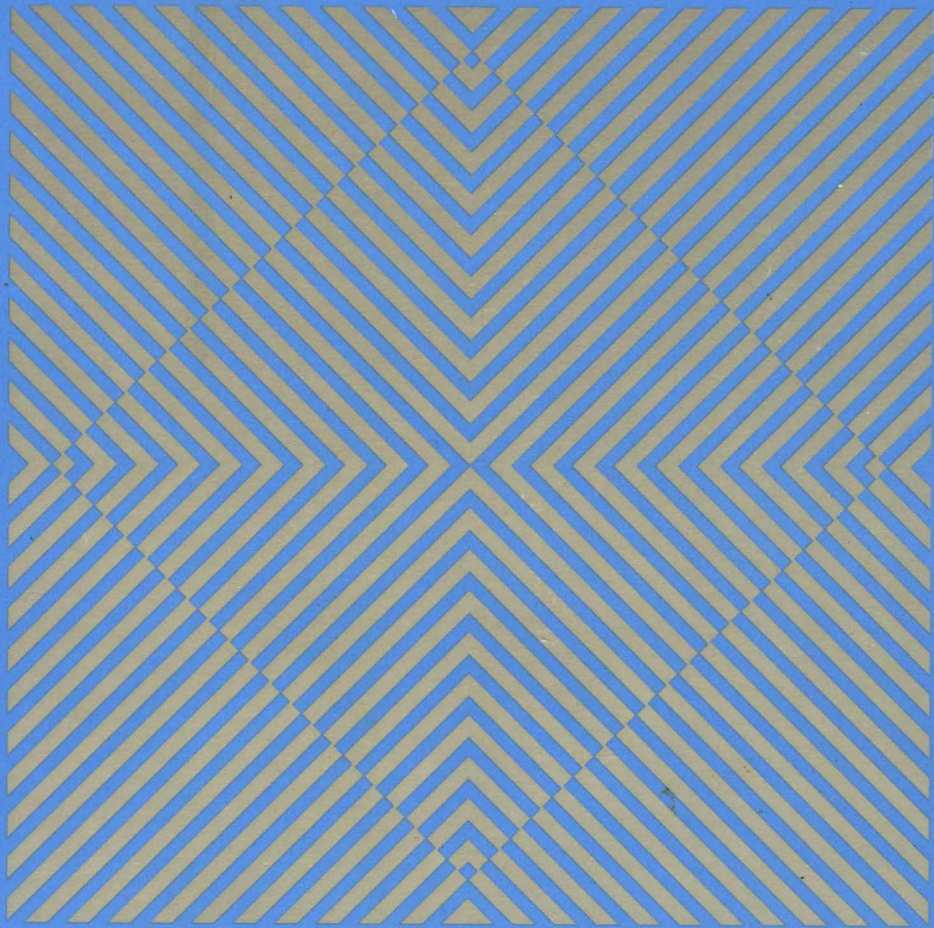


MITSUBISHI SEMICONDUCTORS

M37732 Group M37730 Group

USER'S MANUAL



USER'S MANUAL

M37732 Group
M37730 Group

 MITSUBISHI ELECTRIC

 MITSUBISHI
ELECTRIC

Preface

This manual describes the hardware of the Mitsubishi CMOS 16-bit microcomputers M37732 group and M37730 group. After reading this manual, the user should be able to fully utilize the functions of the microcomputers of M37732 group and M37730 group.

For details concerning the software for the M37732 group and M37730 group, refer to the MELPS 7700 SOFTWARE MANUAL. For details concerning the development support tools (assembler, option boards), refer to the respective operation manuals.

Table of contents

CHAPTER 1. DESCRIPTION

1.1 M37732 group	3
1.2 Performance overview	4
1.3 Pin configuration	5
1.4 Pin description	6
1.5 Block diagram	8

CHAPTER 2. FUNCTIONAL DESCRIPTION

2.1 Central processing unit (CPU)	10
2.1.1 Accumulator (Acc)	10
2.1.2 Index register X (X)	10
2.1.3 Index register Y (Y)	10
2.1.4 Stack pointer (S)	12
2.1.5 Program counter (PC)	13
2.1.6 Program bank register (PG)	13
2.1.7 Data bank register (DT)	14
2.1.8 Direct page register (DPR)	14
2.1.9 Processor status register (PS)	15
2.2 Internal bus interface	17
2.2.1 Internal bus interface overview	17
2.2.2 Bus interface unit functions	17
2.2.3 Bus interface unit operations	19
2.2.4 Data read/write operations	21
2.3 Addressable memory space	22
2.3.1 Banks	22
2.3.2 Direct page	22
2.4 Memory allocation	24
2.4.1 SFR (Special Function Registers), internal RAM, external memory area	24
2.4.2 Processor mode	27
2.4.3 External memory area bus control	28
2.5 Input/Output pins	30
2.5.1 Programmable I/O ports	30
2.5.2 Ports P4–P8 functions	33
2.5.3 Pin functions	35
2.6 Interrupts	40
2.6.1 Interrupt functions	40
2.6.2 Sources of interrupts	42
2.6.3 Interrupt control	44
2.6.4 Interrupt priority level	47
2.6.5 Interrupt priority level detection circuit	47
2.6.6 Interrupt priority level detection time	50
2.6.7 Interrupt processing sequence	51
2.6.8 Returning from an interrupt service routine	54
2.6.9 Multiple interrupts	54
2.6.10 Interrupt response time and interrupt delay time	55
<i>[Precautions when using interrupts]</i>	56

2.7 Timer A	58
2.7.1 Timer A description	58
2.7.2 Block description	59
2.7.3 Timer mode	66
[Precautions when using timer mode]	71
2.7.4 Event counter mode	72
[Precautions when using event counter mode]	78
2.7.5 One-shot pulse mode	79
[Precautions when using one-shot pulse mode]	83
2.7.6 Pulse width modulation (PWM) mode	84
[Precautions when using PWM mode]	88
2.8 Timer B	89
2.8.1 Timer B description	89
2.8.2 Block description	89
2.8.3 Timer mode	95
2.8.4 Event counter mode	98
2.8.5 Pulse period/pulse width measurement mode	100
[Precautions when using timer B]	103
2.9 Pulse output function	104
2.9.1 Pulse output function overview	104
2.9.2 Block description	105
2.9.3 Operation description	115
[Precautions when using pulse output function (pulse output port mode)]	121
2.10 Serial I/O	122
2.10.1 Serial I/O description	122
2.10.2 Block description	123
2.10.3 Clock synchronous serial I/O	135
[Precautions during clock synchronous serial I/O receive]	144
2.10.4 Clock asynchronous serial I/O (UART)	145
2.10.5 Sleep mode	157
2.11 A-D converter	158
2.11.1 A-D converter overview	158
2.11.2 Block description	159
2.11.3 Successive approximation conversion	167
2.11.4 A-D conversion mode	169
[Precautions when using A-D converter]	171
2.12 Watchdog timer	172
2.12.1 Operation description	173
2.13 Hold function	174
2.14 Ready function	176
CHAPTER 3. RESET	
3.1 Reset	178
3.1.1 Reset operation	178
3.1.2 Internal registers state at reset	179
3.1.3 Bus state during reset	181
3.2 Reset circuit	182
3.3 Software reset	183
CHAPTER 4. OSCILLATING CIRCUIT	
4.1 Oscillating circuit	186
4.1.1 Circuit using a ceramic resonator or a crystal resonator	186
4.1.2 External clock input circuit	186
4.2 Clock generating circuit	187

CHAPTER 5. ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics	190
5.1.1 Absolute maximum ratings	190
5.1.2 Recommended operating conditions	191
5.1.3 Electrical characteristics and A-D converter characteristics	192
5.1.4 Timing requirements	195
5.1.5 Switching characteristics	198
5.1.6 Equations for calculating the parameters	200
5.1.7 Timing diagrams	201

CHAPTER 6. STANDARD CHARACTERISTICS

6.1 Standard characteristics	208
6.1.1 Standard port characteristics	208
6.1.2 $I_{CC}-f(X_{IN})$ standard characteristics	209
6.1.3 A-D converter standard characteristics	210

CHAPTER 7. LOW SUPPLY VOLTAGE VERSION

7.1 Description	212
7.1.1 Low supply voltage version	212
7.1.2 Performance overview	214
7.1.3 Pin description	215
7.1.4 Block diagram	217
7.2 Functional description	218
7.2.1 Reset	218
7.2.2 Oscillating circuit	220
7.3 Electrical characteristics	221
7.3.1 Absolute maximum ratings	221
7.3.2 Recommended operating conditions	222
7.3.3 Electrical characteristics and A-D converter characteristics	223
7.3.4 Timing requirements	224
7.3.5 Switching characteristics	227
7.3.6 Equations for calculating the parameters	229
7.3.7 Timing diagrams	230
7.4 Standard characteristics	235
7.4.1 Standard port characteristics	235
7.4.2 $I_{CC}-f(X_{IN})$ standard characteristics	236
7.4.3 A-D converter standard characteristics	237

CHAPTER 8. M37730 GROUP

8.1 Description	240
8.1.1 M37730 group	240
8.1.2 Performance overview	241
8.1.3 Pin configuration	242
8.1.4 Pin description	244
8.1.5 Block diagram	246
8.2 Functional description	247
8.2.1 Addressable memory space	248
8.2.2 Memory allocation	249
8.2.3 Input/Output pins	253
8.2.4 Interrupts	258
8.2.5 Timer A, Timer B	260
8.2.6 Pulse output function	262
8.2.7 Serial I/O	269
8.2.8 Watchdog timer, Hold, Ready function	270
8.2.9 Reset	270

8.3 Electrical characteristics	272
8.3.1 Absolute maximum ratings	272
8.3.2 Recommended operating conditions	273
8.3.3 Electrical characteristics	274
8.3.4 Timing requirements	277
8.3.5 Switching characteristics	280
8.3.6 Equations for calculating the parameters	282
8.3.7 Timing diagrams	283
 CHAPTER 9. APPLICATION	
9.1 Memory connection	290
9.1.1 Memory connection model	290
9.1.2 Memory access time calculation	292
9.1.3 Memory connection precautions	296
9.1.4 Connection to devices that request long access time	298
9.1.5 Memory connection examples	299
9.2 I/O expansion	306
9.2.1 I/O expansion model	306
9.2.2 I/O expansion examples	306
9.3 Low supply voltage version applications	310
9.3.1 Memory connection	310
9.4 Power saving	317
9.4.1 Wait mode	317
9.4.2 Stop mode	320
9.5 Pulse motor control application	322
9.6 Application system examples	326
9.7 Program examples	329
9.7.1 Hardware definition examples	330
9.7.2 Initialization examples	334
9.7.3 Timer setting examples	335
9.7.4 Pulse output port mode setting examples	345
9.7.5 Serial I/O modes setting examples	347
9.7.6 A-D conversion modes setting examples	354
9.7.7 Interrupt processing examples	358
9.7.8 Watchdog timer setting examples	361
9.7.9 Software timer setting examples	363
9.7.10 Interrupt vector table setting example	364
 APPENDIX	
Appendix 1. SFR area memory map	366
Appendix 2. Control registers	370
Appendix 3. Stop, wait, one-wait, Ready, Hold state	385
Appendix 4. Package outlines	387
Appendix 5. Setting example of unused pins	390
Appendix 6. Print foot pattern examples	391
Appendix 7. System development precautions	399
Appendix 8. M66800SP/FP	400
Appendix 9. Machine instructions	402
Appendix 10. Instruction code table	416

List of figures

CHAPTER 1.DESCRPTION

Fig. 1.3.1 M37732S4FP pin configuration (top view)	5
Fig. 1.5.1 M37732S4FP block diagram	8

CHAPTER 2.FUNCTIONAL DESCRIPTION

Fig. 2.1.1 CPU registers structure	11
Fig. 2.1.2 Stored registers when an interrupt request is accepted	12
Fig. 2.1.3 Program counter PC and program bank register PG	13
Fig. 2.1.4 Setting example of direct page with the direct page register (DPR)	14
Fig. 2.1.5 Processor status register structure	15
Fig. 2.2.1 Bus interface unit block diagram	17
Fig. 2.2.2 BIU registers	18
Fig. 2.2.3 Bus interface unit operating waveform	20
Fig. 2.3.1 Addressable memory space	23
Fig. 2.4.1 Memory map	25
Fig. 2.4.2 SFR area memory map	26
Fig. 2.4.3 Processor mode register structure	27
Fig. 2.4.4 Effect of wait bit for external area access (when BYTE pin="L" level)	29
Fig. 2.5.1 Memory allocation of direction registers and port registers	30
Fig. 2.5.2 Port Pi (i=4 to 8) direction registers structure	31
Fig. 2.5.3 Relationship between port registers and pins	32
Fig. 2.5.4 Port peripheral circuits	34
Fig. 2.5.5 Relationship between address bus and output pins	35
Fig. 2.5.6 Relationship between data bus and I/O pins (when BYTE pin="L" level)	36
Fig. 2.5.7 Bus timing when external bus width is 16 bits (when BYTE pin="L" level)	36
Fig. 2.5.8 Relationship between data bus and I/O pins (when BYTE pin="H" level)	37
Fig. 2.5.9 Bus timing when external bus width is 8 bits (when BYTE pin="H" level)	37
Fig. 2.6.1 Interrupt mechanism	40
Fig. 2.6.2 State of the stack area when an interrupt request is accepted	41
Fig. 2.6.3 Interrupt control register memory map	44
Fig. 2.6.4 Interrupt control register structure	45
Fig. 2.6.5 Hardware controlled interrupt priority levels	47
Fig. 2.6.6 Interrupt priority level detection circuit	48
Fig. 2.6.7 Interrupt priority level detection model	49
Fig. 2.6.8 Interrupt priority level detection time	50
Fig. 2.6.9 Interrupt processing sequence	51
Fig. 2.6.10 INTACK sequence timing	52
Fig. 2.6.11 Register storing operation	53
Fig. 2.6.12 Multiple interrupt mechanism	54
Fig. 2.6.13 Interrupt response time and interrupt delay time	55
Fig. 2.6.14 \overline{INT}_i Interrupt during level sense mode	56
Fig. 2.6.15 Repeating \overline{INT}_i interrupt (level sense)	56
Fig. 2.6.16 \overline{INT}_i interrupt control register setting flow	57

