available at the time of publication and each is revised and reissued regularly.

Loose data sheets are sent to subscribers to keep them up-to-date on additions or alterations made during the lifetime of a data handbook.

Catalogs are available for selected product ranges (some catalogs are also on floppy discs).

Our data handbook titles are listed here.

Integrated Circuits

Book	*Title*
IC01	Semiconductors for Radio and Audio Systems
IC02	Semiconductors for Television and Video Systems
IC03	Semiconductors for Telecom Systems
IC04	CMOS HE4000B Logic Family
IC05	Advanced Low-power Schottky (ALS) Logic Series
IC06	High-speed CMOS Logic Family
IC08	100K ECL Logic Families
IC10	Memories
IC11	General-purpose/Linear ICs
IC12	Display Drivers and Microcontroller Peripherals (planned)
IC13	Programmable Logic Devices (PLD)
IC14	8048-based 8-bit Microcontrollers
IC15	FAST TTL Logic Series
IC16	ICs for Clocks and Watches
IC18	Semiconductors for In-car Electronics and General Industrial Applications (planned)
IC17	RF/Wireless Communications
IC19	Semiconductors for Datacom: LANs, UARTs, Multi-protocol Controllers and Fibre Optics
IC20	80C51-based 8-bit Microcontrollers
IC21	68000-based 16-bit Microcontrollers (planned)
IC22	ICs for Multi-media Systems (planned)
IC23	QUBiC Advanced BiCMOS Bus Interface Logic ABT, MULTIBYTE™
IC24	Low Voltage CMOS & BiCMOS Logic

Discrete Semiconductors

Book	*Title*
SC01	Diodes
SC02	Power Diodes
SC03	Thyristors and Triacs
SC04	Small-signal Transistors
SC05	Low-frequency Power Transistors and Hybrid IC Power Modules
SC06	High-voltage and Switching NPN Power Transistors
SC07	Small-signal Field-effect Transistors
SC08a	RF Power Bipolar Transistors
SC08b	RF Power MOS Transistors
SC09	RF Power Modules
SC10	Surface Mounted Semiconductors
SC13	Power MOS Transistors including TOPFETs and IGBTs
SC14	RF Wideband Transistors, Video Transistors and Modules
SC15	Microwave Transistors
SC16	Wideband Hybrid IC Modules
SC17	Semiconductor Sensors

Professional Components

PC01	High-power Klystrons and Accessories
PC06	Circulators and Isolators

MORE INFORMATION FROM PHILIPS SEMICONDUCTORS?

For more information about Philips Semiconductors data handbooks, catalogs and subscriptions contact your nearest Philips Semiconductors national organization, select from the **address list on the back cover of this handbook**. Product specialists are at your service and inquiries are answered promptly.

Data handbook system — Appendix A

OVERVIEW OF PHILIPS COMPONENTS DATA HANDBOOKS

Our sister product division, Philips Components, also has a comprehensive data handbook system to support their products. Their data handbook titles are listed here.

Display Components

Book	*Title*
DC01	Colour Display Components Colour TV Picture Tubes and Assemblies Colour Monitor Tube Assemblies
DC02	Monochrome Monitor Tubes and Deflection Units
DC03	Television Tuners, Coaxial Aerial Input Assemblies
DC05	Flyback Transformers, Mains Transformers and General-purpose FXC Assemblies

Magnetic Products

MA01	Soft Ferrites
MA03	Piezoelectric Ceramics Specialty Ferrites
MA04	Dry-reed Switches

Passive Components

PA01	Electrolytic Capacitors
PA02	Varistors, Thermistors and Sensors
PA03	Potentiometers and Switches
PA04	Variable Capacitors
PA05	Film Capacitors
PA06	Ceramic Capacitors
PA07	Quartz Crystals for Special and Industrial Applications
PA08	Fixed Resistors
PA10	Quartz Crystals for Automotive and Standard Applications
PA11	Quartz Oscillaors

Professional Components

PC04	Photo Multipliers
PC05	Plumbicon Camera Tubes and Accessories
PC07	Vidicon and Newvicon Camera Tubes and Deflection Units
PC08	Image Intensifiers
PC12	Electron Multipliers

MORE INFORMATION FROM PHILIPS COMPONENTS?

For more information contact your nearest Philips Components national organizaiton shown in the following list.