Fig. 2.7.1 Timer Ai block diagram	59
Fig. 2.7.2 Count start flag register structure	61
Fig. 2.7.3 One-shot start flag register structure	61
Fig. 2.7.4 Up-down flag register structure	62
Fig. 2.7.5 Timer Ai mode register structure	63
Fig. 2.7.6 Timer Ai interrupt control register structure	64
Fig. 2.7.7 Relationship between port P5 and P6 direction registers and timer pins	65
Fig. 2.7.8 Timer Ai mode register structure in timer mode	67
Fig. 2.7.9 Setting example of the timer mode related registers	66
Fig. 2.7.10 Timer mode operation diagram (neither pulse output nor gate function)	68
Fig. 2.7.11 Timer operation example when the gate function is selected	69
Fig. 2.7.12 Timer operation example when the pulse output function is selected	70
Fig. 2.7.13 Reading the timer Ai register	71
Fig. 2.7.14 Timer Ai mode register structure in event counter mode	73
Fig. 2.7.15 Up-down flag register structure	73
Fig. 2.7.16 Setting example of the event counter mode related registers	74
Fig. 2.7.17 Event counter mode operation diagram (when up-down switching factor selection bit is "0")	75
Fig. 2.7.18 Timer Aj mode register (j=2 to 4) when two-phase pulse signal processing function is selected	76
Fig. 2.7.19 Timers A2 and A3 two-phase pulse signal processing operation	77
Fig. 2.7.20 Timer A4 two-phase pulse signal processing operation (quadruple processing)	77
Fig. 2.7.21 Reading the timer Ai register	78
Fig. 2.7.22 The characteristics of the two-phase pulse signal	78
Fig. 2.7.23 Timer Ai mode register structure in one-shot pulse mode	80
Fig. 2.7.24 One-shot start flag register structure	80
Fig. 2.7.25 Setting example of the one-shot pulse mode related registers	81
Fig. 2.7.26 One-shot pulse mode operation (when external trigger is selected)	82
Fig. 2.7.27 Timer Ai mode register structure in PWM mode	85
Fig. 2.7.28 Count start flag register structure	84
Fig. 2.7.29 16-bit PWM mode output waveform example	87
Fig. 2.7.30 8-bit PWM mode output waveform example	88
Fig. 2.8.1 Timer Bi block diagram	89
Fig. 2.8.2 Count start flag register structure	90
Fig. 2.8.3 Timer Bi mode register structure	91
Fig. 2.8.4 Timer Bi interrupt control register structure	93
Fig. 2.8.5 Relationship between port P6 direction register and timer pins	94
Fig. 2.8.6 Timer Bi mode register structure in timer mode	95
Fig. 2.8.7 Setting example of the timer mode related registers	96
Fig. 2.8.8 Timer mode operation diagram	97
Fig. 2.8.9 Timer Bi mode register structure in event counter mode	98
Fig. 2.8.10 Setting example of the event counter mode related registers	99
Fig. 2.8.11 Timer Bi mode register structure in pulse period/pulse width measurement mode	101
Fig. 2.8.12 Pulse period/pulse width measurement mode operation diagram in pulse period measurement mode	102
Fig. 2.8.13 Pulse period/pulse width measurement mode operation diagram in pulse width measurement mode	102
Fig. 2.8.14 Reading the timer Bi register	103
Fig. 2.9.1 Pulse output port mode block diagram	105
Fig. 2.9.2 Port P5 output control circuit	106
Fig. 2.9.3 Count start flag register structure	107
Fig. 2.9.4 Timer A0, A2 mode registers structure (in pulse output port mode)	108
Fig. 2.9.5 Timer A1, A3 mode registers structure (using pulse width modulation in pulse output port mode)	109
Fig. 2.9.6 Timer A1, A3 mode registers structure (not using pulse width modulation in pulse output port mode)	110
Fig. 2.9.7 Waveform output mode register structure	111
Fig. 2.9.8 Relationship between pulse output data register j and pulse output pins	113
Fig. 2.9.9 Timer Ai interrupt control register structure	114
Fig. 2.9.10 Pulse output pins and pulse output data register j structure	115

Fig. 2.9.11 RTP0 pulse output example (positive polarity, not modulated)	117
Fig. 2.9.12 RTP0 pulse output example (positive polarity, modulated)	119
Fig. 2.9.13 RTP0 pulse output example (negative polarity, modulated)	120
Fig. 2.10.1 Serial I/O operating modes	122
Fig. 2.10.2 Serial I/O block diagram	123
Fig. 2.10.3 UARTi transmit/receive mode register structure	124
Fig. 2.10.4 UARTi transmit/receive control register 0 structure	128
Fig. 2.10.5 UARTi transmit/receive control register 1 structure	129
Fig. 2.10.6 Serial I/O transmission block diagram	131
Fig. 2.10.7 Serial I/O receive block diagram	132
Fig. 2.10.8 Shift clock generation block diagram	132
Fig. 2.10.9 UARTi transmission interrupt control register and UARTi receive interrupt control register structure	133
Fig. 2.10.10 Relationship between port P8 direction register and serial I/O pins	134
Fig. 2.10.11 Setting example of clock synchronous serial I/O related registers at transmitting	137
Fig. 2.10.12 Clock synchronous serial I/O transmit operation	139
Fig. 2.10.13 Clock synchronous serial I/O timing diagram at transmitting (selecting internal clock)	139
Fig. 2.10.14 Setting example of clock synchronous serial I/O related registers at receiving	141
Fig. 2.10.15 Connecting example of clock synchronous serial I/O	142
Fig. 2.10.16 Clock synchronous serial I/O receive operation	143
Fig. 2.10.17 Clock synchronous serial I/O timing diagram at receiving (selecting external clock)	143
Fig. 2.10.18 Data format for transfer	147
Fig. 2.10.19 Data format example	148
Fig. 2.10.20 Setting example of UART related registers at transmitting	150
Fig. 2.10.21 8-bit UART timing diagram at transmitting (with parity and 1 stop bit)	152
Fig. 2.10.22 9-bit UART timing diagram at transmitting (no parity and 2 stop bits)	152
Fig. 2.10.23 Setting example of UART related registers at receiving	154
Fig. 2.10.24 Connecting example of UART	155
Fig. 2.10.25 8-bit UART timing diagram at receiving (no parity and 1 stop bit)	156
Fig. 2.10.26 Sleep mode	157
Fig. 2.11.1 A-D converter block diagram	159
Fig. 2.11.2 A-D control register structure	160
Fig. 2.11.3 A-D sweep pin selection register structure	163
Fig. 2.11.4 A-D register i structure	164
Fig. 2.11.5 A-D conversion interrupt control register structure	165
Fig. 2.11.6 Relationship between port P7 direction register and analog input pins	166
Fig. 2.12.1 Watchdog timer block diagram	172
Fig. 2.12.2 Watchdog timer frequency selection flag structure	173
Fig. 2.13.1 Timing example when the Hold	175
Fig. 2.14.1 Timing example when the Ready	176

CHAPTER 3. RESET

Fig. 3.1.1 Internal processing sequence after removing reset state	178
Fig. 3.1.2 Internal registers state at reset (1)	179
Fig. 3.1.2 Internal registers state at reset (2)	180
Fig. 3.1.2 Internal registers state at reset (3)	181
Fig. 3.2.1 Power-on reset condition	182
Fig. 3.2.2 Power-on reset circuit example	182
Fig. 3.3.1 Processor mode register structure	183

CHAPTER 4. OSCILLATING CIRCUIT

Fig. 4.1.1 Oscillating circuit using a ceramic resonator	186
Fig. 4.1.2 Oscillating circuit using a crystal resonator	186
Fig. 4.1.3 External clock input circuit	186
Fig. 4.2.1 Clock generating circuit block diagram	187

CHAPTER 5. ELECTRICAL CHARACTERISTICS

Fig. 5.1.1 Testing circuit for ports	199
--	-----

CHAPTER 7. LOW SUPPLY VOLTAGE VERSION

Fig. 7.1.1 M37732S4LGP pin configuration (top view)	213
Fig. 7.1.2 M37732S4LGP block diagram	217
Fig. 7.2.1 Power-on reset condition.....	218
Fig. 7.2.2 Power-on reset circuit example	219
Fig. 7.2.3 Oscillating circuit example using a ceramic resonator.....	220
Fig. 7.3.1 Testing circuit for ports	228

CHAPTER 8. M37730 GROUP

Fig. 8.1.1 M37730S2FP pin configuration (top view)	242
Fig. 8.1.2 M37730S2SP pin configuration (top view)	243
Fig. 8.1.3 M37730S2FP block diagram	246
Fig. 8.2.1 Addressable memory space	248
Fig. 8.2.2 Memory map	250
Fig. 8.2.3 SFR area memory map	251
Fig. 8.2.4 Processor mode register structure	252
Fig. 8.2.5 Memory allocation of direction registers and port registers	253
Fig. 8.2.6 Port P4–P6, P8 direction registers structure	254
Fig. 8.2.7 Relationship between port P4–P6, P8 registers and pins.....	254
Fig. 8.2.8 Port peripheral circuit.....	256
Fig. 8.2.9 Interrupt control registers memory map	258
Fig. 8.2.10 Memory map of timer related control registers.....	260
Fig. 8.2.11 Relationship between port P6 direction register and timer pins	261
Fig. 8.2.12 Count start flag register structure	261
Fig. 8.2.13 Pulse output port mode block diagram	262
Fig. 8.2.14 Pulse output pins and pulse output data register 0, 1 structure	263
Fig. 8.2.15 Relationship between port P6 direction register and pins	264
Fig. 8.2.16 Waveform output mode register structure	265
Fig. 8.2.17 How to resume pulse output after disabling it with $\overline{INT0}$ input signal	266
Fig. 8.2.18 Timer A3 mode register structure in timer mode (not using pulse width modulation for RTP 1)	267
Fig. 8.2.19 Timer A3 mode register structure in event counter mode (not using pulse width modulation for RTP1).....	268
Fig. 8.2.20 Memory map of serial I/O related control registers	269
Fig. 8.2.21 Relationship between port P8 direction register and serial I/O pins	269
Fig. 8.2.22 Internal registers state at reset (1).....	270
Fig. 8.2.22 Internal registers state at reset (2).....	271
Fig. 8.3.1 Testing circuit for ports	281

CHAPTER 9. APPLICATION

Fig. 9.1.1 Bus timing diagram	293
Fig. 9.1.2 Relationship between memory access time and external clock input frequency	294
Fig. 9.1.3 Relationship between data setup time for write and external clock input frequency	295
Fig. 9.1.4 Memory data read timing	296
Fig. 9.1.5 Bus buffer usage example (1)	297
Fig. 9.1.6 Bus buffer usage example (2) (connection to device that requests long hold time during write)	298
Fig. 9.1.7 Ready generating circuit example	299
Fig. 9.1.8 Minimum model memory connection example (1)	300
Fig. 9.1.9 Minimum model memory connection example (2)	300
Fig. 9.1.10 Maximum model memory connection example	301

List of figures

Fig. 9.1.11 Memory connection example at 20MHz using software one-wait	302
Fig. 9.1.12 Memory connection example at 25MHz using software one-wait	302
Fig. 9.1.13 Memory connection example at 25MHz using software one-wait and Ready function	303
Fig. 9.1.14 Memory connection example using "M66800" (1)	304
Fig. 9.1.15 Memory connection example using "M66800" (2)	305
Fig. 9.1.16 Memory connection example using "M66800" (3)	305
Fig. 9.2.1 Port expansion example using M5M82C55AP-2/FP-2	307
Fig. 9.2.2 Port expansion example using M66500SP/FP	308
Fig. 9.2.3 Port expansion example using M37451M4-XXXFP	309
Fig. 9.3.1 Relationship between memory access time and external clock input frequency	312
Fig. 9.3.2 Relationship between write data setup time and external clock input frequency	312
Fig. 9.3.3 Minimum model memory connection example (without external RAM)	313
Fig. 9.3.4 Minimum model memory connection example (with external RAM)	314
Fig. 9.3.5 Maximum model memory connection example	315
Fig. 9.3.6 Ready generating circuit example	316
Fig. 9.4.1 Example of WIT instruction execution circuit	318
Fig. 9.4.2 Example of WIT instruction execution program	319
Fig. 9.4.3 Example of STP instruction execution program for memory back-up	321
Fig. 9.4.4 Example of power-on reset circuit for internal RAM back-up	321
Fig. 9.5.1 Pulse output port mode setting procedure	322
Fig. 9.5.2 Example of initialization for stepping motor control	323
Fig. 9.5.3 Example of updating pulse output data and output timing for stepping motor control	324
Fig. 9.5.4 Example of output waveform for stepping motor 1-2 excitation (without pulse width modulation)	324
Fig. 9.5.5 RTP0 _o –RTP1 _s output control block diagram	325
Fig. 9.6.1 Serial printer	326
Fig. 9.6.2 Hard disk (HDD)	327
Fig. 9.6.3 ISDN digital telephone	327
Fig. 9.6.4 Cellular phone	328