Argentina: BUENOS AIRES, Tel. (541)541 4261, Fax. (541)786 9367.
Australia: NORTH RYDE, Tel. (02)805 4455, Fax. (02)805 4466.
Austria: WIEN, Tel (01)60101 1820, Fax. (01)60101 1210.
Belgium: BRUXELLES, Tel. (02)741 8211, Fax. (02)735 8667.
Brazil: SÃO PAULO, Tel. (011)829 1166, Fax. (011)829 1849.
Canada: SCARBOROUGH, Tel. (416)292 5161, Fax. (416)754 6248.
Chile: SANTIAGO, Tel. (02)773 816, Fax. (02)5602 735 3594.
China (Peoples Republic of): SHANGHAI, Tel. (021)3264140, Fax. (021)3202160.
Columbia: BOGOTA, Tel. (01)249 7624, Fax. (01)261 0139.
Denmark: COPENHAGEN, Tel. (032)883 333, Fax. (031)571 949.
Finland: ESPOO, Tel. (9)0-50261, Fax. (9)0-520971.
France: SURESNES, Tel. (01)4099 6161, Fax. (01)4099 6431.
Germany: HAMBURG, Tel. (040)3296-0, Fax. (040)3296 216.
Greece: TAVROS, Tel. (01)489 4339/(01)489 4911, Fax. (01)481 5180.
Hong Kong: KWAI CHUNG, Tel. (852)724 5121, Fax. (852)480 6960.
India: BOMBAY, Tel. (022)4938 541, Fax. (022)4938 722.
Indonesia: JAKARTA, Tel. (021)5201 122, Fax. (021)5205 189.
Ireland: DUBLIN, Tel. (01)640 203, Fax. (01)640 210.
Italy: MILANO, Tel. (02)6752.1, Fax. (02)6752 3300.
Japan: TOKIO, Tel. (03)3740 5143, Fax. (03)3740 5035.
Korea (Republic of): SEOUL, Tel. (02)709-1412, Fax. (02)709-1415.
Malaysia: KUALA LUMPUR, Tel. (03)757 5511, Fax. (03)757 4880.
Mexico: CHI HUA HUA, Tel. (016)18-67-01/(016)18-67-02, Fax. (016)778 0551.
Netherlands: EINDHOVEN, Tel. (040)7 83749, Fax. (040)7 88399.
New Zealand: AUKLAND, Tel. (09)849-4160, Fax. (09)849-7811.
Norway: OSLO, Tel. (22)74 8000, Fax. (22)74 8341.
Pakistan: KARACHI, Tel. (021)577 032, Fax. (021)569 1832.
Philippines: MANILA, Tel. (02)810-0161, Fax. (02)817-3474.
Portugal: LISBOA, Tel. (01)388 3121, Fax. (01)388 3208.
Singapore: SINGAPORE, Tel. (65)350 2000, Fax. (65)355 1758.
South Africa: JOHANNESBURG, Tel. (011)470-5434, Fax. (011)470-5494.
Spain: BARCELONA, Tel. (93)301 6312, Fax. (93)301 4243.
Sweden: STOCKHOLM, Tel. (08)632 2000, Fax. (08)632 2745.
Switzerland: ZÜRICH, Tel. (01)488 2211, Fax. (01)481 7730.
Taiwan: TAIPEI, Tel. (2)509 7666, Fax. (2)500 5912.
Thailand: BANGKOK, Tel (2)399-3280 to 9, (2)398-2083, Fax. (2)398-2080.
Turkey: ISTANBUL, Tel. (01)279 2770, Fax. (01)269 3094.
United Kingdom: LONDON, Tel. (071)580 6633, Fax. (071)636 0394.
United States: RIVIERA BEACH, Tel. (800)447-3762/(407)881-3200, Fax. (407)881-3300.
Uruguay: MONTEVIDEO, Tel. (02)704 044, Fax. (02)920 601.

For all other countries apply to: **Philips Components**.
Marketing Communications, Building BAE,
P.O. Box 218, 5600 MD, EINDHOVEN, The Netherlands
Telex 35000 phtcnl, Fax. +31–40–724547.

Philips Semiconductors — a worldwide company

Argentina: IEROD, Av. Juramento 1992 – 14.b (1428) BUENOS AIRES,
Tel. (541) 786 7633, Fax . (541) 786 9367

Australia: 34 Waterloo Road,NORTH RYDE, NSW 2113,
Tel. (02) 805-4455, Fax. (02) 805 4466

Austria: Triester Str. 64, A-1101 WIEN, P.O. Box 213,
Tel. (01) 60 101-1236, Fax. (01) 60101-1211

Belgium: Postbus 90050, 5600 PB EINDHOVEN, The Netherlands,
Tel. (31)40 783 749, Fax. (31)40 788 399

Brazil: Rua do Rocia 220 – 5th Floor, Suite 51
CEP: 04552-903 SAO PAULO-SP, Brazil
P.O. Box 7383–(01064-970),
Tel. (011) 829-1166, Fax (011) 829-1849

Canada: INTEGRATED CIRCUITS:
Tel. (800) 234-7381, Fax. (708) 296-8556
DISCRETE SEMICONDUCTORS: 601 Milner Ave.,
SCARBOROUGH, ONTARIO, M1B 1M8
Tel. (0416) 292-5161 ext. 2336, Fax. (0416) 292-4477

Chile: Av. Santa Maria 0760, SANTIAGO,
Tel. (02) 773 816, Fax. (02) 777 6730

Colombia: Carrera 21 No. 56-17, BOGOTA, D.E., P.O. Box 77621,
Tel. (571)217 4609, Fax (01) 217 4549