APPENDIX

Fig. 1 Setting example of unused pins	390
Fig. 2 Print foot pattern example for 80P6N package	391
Fig. 3 IC socket (type name : IC61-0804-034) by YAMAICHI ELECTRONICS Co., Ltd. outline	392
Fig. 4 IC socket (type name : IC61-0804-046) by YAMAICHI ELECTRONICS Co., Ltd. outline	393
Fig. 5 Print foot pattern example for 80P6S package	394
Fig. 6 80P6-80P6S conversion adapter	395
Fig. 7 Print foot pattern example for 80P6D package	396
Fig. 8 Print foot pattern example for 64P6N package	397
Fig. 9 IC socket (type name : IC61-0644-052) by YAMAICHI ELECTRONICS Co., Ltd. outline	398
Fig. 10 Print foot pattern example (parts view)	399
Fig. 11 Print foot pattern example (solder view)	399
Fig. 12 M66800SP/FP pin configuration (top view)	401

List of tables

CHAPTER 1.DESCRPTION

Table 1.1.1 M37732 group	3
Table 1.2.1 M37732S4FP performance overview	4
Table 1.4.1 Pin description (1)	6
Table 1.4.1 Pin description (2)	7

CHAPTER 2.FUNCTIONAL DESCRIPTION

Table 2.2.1 Functions of BIU registers and buffers	17
Table 2.2.2 Data bus state according to \bar{E} and R/W	19
Table 2.5.1 Function of ports P4-P8	33
Table 2.6.1 Interrupt sources and vector address	42
Table 2.6.2 Internal interrupt request sources	43
Table 2.6.3 \overline{INT}_i Interrupt request sources	43
Table 2.6.4 Setting of interrupt priority levels	47
Table 2.6.5 Interrupt enable levels corresponding to IPL setting	47
Table 2.6.6 Change in IPL	52
Table 2.7.1 Timer Ai register memory allocation	60
Table 2.7.2 Timer Ai register read and write	60
Table 2.7.3 Relationship between operating mode selection bits and operating modes	63
Table 2.7.4 Relationship between count source selection bits and count sources	63
Table 2.7.5 Relationship between gate function selection bits and effective level	68
Table 2.8.1 Timer Bi register memory allocation	90
Table 2.8.2 Relationship between operating mode selection bits and operating modes	92
Table 2.8.3 Relationship between count source selection bits and count sources	92
Table 2.8.4 Relationship between count polarity selection bits and effective edge	99
Table 2.8.5 Relationship between measurement mode selection bits and measured intervals	100
Table 2.9.1 Pulse output function overview	104
Table 2.9.2 Timer Ai register memory allocation	107
Table 2.9.3 Timer Ai register read and write	107
Table 2.9.4 Relationship between waveform output mode selection bits and waveform output mode	112
Table 2.9.5 Waveform output mode	115
Table 2.10.1 Relationship between serial I/O mode selection bits and serial I/O modes	125
Table 2.10.2 Function of port P8 when serial I/O is selected	125
Table 2.10.3 Relationship between BRG count source selection bits and count source	127
Table 2.10.4 Clock synchronous serial I/O performance overview	135
Table 2.10.5 UART performance overview	145
Table 2.10.6 Baud rate selection table (1)	146
Table 2.10.6 Baud rate selection table (2)	146
Table 2.10.7 Transmission data in UART mode	148
Table 2.10.8 Setting of UART mode	149
Table 2.10.9 Setting of UART mode	153
Table 2.11.1 A-D converter performance overview	158
Table 2.11.2 Relationship between analog input selection bits and analog input pins	161
Table 2.11.3 Relationship between A-D conversion mode selection bits and conversion modes	161
Table 2.11.4 Relationship between A-D conversion frequency selection flag and A-D converter operating clock and conversion time	162
Table 2.11.5 Relationship between A-D sweep pin selection bits and number of analog input pins	163
Table 2.11.6 Relationship between analog input pin and A-D register i	164
Table 2.11.7 Relationship between successive approximation register and Vref	167
Table 2.11.8 Change in successive approximation register and compare reference voltage during A-D conversion	168
Table 2.13.1 Microcomputer state during Hold	174
Table 2.14.1 Microcomputer state during Ready	176

CHAPTER 3. RESET

Table 3.1.1 Bus state during reset 181

CHAPTER 5. ELECTRICAL CHARACTERISTICS

Table 5.1.1 Equations for calculating parameters 200

CHAPTER 7. LOW SUPPLY VOLTAGE VERSION

Table 7.1.1 Low supply voltage version 212
 Table 7.1.2 M37732S4LGP performance overview 214
 Table 7.1.3 Pin description (1) 215
 Table 7.1.3 Pin description (2) 216
 Table 7.3.1 Equations for calculating parameters 229

CHAPTER 8. M37730 GROUP

Table 8.1.1 M37730 group 240
 Table 8.1.2 M37730S2FP performance overview 241
 Table 8.1.3 Pin description (1) 244
 Table 8.1.3 Pin description (2) 245
 Table 8.2.1 Differences between M37730 group and M37732 group 247
 Table 8.2.2 Function of ports P4–P6, P8 257
 Table 8.2.3 Interrupt sources and vector address 259
 Table 8.3.1 Equations for calculating parameters 282

CHAPTER 9. APPLICATION

Table 9.1.1 Memory connection model 291
 Table 9.1.2 Parameter equation and constants depending on the external clock input frequency 292
 Table 9.1.3 Address hold time and data floating time 296
 Table 9.1.4 Usable memory without bus buffer 297
 Table 9.3.1 Parameter equation and constants depending on the external clock input frequency 311

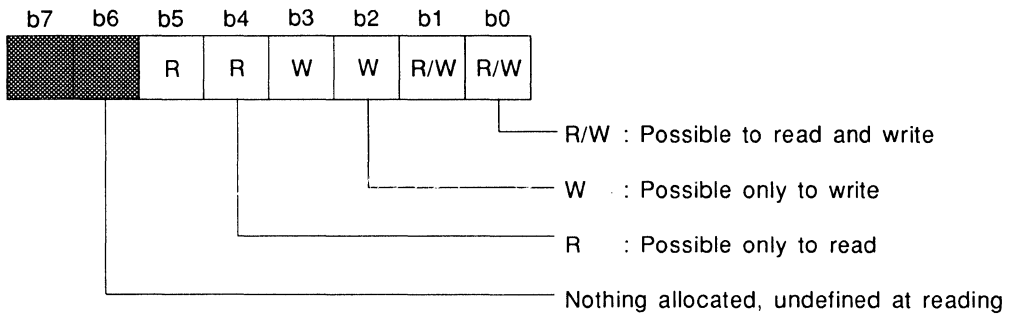
APPENDIX

Table 1 Stop, wait, one-wait, Ready, Hold state 386
 Table 2 External interrupts used to remove stop mode 385
 Table 3 Number of NOP instructions to insert 385
 Table 4 Setting example of unused pins 390

For using this manual

This manual defines following items.

- $f(X_{IN})$
 $f(X_{IN})$ means external clock input frequency.
- Internal clock ϕ
Internal clock ϕ means operating clock of this microcomputer. It is obtained by dividing the input clock to X_{IN} pin by 2 ($= f(X_{IN})/2$).
- Clock ϕ_1
Clock ϕ_1 means the internal clock ϕ output from ϕ_1 pin.
- Bit attribute
Bit attributes are described in the figure of register structure.
The following abbreviations are used to indicate the attributes.



•Overflow and Underflow of timers

Overflow of the timer means that the counter content reaches $FFFF_{16}$ → reload value "n".
Underflow of the timer means that the counter content reaches 0000_{16} → reload value "n".

"n" : Value set in reload register



CHAPTER 1

DESCRIPTION

- 1.1 M37732 group
- 1.2 Performance overview
- 1.3 Pin configuration
- 1.4 Pin description
- 1.5 Block diagram

DESCRIPTION

The M37732 group is an external ROM version 16-bit microcomputer designed with high-performance silicon gate CMOS technology. It is housed in an 80-pin plastic molded QFP.

This microcomputer has a large 16M-byte address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can be also switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.

M37732 group is also equipped with low supply voltage version that is suitable for application of portable equipment controller.

M37730 group is a microcomputer housed in a 64-pin package. M37730 group has the same functions as M37732 group except for a few parts. Therefore, the differences between M37730 group and M37732 group are mainly described in "CHAPTER 8. M37730 GROUP".

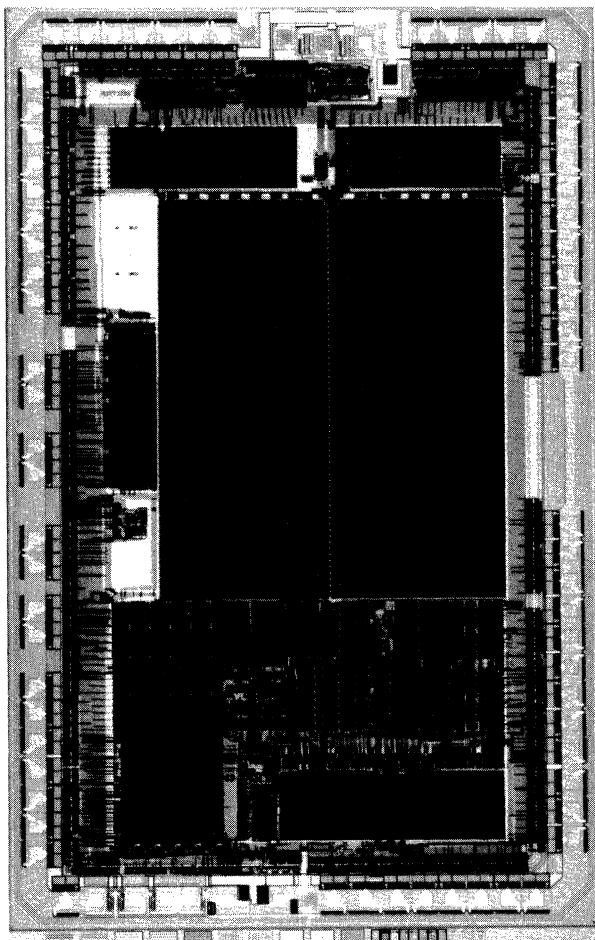


Photo of M37732S4LHP Chip

DESCRIPTION

1.1 M37732 group

1.1 M37732 group

The M37732 group consists of chips shown in Table 1.1.1 with the M37732S4FP as the base chip. The M37732S4LGP and the M37732S4LHP are the low supply voltage version microcomputers that have characteristics of low supply voltage and a small package. Except for low supply voltage version, the difference between the other chips is only the operating clock frequencies. Hereafter, the M37732 group microcomputers will be referred to simply as the M37732S4FP unless there is a specific difference by version.

Low supply voltage version is described in "CHAPTER 7. LOW SUPPLY VOLTAGE VERSION".

Table 1.1.1 M37732 group

Type name	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)	Package outline	Remarks
M37732S4FP	External	2048	8	80P6N-A	High-speed version of M37732S4FP
M37732S4AFP			16		
M37732S4BFP			25		
M37732S4LGP			8	80P6S-A	Low supply voltage version and small package of M37732S4FP
M37732S4LHP			8	80P6D-A	

DESCRIPTION

1.2 Performance overview

1.2 Performance overview

Table 1.2.1 shows the performance overview of the M37732S4FP.

Table 1.2.1 M37732S4FP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37732S4FP	500ns (the fastest instruction at 8MHz frequency)
	M37732S4AFP	250ns (the fastest instruction at 16MHz frequency)
	M37732S4BFP	160ns (the fastest instruction at 25MHz frequency)
Operating clock frequency	M37732S4FP	8MHz (maximum)
	M37732S4AFP	16MHz (maximum)
	M37732S4BFP	25MHz (maximum)
Memory size	ROM	External
	RAM	2048 bytes
Input/Output ports	Ports P5–P8	8 bits × 4
	Port P4	5 bits × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits × 5
	TB0, TB1, TB2	16 bits × 3
Serial I/O		(UART or clock synchronous serial I/O) × 2
A-D converter		8 bits × 1 (8 channels)
Watchdog timer		12 bits × 1
Interrupts		3 external, 16 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic or crystal resonator)
Supply voltage		5V±10%
Power dissipation		30mW (operating clock frequency = 8MHz)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		–20 to 85°C
Device structure		High-performance silicon gate CMOS process
Package		80-pin plastic molded QFP

DESCRIPTION

1.3 Pin configuration

1.3 Pin configuration

Figure 1.3.1 shows the M37732S4FP pin configuration.

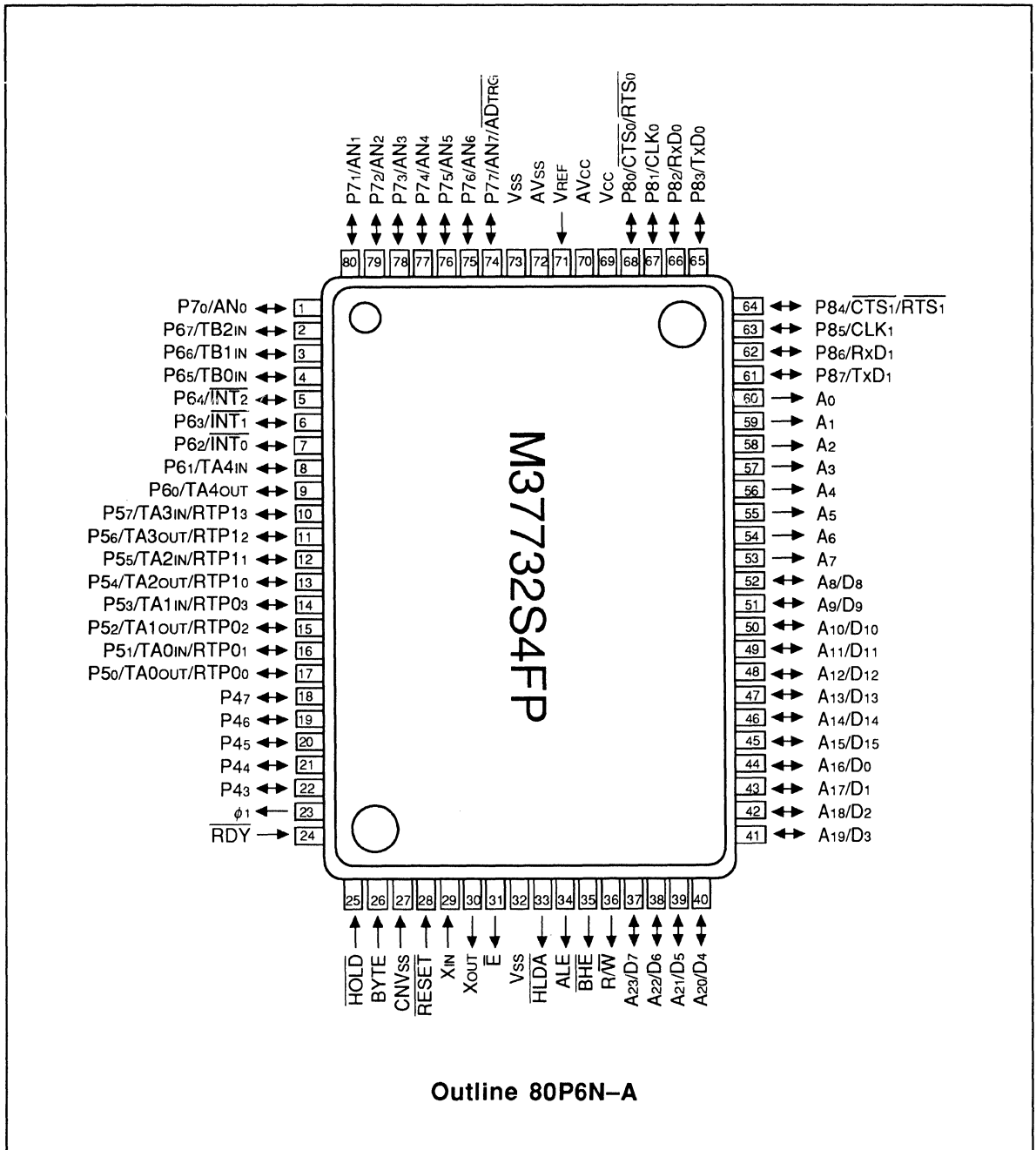


Fig. 1.3.1 M37732S4FP pin configuration (top view)

DESCRIPTION

1.4 Pin description

1.4 Pin description

Table 1.4.1 shows the pin description.

Table 1.4.1 Pin description (1)

Pin	Name	Input/Output	Functions
V _{CC} , V _{SS}	Power supply		Supply 5V±10% to V _{CC} and 0V to V _{SS} .
CNV _{SS}	CNV _{SS} input	Input	Connect to V _{CC} .
RESET	Reset input	Input	The microcomputer is reset when this pin is set to "L" level.
X _{IN}	Clock input	Input	These are the I/O pins of the internal clock generating circuit. Connect a ceramic or crystal resonator between X _{IN} and X _{OUT} . When an external clock is used, the clock source should be connected to the X _{IN} pin and the X _{OUT} pin should be left open.
X _{OUT}	Clock output	Output	
\bar{E}	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L" level.
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when the signal level is "L" and 8 bits when the signal level is "H".
AV _{CC} , AV _{SS}	Analog supply input		Power supply for the A-D converter. Externally connect AV _{CC} to V _{CC} and AV _{SS} to V _{SS} .
V _{REF}	Reference voltage input	Input	This is a reference voltage input pin for the A-D converter.
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X _{IN} pin by 2.
RDY	Ready input	Input	The microcomputer is in Ready state while "L" level is input to this pin.
HOLD	Hold input	Input	The microcomputer is in Hold state while "L" level is input to this pin.
HLDA	Hold acknowledge output	Output	This pin outputs "L" level when microcomputer is in Hold state.
R/W	Read/Write output	Output	The read/write signal indicates the data bus state. The data bus state is read when this signal is "H" level and write when "L" level.
\bar{BHE}	Byte high enable output	Output	"L" level is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to obtain only the address from address and data multiplex signals.
A ₀ –A ₇	Address bus low-order output	Output	Low-order 8 bits (A ₀ –A ₇) of the address bus are output.
A ₈ /D ₈ –A ₁₅ /D ₁₅	Address bus middle-order output/ Data bus high-order I/O	I/O	<ul style="list-style-type: none"> ●When the external bus width is 16 bits. (The BYTE pin is at "L" level.) High-order 8 bits (D₈–D₁₅) of the data bus are input or output when the \bar{E} output is "L" level. Middle-order 8 bits (A₈–A₁₅) of the address bus are output when \bar{E} output is "H" level. ●When the external bus width is 8 bits. (The BYTE pin is at "H" level.) Middle-order 8 bits (A₈–A₁₅) of the address bus are output.
A ₁₆ /D ₀ –A ₂₃ /D ₇	Address bus high-order output/ Data bus low-order I/O	I/O	Low-order 8 bits (D ₀ –D ₇) of the data bus are input or output when the \bar{E} output is "L" level. High-order 8 bits (A ₁₆ –A ₂₃) of the address bus are output when the \bar{E} output is "H" level.

DESCRIPTION

1.4 Pin description

Table 1.4.1 Pin description (2)

Pin	Name	Input/Output	Functions
P4 ₃ –P4 ₇	I/O port P4	I/O	Port P4 is a 5-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output.
P5 ₀ –P5 ₇	I/O port P5	I/O	Port P5 is an 8-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output. These pins also function as I/O pins for timers A0–A3.
P6 ₀ –P6 ₇	I/O port P6	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as I/O pins for timer A4, external interrupt input pins for INT ₀ –INT ₂ , and input pins for timers B0–B2.
P7 ₀ –P7 ₇	I/O port P7	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as analog input pins AN ₀ –AN ₇ . P7 ₇ also functions as the AD _{TRG} input pin for an A-D conversion trigger.
P8 ₀ –P8 ₇	I/O port P8	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as CTS/RTS, CLK, Rx/D, Tx/D pins for UART0 and UART1.

DESCRIPTION

1.5 Block diagram

1.5 Block diagram

Figure 1.5.1 shows the M37732S4FP block diagram.

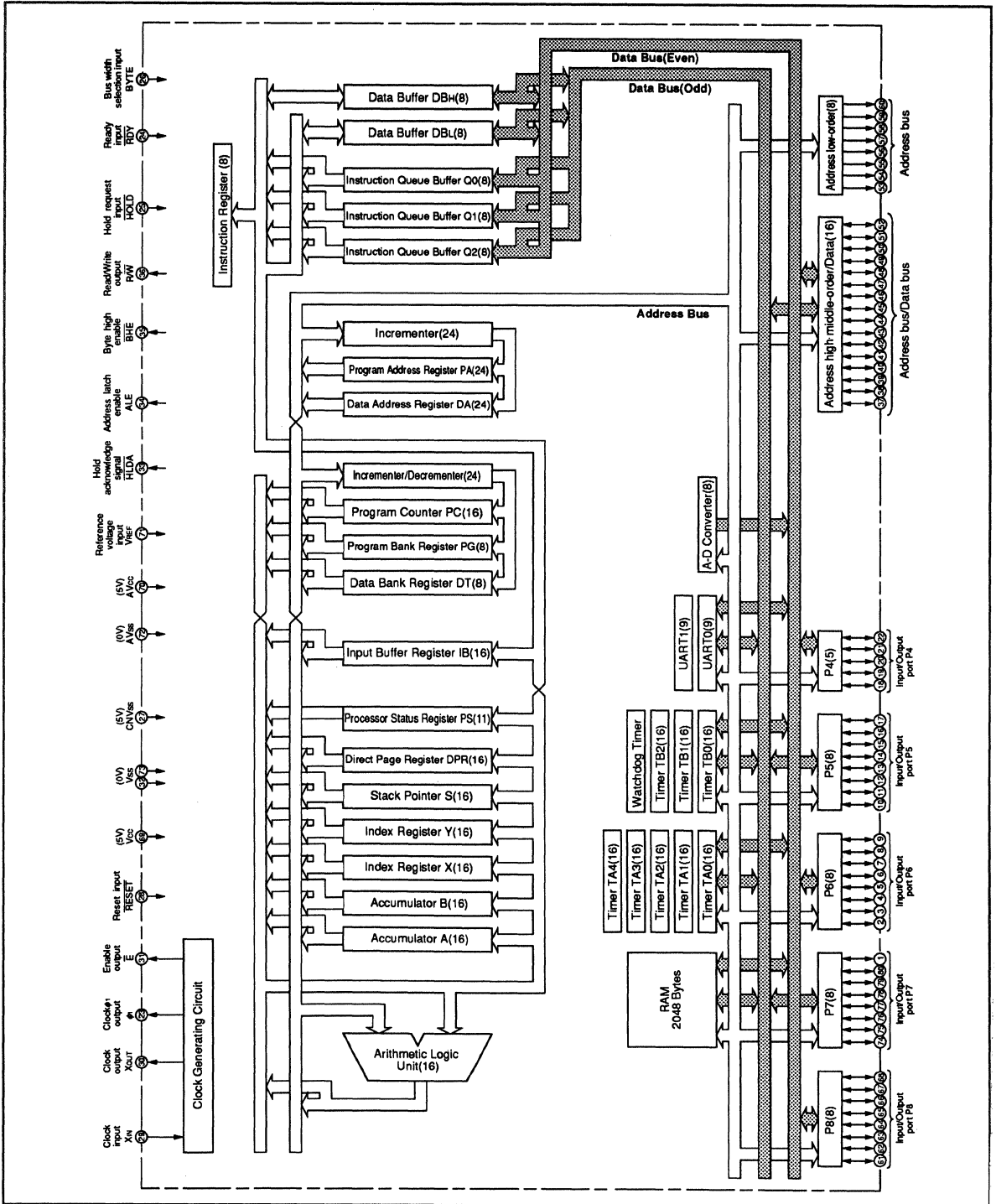


Fig. 1.5.1 M37732S4FP block diagram



CHAPTER 2

FUNCTIONAL DESCRIPTION

- 2.1 Central processing unit (CPU)
- 2.2 Internal bus interface
- 2.3 Addressable memory space
- 2.4 Memory allocation
- 2.5 Input/Output pins
- 2.6 Interrupts
- 2.7 Timer A
- 2.8 Timer B
- 2.9 Pulse output function
- 2.10 Serial I/O
- 2.11 A-D converter
- 2.12 Watchdog timer
- 2.13 Hold function
- 2.14 Ready function

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

2.1 Central processing unit (CPU)

The MELPS 7700 CPU has ten registers as shown in Figure 2.1.1. Each of these registers is described below.

2.1.1 Accumulator (Acc)

Accumulators A and B are available and each can be used as 8-bit or 16-bit register as necessary.

(1) Accumulator A (A)

Accumulator A is the main register of the microcomputer. Data operations such as calculations, data transfer, and input/output are executed mainly through accumulator A. It consists of 16 bits and the low-order 8 bits can be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag m is "0" and as an 8-bit register when flag m is "1". Flag m is a part of the processor status register (PS) which is described later. When an 8-bit register is selected, the low-order 8 bits of the accumulator A are used and the contents of the high-order 8 bits are unchanged.

(2) Accumulator B (B)

Accumulator B has the same functions as accumulator A. The MELPS 7700 instructions can use accumulator B instead of accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A. Accumulator B is also controlled by the data length flag r

2.1.2 Index register X (X)

Index register X consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag x is "0" and as an 8-bit register when flag x is "1". Flag x is a part of the processor status register (PS) which is described later. When an 8-bit register is selected, the low-order 8 bits of the index register X are used and the contents of the high-order 8 bits are unchanged. In addressing mode in which the index register X is used as the index register, the contents of this register is added to obtain the real address.

Also, when executing the block transfer instruction **MVP** or **MVN**, the contents of the index register indicate the low-order 16 bits of the source data address. The third byte of the **MVP** or **MVN** instruction is the high-order 8 bits of the source data address.

2.1.3 Index register Y (Y)

Index register Y is a 16-bit register with the same function as index register X. As with index register X, the index register length flag (x) determines whether this register is used as a 16-bit register or as an 8-bit register. Also, when executing the block transfer instruction **MVP** or **MVN**, the contents of index register Y indicate the low-order 16 bits of the destination data address. The second byte of the **MVP** or **MVN** instruction is the high-order 8 bits of the destination data address.