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. (032) 88 2636, Fax. (031) 57 1949

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. (9)0 50261, Fax. (9)0 520971

France: 4 rue du Port-aux-Vins, BP317
92156 SURESNES Cedex,
Tel. (01)4099 6161, Fax, (01)4099 6427

Germany: P.O. Box 10 63 23, 20095 HAMBURG,
Tel. (040) 3296-0, Fax. (040) 3296 213

Greece: No. 15, 25th March Street, GR 17778 TAVROS,
Tel. (01) 4894 339/4894 911, Fax. (01) 4814 240

Hong Kong: 15/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG,
Tel. (0)42 45 121, Fax. (0) 4806 960

India: PEICO ELECTRONICS & ELECTRICALS Ltd.,
Components Dept., Shivsagar Estate, Block 'A' ,
Dr. Annie Besant Rd., Worli, BOMBAY-400 018,
Tel. (022) 49 38 541, Fax. (022) 4938 722

Indonesia: Philips House, Jalan H.R. Rasuna Said Kav. 3-4,
P.O. Box 4252, JAKARTA 12950
Tel. (021) 5201 122, Fax (021) 5205 189

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. (01) 640 000, Fax. (01) 640 200

Italy: Viale F. Testi, 327, 20162–MILANO,
Tel. (02) 6752.1, Fax. (02) 6752 3350

Japan: Philips Bldg. 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. (03) 3740 5101, Fax. (03) 37400 570

Korea (Republic of): Philips House, 260-199 Itaewon-dong,
Yongsan-ku, SEOUL, Tel. (02) 794 5011, Fax. (02) 798 8022

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA,
SELANGOR, Tel. (03) 757 5511, Fax. (03) 757 4880

Mexico: Philips Components, '5900 Gateway East, Suite 200,
EL PASO, TX 79905, Tel. 9-5(800) 234-7381, Fax. (708) 296-8556

Netherlands: Postbus 90050, 5600 PB EINDHOVEN,
Tel. (040) 78 37 49, Fax. (040)78 83 99

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. (09) 849-4180, Fax. (09) 849-7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. (22) 74 8000, Fax (22)74 8341

Pakistan: Philips Markaz, M.A. Jinnah Rd., KARACHI-3,
Tel. (021) 577 039, Fax. (021) 569 1832

Philippines: PHILIPS SEMICONDUCTORS PHILIPPINES Inc.,
106 Valero St. Salcedo Village, P.O. Box 911, MAKATI,
Metro MANILA, Tel. (02) 810 0161, Fax. (02) 817 3474

Portugal: Av. Eng. Duarte Pacheco 6, 1009 LISBOA Codex,
Tel. (01) 683 121, Fax. (01) 658 013

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. (65) 350 2000, Fax. (65) 251 6500

South Africa: 195-215 Main Road, Martindale,
P.O. Box 7430, JOHANNESBURG 2000,
Tel. (011) 470-5433, Fax. (011) 470-5494

Spain: Balmes 22, 08007 BARCELONA,
Tel. (03) 301 6312, Fax. (03) 301 42 43

Sweden: Kottbygatan 7, Akalla. S-164 85 STOCKHOLM,
Tel. (0)8-632 2000, Fax. (0)8-632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. (01) 488 22 11, Fax. (01) 481 7730

Taiwan: 69, Min Sheng East Road, Sec. 3, P.O. Box 22978,
TAIPEI 10446, Tel. (2) 509 7666, Fax. (2)500 5899

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.
60/14 MOO 11, Bangna – Trad Road Km. 3
Prakanong, BANGKOK 10260
Tel. (2)399-3280 to 9, (2)398-2083, Fax. (2)398-2080

Turkey: Talatpasa Cad. No. 5, 80640 LEVENT/ISTANBUL,
Tel. (0212) 279 2770, Fax. (0212) 269 3094

United Kingdom: Philips Semiconductors Limited, P.O. Box 65,
Philips House, Torrington Place, LONDON WC1E 7HD,
Tel. (071) 436 4144, Fax. (071) 323 0342

United States: INTEGRATED CIRCUITS:
811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. (800) 234-7381, Fax. (708) 296-8556
DISCRETE SEMICONDUCTORS: 2001 West Blue Heron Blvd.,
P.O. Box 10330, RIVIERA BEACH, FLORIDA 33404,
Tel. (800) 447-3762 and (407)881-3200, Fax. (407) 881-3300

Uruguay: Coronel Mora 433, MONTEVIDEO,
Tel. (02) 70 4044, Fax (02) 92 0601

For all other countries apply to: Philips Semiconductors,
Marketing Communications, Building BAF-1,
P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands,
Telex 35000 phtcnl, Fax +31-40-724825

Printed in the USA 2093B/24M/CR2/1293 904pp

Document order number: 9398 652 47011
Document order number USA: 98-7001-230-02

Philips Semiconductors

PHILIPS

PHILIPS