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

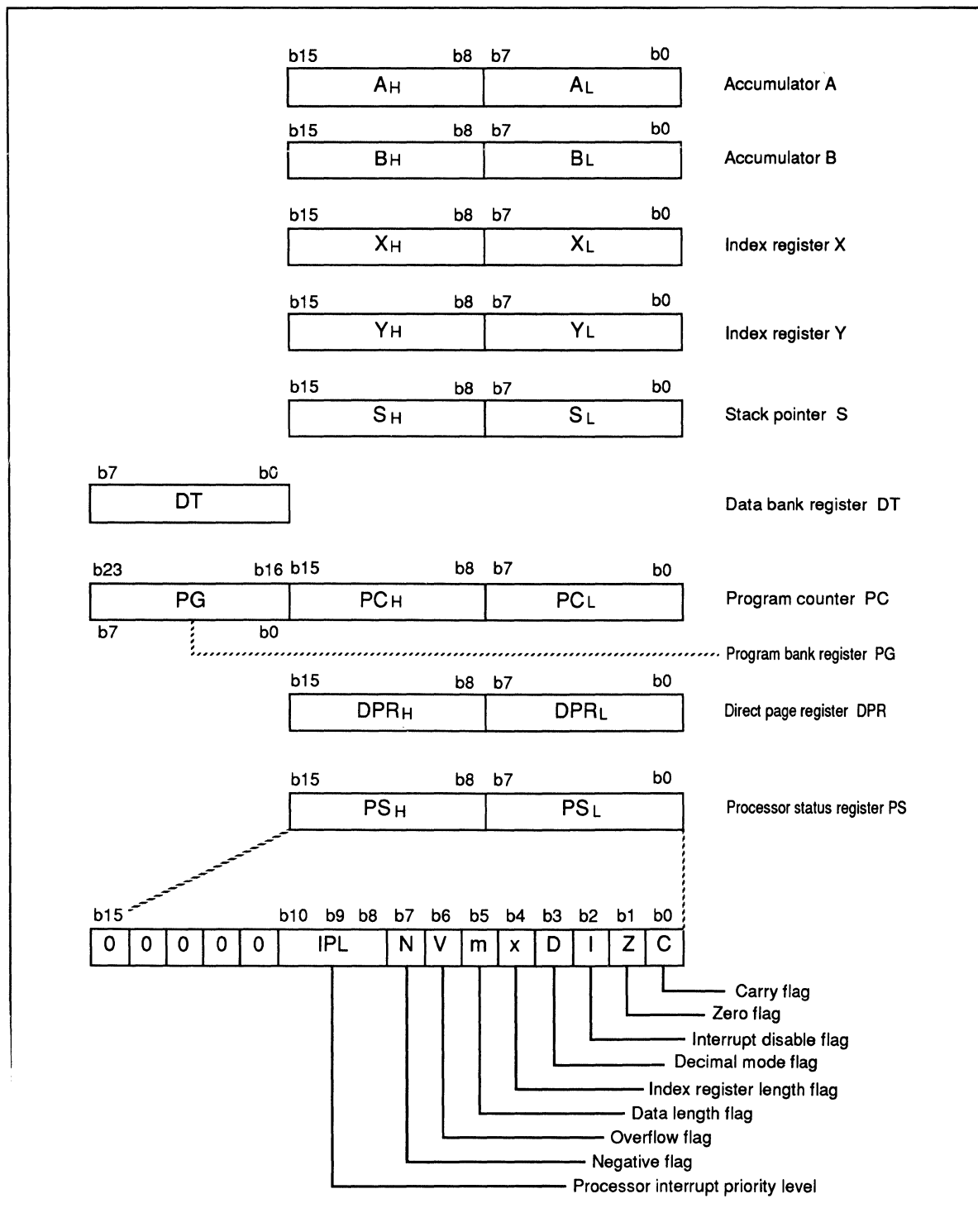


Fig. 2.1.1 CPU registers structure

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

2.1.4 Stack pointer (S)

Stack pointer S is a 16-bit register. It is used during a subroutine call or interrupt. It is also used during addressing modes using the stack. The contents of the stack pointer S indicate the address (stack area) for storing registers during subroutine calls and interrupts. The bank 0_{16} must be designated for the stack area (refer to section "2.3 Addressable memory space"). Normally, the stack area is reserved in internal RAM.

When an interrupt request is accepted, the contents of the program bank register PG is stored at the address indicated by the contents of the stack pointer S, and the contents of the stack pointer S are decremented by 1. Then the contents of the program counter PC and the processor status register PS are stored with the high-order bytes and the low-order bytes (PCH, PCL, PSH, PSL) in that order. The contents of the stack pointer S after accepting an interrupt request are equal to the contents of the stack pointer S before the accepting of interrupt request decremented by 5.

Figure 2.1.2 shows the stored registers when an interrupt request is accepted.

When returning to the original routine after processing the interrupt service routine, the registers stored in the stack area are restored to the original registers in the reverse sequence and the contents of the stack pointer are returned to the state before the accepting of interrupt request.

The same operation is performed during a subroutine call, but the contents of the processor status register PS are not stored (the contents of the program bank register PG may not be stored either depending on the addressing mode).

The user is responsible for storing registers other than those described above by program during interrupts or subroutine calls.

In addition, the stack pointer S must be initialized at the beginning of the program because its contents are undefined at reset. Normally, the stack pointer is initialized with the highest address of the internal RAM. The contents of the stack area change when subroutines are nested or when multiple interrupt requests are accepted. Therefore, make sure necessary data in the internal RAM are not destroyed when nesting subroutines.

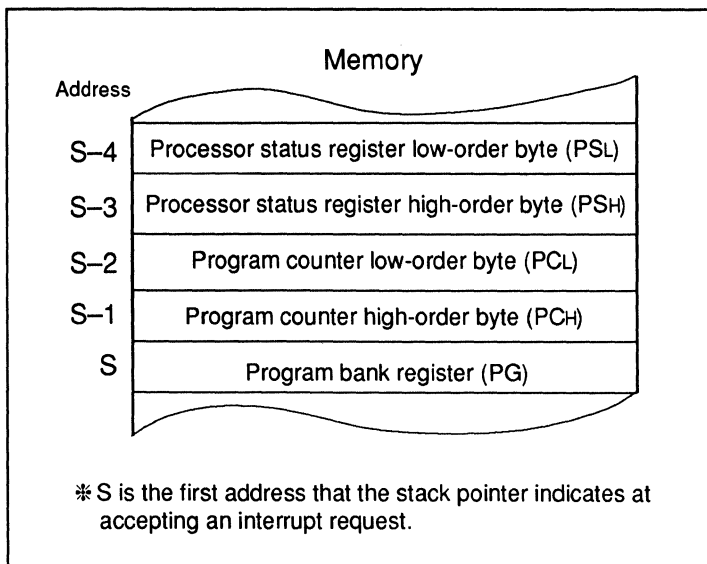


Fig. 2.1.2 Stored registers when an interrupt request is accepted

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

2.1.5 Program counter (PC)

Program counter PC is a 16-bit counter that indicates the low-order 16 bits of the next program memory address (24 bits) to be executed. The contents of the high-order program counter (PC_H) become "FF₁₆", and the low-order program counter (PC_L) become "FE₁₆" at reset. The contents of the program counter PC become the contents of the reset vector address (addresses FFFE₁₆, FFFF₁₆) after removing reset state. Figure 2.1.3 shows the program counter PC and the program bank register PG.

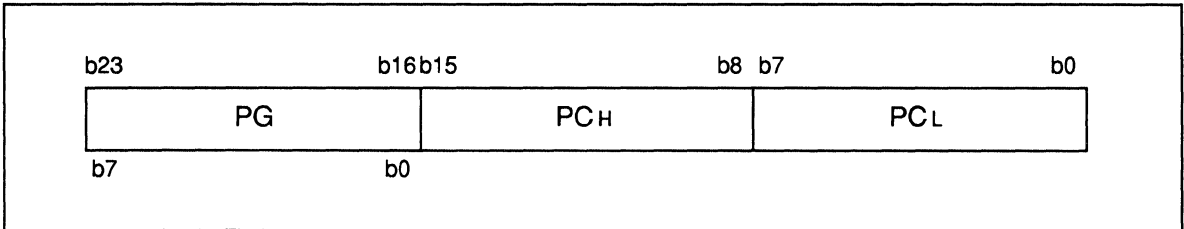


Fig. 2.1.3 Program counter PC and program bank register PG

2.1.6 Program bank register (PG)

Program bank register PG is an 8-bit register that indicates the high-order 8 bits (bank) of the next program memory address (24 bits) to be executed. When a carry occurs after adding the contents of the program counter PC, the contents of the program bank register PG are incremented by 1. Also, when a borrow occurs after subtracting the contents of the program counter PC, the contents of the program bank register PG are decremented by 1. Therefore, programs can be written without considering bank boundaries, usually.

This register is cleared to "00₁₆" at reset.

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

2.1.7 Data bank register (DT)

Data bank register DT is an 8-bit register. With some addressing modes using the data bank register DT, the contents of this register are used as the high-order 8 bits (bank) of a 24-bit address. Use the LDT instruction to set the value in this register. This register is cleared to "00₁₆" at reset.

* Refer to "MELPS 7700 SOFTWARE MANUAL" for the addressing modes.

2.1.8 Direct page register (DPR)

Direct page register DPR is a 16-bit register. The contents of this register indicate whether the direct page area is allocated in bank 0₁₆ or spans across banks 0₁₆ and 1₁₆. This area can be accessed with two bytes by using the direct page addressing mode.

The contents of the DPR are the base address (lowermost address) of the direct page area which extends 256 bytes above this address. The DPR can contain a value from 0000₁₆ to FFFF₁₆. If it contains a value equal to or greater than "FF01₁₆", the direct page area spans across banks 0₁₆ and 1₁₆. If the low-order 8 bits of the DPR is "00₁₆", the number of cycles required to generate an address is minimized. Therefore, the low-order 8 bits of the DPR should normally be set to "00₁₆".

This register is cleared to "0000₁₆" at reset. Figure 2.1.4 shows the setting example of the direct page with the direct page register (DPR).

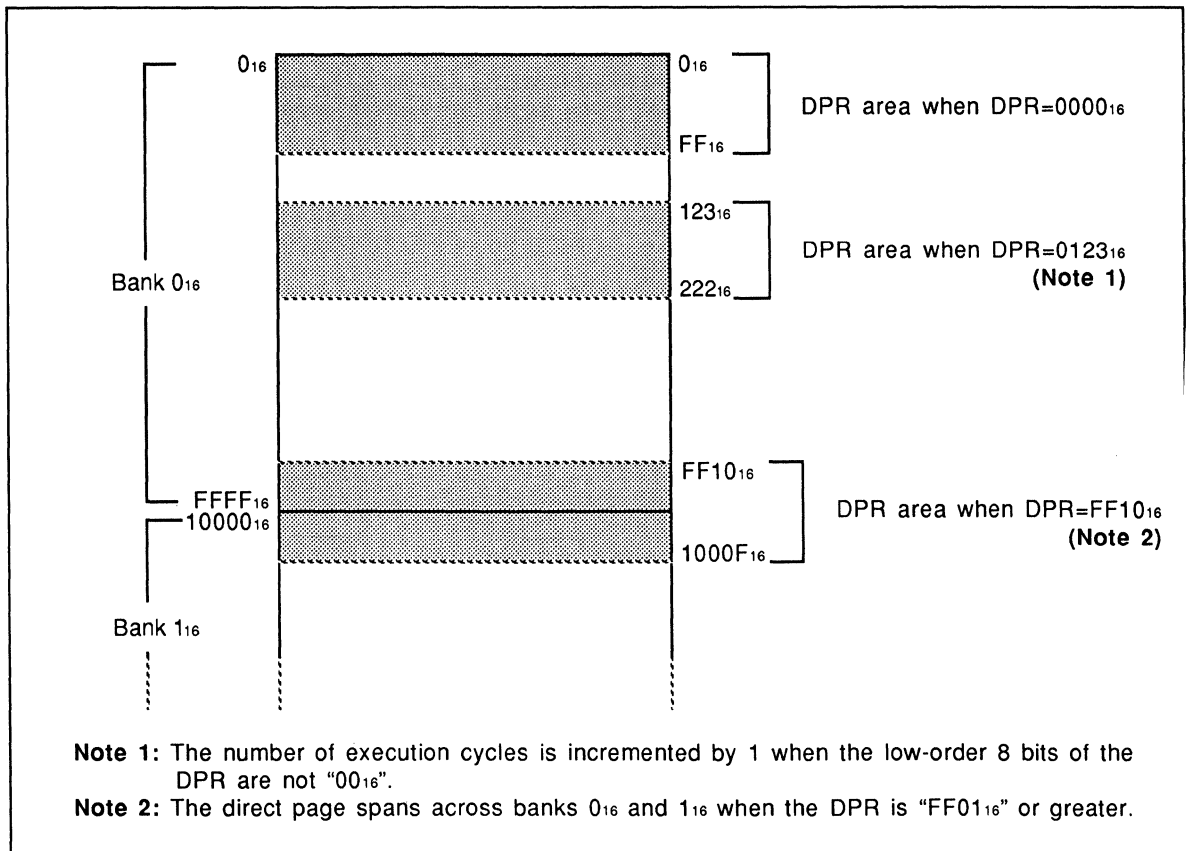


Fig. 2.1.4 Setting example of direct page with the direct page register (DPR)

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

2.1.9 Processor status register (PS)

Processor status register is an 11-bit register. It consists of flags to indicate the result of operation and processor interrupt priority level. The flags C, Z, V, and N are tested by branch instructions.

Figure 2.1.5 shows the register structure of the processor status register.

The details of the processor status register bits are described below.

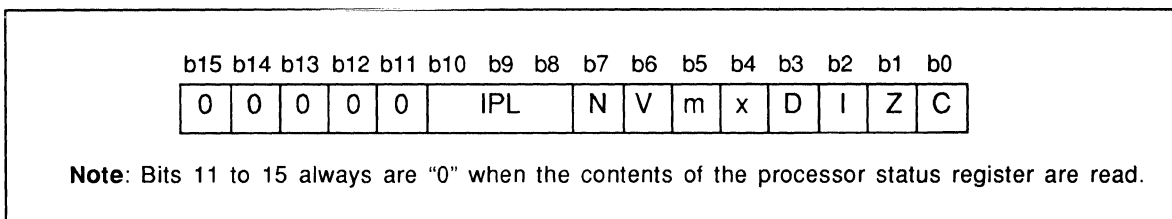


Fig. 2.1.5 Processor status register structure

(1) Carry flag (C)

The carry flag is assigned to bit 0 of the processor status register. It contains the carry or borrow bit from the arithmetic and logic unit (ALU) after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set with the **SEC** or **SEP** instruction and cleared with the **CLC** or **CLP** instruction.

(2) Zero flag (Z)

The zero flag is assigned to bit 1 of the processor status register. It is set to "1" if the result of an arithmetic operation or data transfer is zero, and cleared to "0" if otherwise. This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction.

Note: This flag has no meaning in decimal mode addition (the **ADC** instruction).

(3) Interrupt disable flag (I)

The interrupt disable flag is assigned to bit 2 of the processor status register. It disables all maskable interrupts (interrupts other than watchdog timer, the **BRK** instruction, and zero division). Interrupts are disabled when this flag is "1". When an interrupt request is accepted, it is set to "1" automatically to prevent multiple interrupts. This flag can be set with the **SEI** or **SEP** instruction and cleared with the **CLI** or **CLP** instruction. This flag is set to "1" at reset.

(4) Decimal mode flag (D)

The decimal mode flag is assigned to bit 3 of the processor status register. It determines whether addition and subtraction are performed in binary or decimal. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as two or four digit decimal (determined by the data length flag m). Decimal adjust is performed automatically. Decimal operation is possible only with the **ADC** and **SBC** instructions. This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction. This flag is cleared to "0" at reset.

(5) Index register length flag (x)

The index register length flag is assigned to bit 4 of the processor status register. It determines whether the index register X or index register Y is used as a 16-bit register or an 8-bit register. The register is used as a 16-bit register when flag x is "0" and as an 8-bit register when it is "1". This flag can be set with the **SEP** instruction and cleared with the **CLP** instruction. This flag is cleared to "0" at reset.

* When transferring between different bit lengths, the data is transferred with the length of the destination register, but except for the **TXA**, **TYA**, **TXB**, and **TYB** instructions. Refer to "MELPS 7700 SOFTWARE MANUAL".

FUNCTIONAL DESCRIPTION

2.1 Central processing unit

(6) Data length flag (m)

The data length flag is assigned to bit 5 of the processor status register. It determines whether to treat data as 16-bit or as 8-bit. A data is treated as 16-bit when flag m is "0" and as 8-bit when it is "1". This flag can be set with the **SEM** or **SEP** instruction and cleared with the **CLM** or **CLP** instruction. This flag is cleared to "0" at reset.

* When transferring between different bit lengths, the data is transferred with the length of the destination register, but except for the **TXA**, **TYA**, **TXB**, and **TYB** instructions. Refer to "MELPS 7700 SOFTWARE MANUAL".

(7) Overflow flag (V)

The overflow flag is assigned to bit 6 of the processor status register. It is used when adding or subtracting a word as signed binary. In case the data length flag m is "0", the overflow flag is set to "1" when the result of addition or subtraction is outside the range between -32768 and +32767, and cleared to "0" in all other cases. In case the data length flag m is "1", the overflow flag is set to "1" when the result of addition or subtraction is outside the range between -128 and +127, and cleared to "0" in all other cases. The overflow flag can be set with the **SEP** instruction and cleared with the **CLV** or **CLP** instructions.

Note: This flag has no meaning in decimal mode.

(8) Negative flag (N)

The negative flag is assigned to bit 7 of the processor status register. It is set when the result of arithmetic operation or data transfer is negative (data bit 15 is 1 when data length flag m is "0", or data bit 7 is 1 when data length flag m is "1"). It is cleared in all other cases. It can be set with the **SEP** instruction and cleared with the **CLP** instruction.

Note: This flag has no meaning in decimal mode.

(9) Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) is assigned to bits 8, 9, and 10 of the processor status register. These three bits determine the priority level of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority level of the requested interrupt (set with the interrupt control register) is higher than IPL. When an interrupt request is accepted, the IPL is stored in the stack and IPL is replaced by the interrupt priority level of the accepted interrupt request. This simplifies control of multiple interrupts.

There are no instructions to directly set or clear the IPL. It can be changed by placing the new IPL on the stack and updating the processor status register with the **PUL** or **PLP** instruction.

The contents of the IPL are cleared to "000" at reset.

FUNCTIONAL DESCRIPTION

2.2 Internal bus interface

2.2 Internal bus interface

A bus interface unit (BIU) is provided between the CPU and the internal bus.

2.2.1 Internal bus interface overview

Transfer of data between the CPU and memory or I/O device is always performed through the BIU provided between the CPU and the internal bus. When the CPU reads data from memory or I/O device, it sends the address to be read to the BIU. The BIU reads the data from the specified address and the CPU receives the data from the BIU. Similarly, the CPU sends the address to be written to the BIU when writing data. Thus the BIU controls the transfer of data between the CPU and bus.

Figure 2.2.1 shows the block diagram of the bus interface unit.

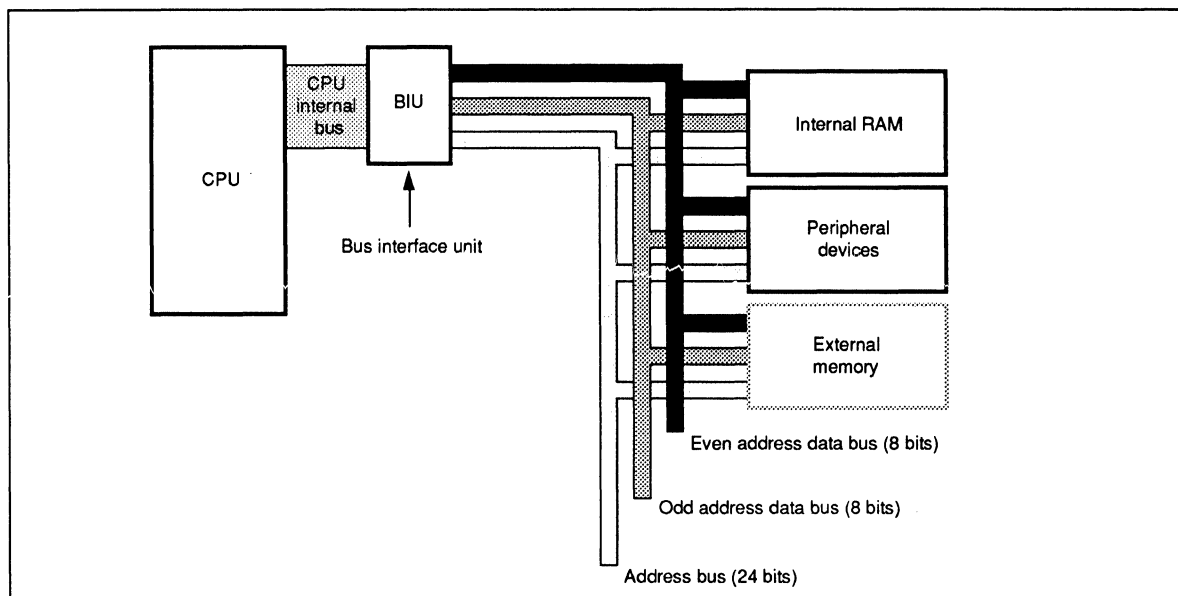


Fig. 2.2.1 Bus interface unit block diagram

2.2.2 Bus interface unit functions

The M37732 group uses the internal clock ϕ ($=f(X_{IN})/2$) as the standard clock. The CPU also uses internal clock ϕ as the standard clock. However, since the CPU clock may be extended due to CPU wait under certain conditions, it is referred to as ϕ_{CPU} to distinguish it from clock ϕ .

The M37732 group internal bus (address bus and data bus) operates at timing \bar{E} which is slower than internal clock ϕ . The operating clock of the CPU is different from the bus cycle because timing \bar{E} is normally $f(X_{IN})/4$. Therefore, the BIU is provided between the CPU and bus to synchronize the transfer of data to and from memory and I/O device. The BIU enables the CPU to transfer data to and from memory through the bus without decreasing the instruction execution speed.

The BIU consists of four registers as shown in Figure 2.2.2. Table 2.2.1 shows the functions of each register of BIU and buffer.

Table 2.2.1 Functions of BIU registers and buffers

Name	Function
Program address register	Indicates the address that stores the next instruction to prefetch in the instruction queue buffer.
Instruction queue buffer	A three-byte buffer for temporarily holding instruction prefetched from memory.
Data address register	Indicates the address to be read from or to be written to memory or I/O.
Data buffer	A two-byte buffer for temporarily holding data read from memory or I/O device by the BIU or data written to memory or I/O device by the CPU.

FUNCTIONAL DESCRIPTION

2.2 Internal bus interface

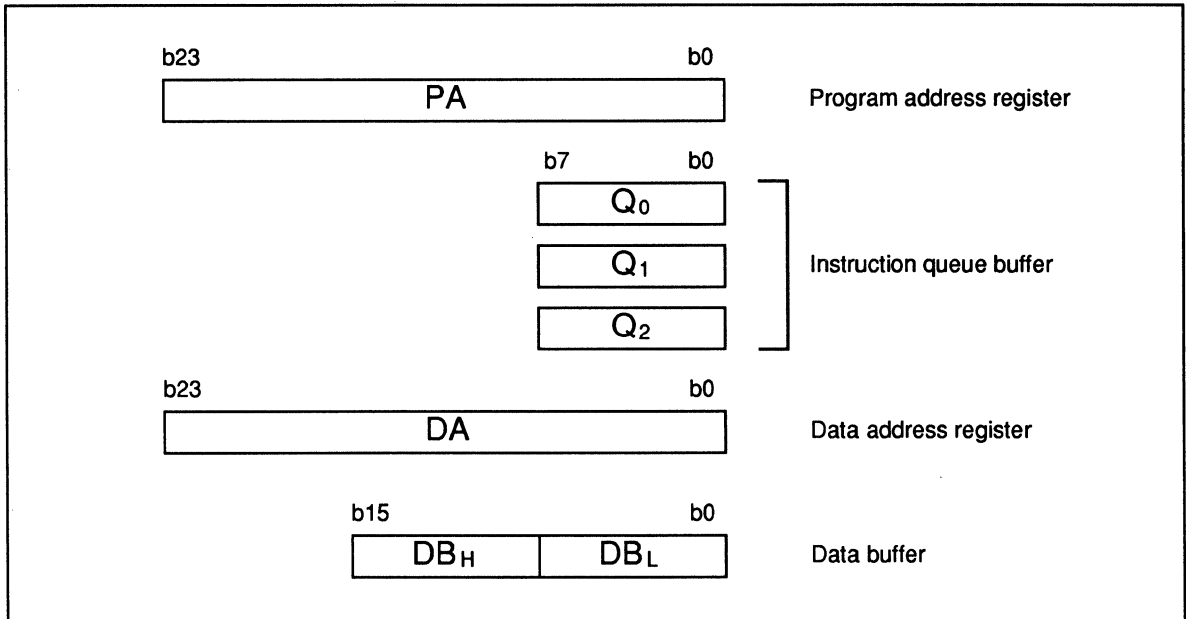


Fig. 2.2.2 BIU registers

The BIU performs the following operations.

1. **Prefetches an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer.**

Normally, a program is executed sequentially in ascending order of address. Therefore, if the next instruction code is prefetched in the instruction queue buffer, the CPU can execute instructions simply by obtaining the instruction code from the instruction queue buffer. This will eliminate the time needed by the CPU to access the memory.

When the CPU is not using the bus (for example when performing register to register operation), the BIU reads an instruction code from the program memory (area where the program is stored) and stores it in the instruction queue buffer. Data up to three bytes can be prefetched because the instruction queue buffer is three bytes long. Refer to section "2.2.4 Data read/write operations" for more information concerning instruction code prefetch.

2. **Reads data at the specified address into the BIU when the CPU requests data in memory and transfers it to the CPU.**

When executing instructions that process data in memory or I/O device, the CPU must access the address assigned to the memory or I/O device and read the data. Because the operating clock of the CPU and bus are different, the CPU reads the data through the data buffer of the BIU.

3. **Writes the data obtained from the CPU to the specified address in memory.**

When writing data to a specific address, the CPU sends the address and data to the BIU. And after that, the CPU continues to execute the next instruction extracting from the instruction queue buffer, because actual writing to memory or I/O device is performed by the BIU.

4. **Controls read of word data from odd number address and outputs the control signals required to access external memory in byte unit.**

The transfer of data between the CPU and BIU is always performed through a 24-bit address bus and 16-bit data bus. This is also true between the BIU and internal area or I/O device. The BYTE pin (external bus width selection input pin) determines these data widths, in addition, the wait bit determines the access timing only when an external memory is accessed.

FUNCTIONAL DESCRIPTION

2.2 Internal bus interface

2.2.3 Bus Interface unit operations

Figure 2.2.3 shows the operating waveforms of the bus interface unit. The M37732 group BIU always operates at one of the waveforms shown in Figure 2.2.3.

The meaning of signals ALE and \bar{E} in Figure 2.2.3 are as follows:

● **ALE** (Address Latch Enable)

Signal used to latch only address signals from multiplex signals containing data and address.

● **\bar{E}**

Signal set to "L" level when the bus interface unit reads instruction code or data from memory or when it writes data to memory. Table 2.2.2 shows the data bus state according to \bar{E} and R/W signals.

Table 2.2.2 Data bus state according to \bar{E} and R/W

\bar{E}	R/W	Data bus state
H	H	Not used
H	L	Not used
L	H	Read
L	L	Write

(1) Basic operation

Waveform (a) is the bus interface operating waveform under the following conditions:

- When a one byte internal area and external memory is accessed.
- When two bytes in internal area are accessed together (starting on an even address).
- When two bytes in external memory are accessed together (starting on an even address when the BYTE pin is at "L" level).
- When the instruction code is obtained from memory into the instruction queue buffer.

Waveform (b) is the bus interface operating waveform when accessing in byte unit under the following conditions:

- When two bytes in internal area and external memory are accessed together (starting on an odd address).
- When two bytes in external memory are accessed together with the BYTE pin at "H" level.

Waveforms (a) and (b) are the basic operating waveforms of the BIU. Waveform (a) or (b) is always used when accessing the internal area.

When accessing the external memory area, the BIU operating waveform changes according to the wait bit.

(2) Effect of the wait bit

With the M37732 group, the external memory access time can be 1.5 times as long (the "L" level width of \bar{E} signal becomes twice) by clearing the wait bit (bit 2) to "0" in the processor mode register (address 005E₁₆). This enables external expansion of slow memories and peripheral LSIs.

Note: Internal memory access is not affected by the wait bit.

Figure 2.2.3 (c) to (f) show the effect of the wait bit on waveforms (a) and (b). Waveform (c) is the waveform when an external memory area is accessed under the conditions for waveform (a) with the wait bit cleared to "0".

Waveforms (d) to (f) are the waveforms when an external memory area is accessed under the conditions for waveform (b) with the wait bit cleared to "0". The entire waveform is affected by the wait bit for waveform (d) and the first half or the last half is affected respectively for waveforms (e) and (f).

FUNCTIONAL DESCRIPTION

2.2 Internal bus interface

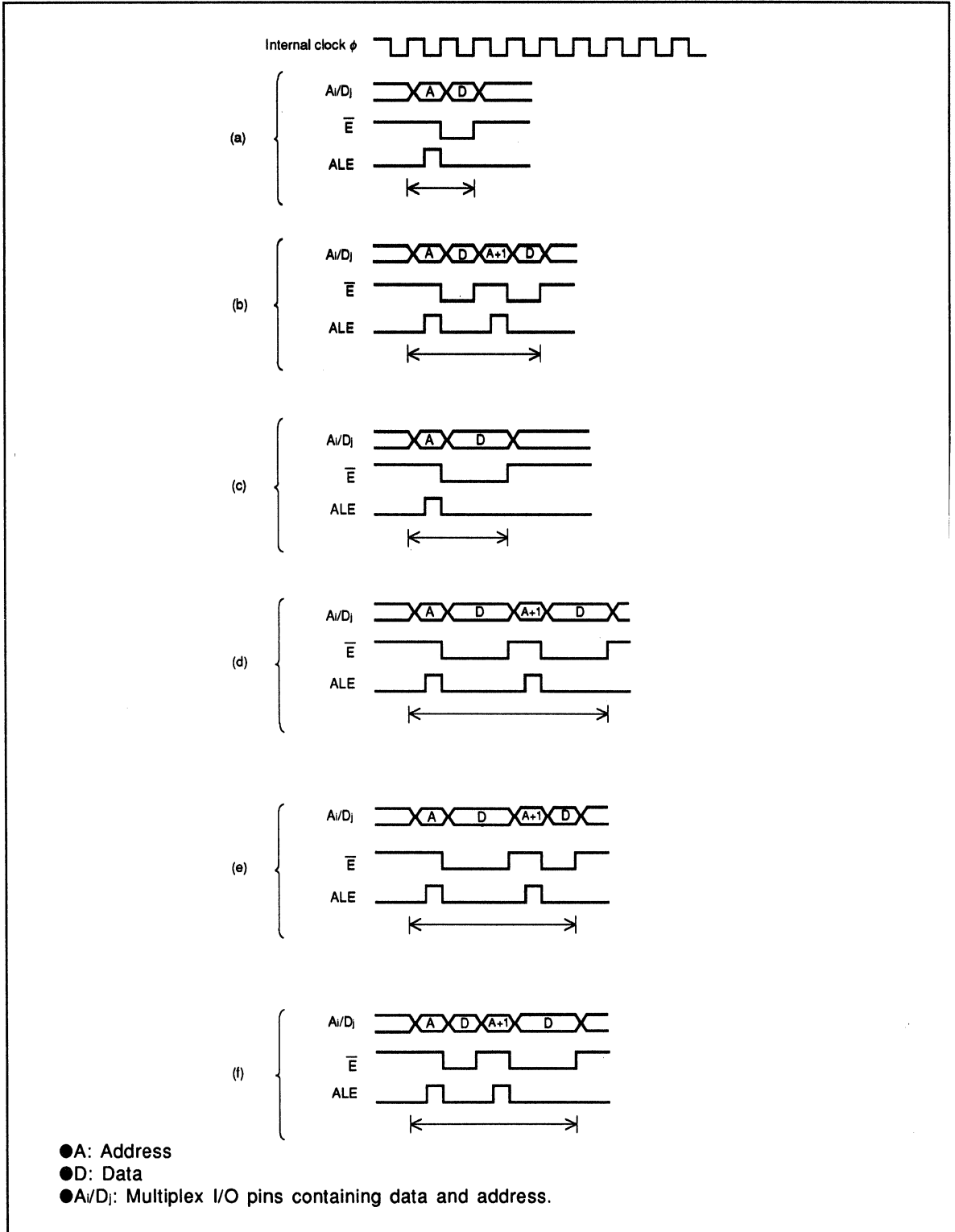


Fig. 2.2.3 Bus Interface unit operating waveform

FUNCTIONAL DESCRIPTION

2.2 Internal bus interface

2.2.4 Data read/write operations

(1) Instruction code read

The CPU reads instruction codes from the instruction queue buffer of the BIU and executes them. The CPU notifies the BIU that an instruction code is needed during the instruction code fetch cycle. At this point, the operation depends on whether the instruction queue buffer contains an instruction code or not. If there is an instruction code in the instruction queue buffer, it is passed to the CPU. If there is no instruction code in the instruction queue buffer, or if the amount of data in the instruction queue buffer is less than the necessary instruction code, the BIU halts the CPU until a sufficient amount of instruction codes is stored in the instruction queue buffer.

Even when there is no request for instruction code from the CPU, if the instruction queue buffer is empty or if there is only one instruction code and the bus is available at the next cycle (the CPU does not use the bus at the next cycle), the BIU reads instruction codes from memory and stores them in the instruction queue buffer (instruction prefetch). During instruction prefetch, if the first address accessed when reading an instruction code from memory is even, then the data at the next odd number address is also read and stored in the instruction queue buffer. If the first accessed address is odd, only one byte is read and stored in the instruction queue buffer. However, if the instruction code is read from external memory with the BYTE pin at "H" level (8-bit external bus width), only one byte is read regardless of the accessed address.

Instruction code read is performed with operation (a) or (c) shown in Figure 2.2.3. When a branch or a jump or subroutine call instruction or an interrupt is executed, the contents of the instruction queue buffer are cleared and a new instruction code is read from the new address.

(2) Data read/write

The CPU reads and writes data from/to the BIU data buffer. The CPU issues a request to BIU when it attempts to read or write data. At this point, if the BIU is using the bus or if there is a higher priority request, the CPU is made to wait until the BIU becomes ready. When the bus is available for data read or write, the BIU operates at one of the waveforms (a) to (f) shown in Figure 2.2.3.

●Data read

When the CPU requests data from the BIU, it waits until the data becomes complete data in the data buffer. The BIU sends the address received from the CPU to the address bus, reads the contents of memory when \bar{E} signal is "L" level, and stores it in the data buffer.

●Data write

The CPU sends address (address in which the data is written) and data to BIU.

The address is written in the BIU data address register and the data is written in the data buffer. The actual writing in memory is performed by BIU and the CPU can proceed to the next step without waiting for the BIU to complete writing data in memory. The BIU sends the address received from the CPU to the address bus, sends the contents of the data buffer to the data bus, and writes it in memory when \bar{E} signal is "L" level.

FUNCTIONAL DESCRIPTION

2.3 Addressable memory space

2.3 Addressable memory space

The M37732 group allocates internal RAM, external memory, I/O, and various control registers in the same memory space. Therefore, data transfer and operation can be performed with the same instruction without distinguishing memory and I/O area.

Program counter (PC) consists of 16 bits. It is used together with an 8-bit program bank register (PG) to directly access a 16M-byte address space from 0_{16} to $FFFFFF_{16}$.

Figure 2.3.1 shows the addressable memory space.

2.3.1 Banks

Address space is divided into 64K-byte blocks called banks. The M37732 group can access 256 banks from bank 0_{16} to bank FF_{16} (255).

The high-order 8 bits of the 24-bit address indicate the bank and the contents of the program bank register (PG) or the data bank register (DT) indicate the bank to be used.

If the program counter overflows at a bank boundary, the contents of the program bank register are incremented by 1. If a borrow occurs in the program counter, the contents of the program bank register are decremented by 1. Therefore, programs can be written without considering the bank boundaries, usually. The banks can be accessed efficiently by using an addressing mode that uses the data bank register.

SFR area in bank 0_{16} (addresses 0_{16} to $FFFF_{16}$) contains timers, interrupts, and internal I/O control registers. Addresses $FFD6_{16}$ to $FFFF_{16}$ are allocated to an interrupt vector table containing the start address of interrupt service routines. Interrupt vector table area must be allocated in ROM.

2.3.2 Direct page

By using the direct page register (DPR), bank 0_{16} or a 256-byte space spanning across bank 0_{16} and bank 1_{16} can be accessed with fewer instruction cycles by using direct page addressing mode. This area is referred to as the direct page and is normally used for frequently accessed information.

The direct page area can be specified by setting the lowermost address of the required area in the direct page register (refer to section "2.1.8 Direct page register").

FUNCTIONAL DESCRIPTION

2.3 Addressable memory space

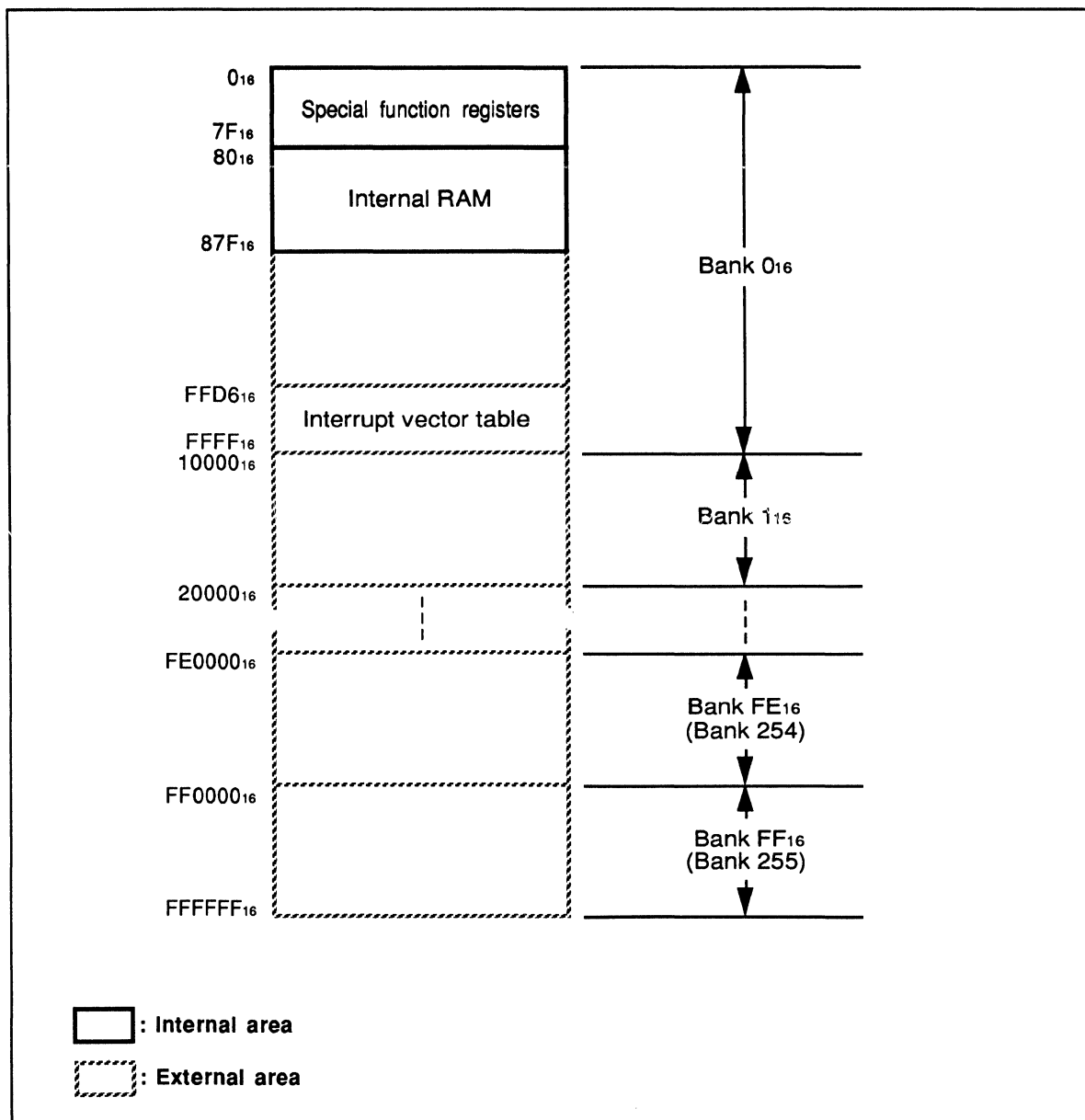


Fig. 2.3.1 Addressable memory space

FUNCTIONAL DESCRIPTION

2.4 Memory allocation

2.4 Memory allocation

SFR (Special Function Registers) and internal RAM are allocated in internal area of bank 0₁₆. ROM must be connected to the external memory area because the M37732 group is an external ROM version.

Figure 2.4.1 shows the memory map.

2.4.1 SFR (Special Function Registers), Internal RAM and external memory area

The allocated SFR, internal RAM and external memory area are described below.

(1) SFR area

Addresses 0000₁₆ to 007F₁₆ of bank 0₁₆ are the SFR (Special Function Registers) area. This area contains the control registers of internal peripheral devices, I/O ports, timers, and so on. Internal peripheral devices can be accessed through these registers. Figure 2.4.2 shows the memory map of the SFR area.

Each bit in the register can be either read-only, write-only, or read/write bit.

Refer to each block description for the register function in the SFR area.

Refer to section "3.1.2 Internal registers state at reset" for the state of the SFR at reset.

(2) Internal RAM area

A 2048-byte static RAM is allocated at addresses 0080₁₆ to 087F₁₆ of bank 0₁₆. In addition to storing data, the internal RAM area is used as stack area during subroutine calls and interrupts. Therefore, be careful of subroutine nesting levels and multiple interrupt levels so that important data is not destroyed.

(3) External memory area

External memory can be set to a 16M-byte space of banks 0₁₆ to FF₁₆ excluding SFR area (addresses 0000₁₆ to 007F₁₆) and internal RAM area (addresses 0080₁₆ to 087F₁₆).

The interrupt vector table allocated to addresses FFD6₁₆ to FFFF₁₆ of bank 0₁₆ must be set in ROM. This area contains the start address of the interrupt service routine for each interrupt source and reset.

FUNCTIONAL DESCRIPTION

2.4 Memory allocation

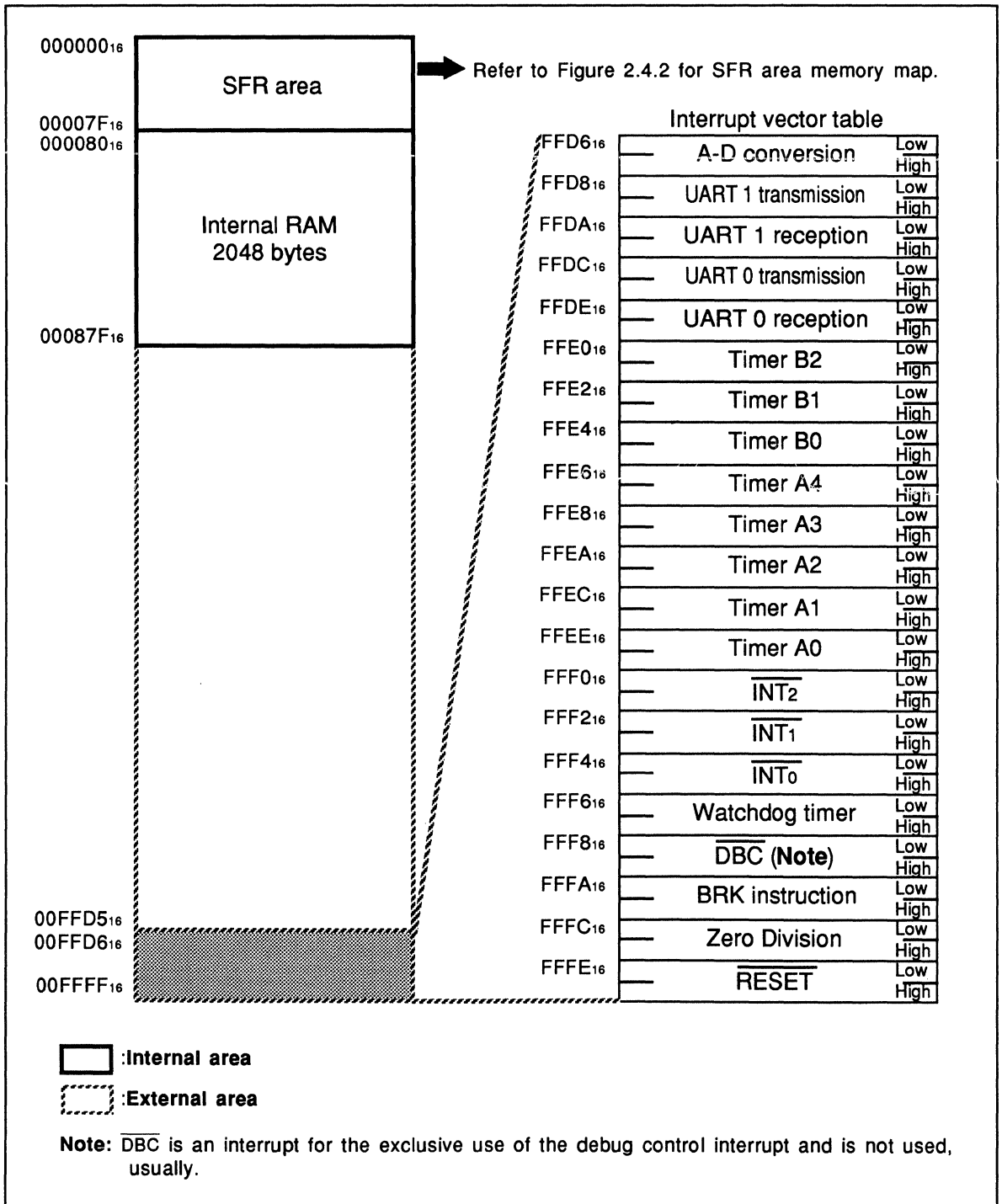


Fig. 2.4.1 Memory map

FUNCTIONAL DESCRIPTION

2.4 Memory allocation

Address (Hexadecimal notation)		Address (Hexadecimal notation)	
000000		000040	Count start flag
000001		000041	
000002		000042	One-shot start flag
000003		000043	
000004		000044	Up-down flag
000005		000045	
000006		000046	Timer A0 register
000007		000047	
000008		000048	Timer A1 register
000009		000049	
00000A	Port P4 register	00004A	Timer A2 register
00000B	Port P5 register	00004B	
00000C	Port P4 direction register	00004C	Timer A3 register
00000D	Port P5 direction register	00004D	
00000E	Port P6 register	00004E	Timer A4 register
00000F	Port P7 register	00004F	
000010	Port P6 direction register	000050	Timer B0 register
000011	Port P7 direction register	000051	
000012	Port P8 register	000052	Timer B1 register
000013		000053	
000014	Port P8 direction register	000054	Timer B2 register
000015		000055	
000016		000056	Timer A0 mode register
000017		000057	Timer A1 mode register
000018		000058	Timer A2 mode register
000019		000059	Timer A3 mode register
00001A		00005A	Timer A4 mode register
00001B		00005B	Timer B0 mode register
00001C		00005C	Timer B1 mode register
00001D		00005D	Timer B2 mode register
00001E	A-D control register	00005E	Processor mode register
00001F	A-D sweep pin selection register	00005F	
000020	A-D register 0	000060	Watchdog timer
000021		000061	Watchdog timer frequency selection flag
000022	A-D register 1	000062	Waveform output mode register
000023		000063	
000024	A-D register 2	000064	Pulse output data register1
000025		000065	Pulse output data register0
000026	A-D register 3	000066	
000027		000067	
000028	A-D register 4	000068	
000029		000069	
00002A	A-D register 5	00006A	
00002B		00006B	
00002C	A-D register 6	00006C	
00002D		00006D	
00002E	A-D register 7	00006E	
00002F		00006F	
000030	UART 0 transmit/receive mode register	000070	A-D conversion interrupt control register
000031	UART 0 baud rate register	000071	UART 0 transmission interrupt control register
000032	UART 0 transmission buffer register	000072	UART 0 receive interrupt control register
000033		000073	UART 1 transmission interrupt control register
000034	UART 0 transmit/receive control register 0	000074	UART 1 receive interrupt control register
000035	UART 0 transmit/receive control register 1	000075	Timer A0 interrupt control register
000036	UART 0 receive buffer register	000076	Timer A1 interrupt control register
000037		000077	Timer A2 interrupt control register
000038	UART 1 transmit/receive mode register	000078	Timer A3 interrupt control register
000039	UART 1 baud rate register	000079	Timer A4 interrupt control register
00003A	UART 1 transmission buffer register	00007A	Timer B0 interrupt control register
00003B		00007B	Timer B1 interrupt control register
00003C	UART 1 transmit/receive control register 0	00007C	Timer B2 interrupt control register
00003D	UART 1 transmit/receive control register 1	00007D	TINT0 interrupt control register
00003E	UART 1 receive buffer register	00007E	TINT1 interrupt control register
00003F		00007F	TINT2 interrupt control register

Fig. 2.4.2 SFR area memory map

FUNCTIONAL DESCRIPTION

2.4 Memory allocation

2.4.2 Processor mode

M37732 group can operate in microprocessor mode. ROM, RAM and peripherals can be expanded to any area (except for SFR and internal RAM area) within a 16M-byte addressable memory space.

Figure 2.4.3 shows the structure of the processor mode register. Refer to section "3.3 Software reset" for software reset bit and section "2.6.6 Interrupt priority level detection time" for interrupt priority detection time selection bits. Wait bit is described next section "2.4.3 External memory area bus control".

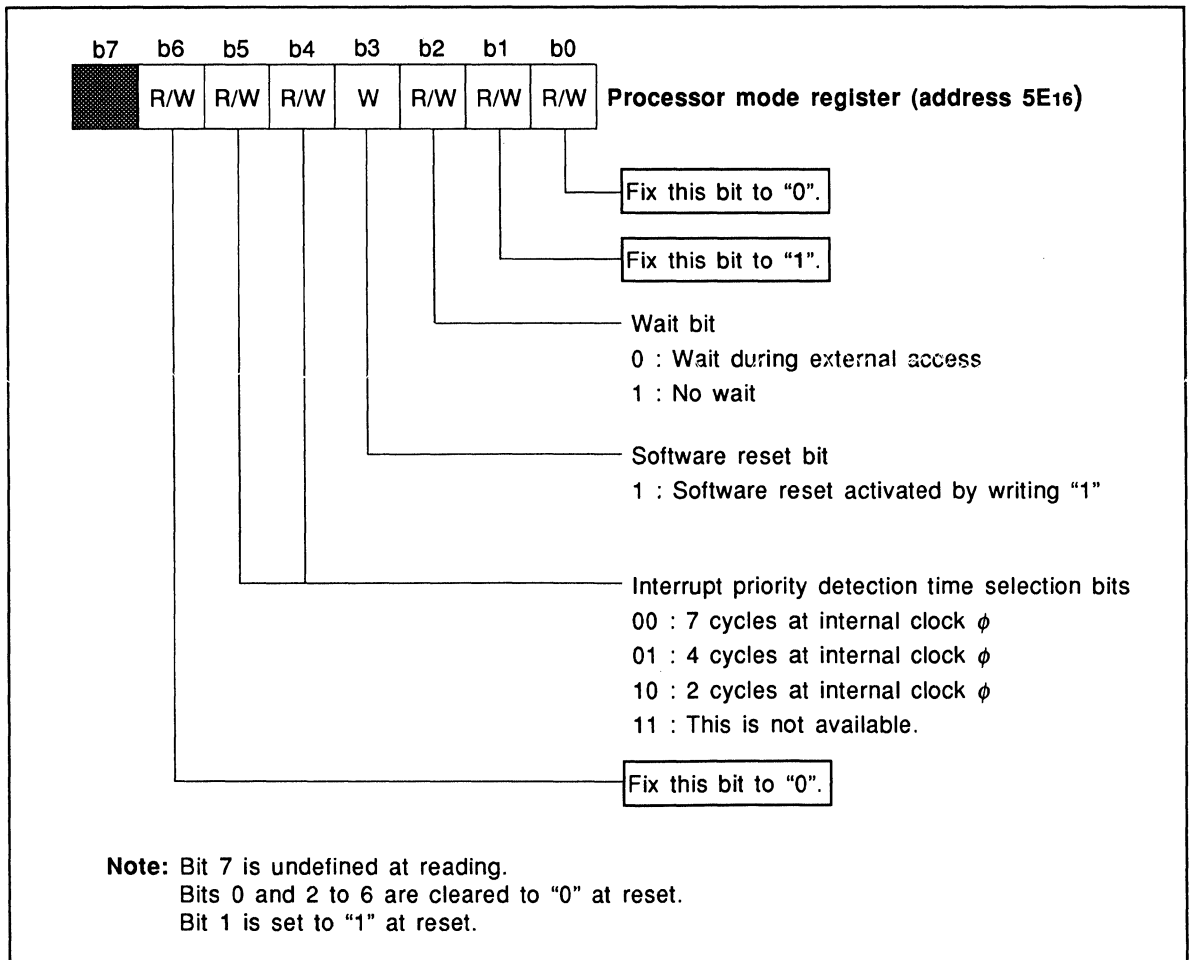


Fig. 2.4.3 Processor mode register structure

FUNCTIONAL DESCRIPTION

2.4 Memory allocation

2.4.3 External memory area bus control

The BYTE pin and the wait bit provided to simplify access to external memory area are described below. The BYTE pin and the wait bit are valid only when accessing external memory area and have no effect when accessing internal RAM or SFR.

If external memories are connected to the area overlapping with internal area, the data in internal area is read into the CPU when reading. In this case the data in external memory is not read into the CPU. When writing data, the data is written to both internal and external area.

(1) BYTE pin (external bus width selection pin)

When accessing the external memory, the input level to the BYTE pin is used to select whether 8-bit data bus or 16-bit data bus (refer to section "2.5.3 (2) Data bus").

The external bus width becomes 16 bits when the BYTE pin is at "L" level. In this case, data read/write to the external area is always performed in 16-bit (1-word) unit. And the pins $A_{16}/D_0-A_{23}/D_7$ become the data I/O pins for the low-order byte (even address data: D_0-D_7) of a 16-bit data and the pins $A_8/D_8-A_{15}/D_{15}$ become the data I/O pins for the high-order byte (odd address data: D_8-D_{15}) of a 16-bit data.

The external bus width becomes 8 bits when the BYTE pin is at "H" level. In this case, data read/write to the external area is always performed in 8-bit (1-byte) unit. And the pins $A_{16}/D_0-A_{23}/D_7$ become the data (D_0 to D_7) I/O pins. The use of 8-bit peripheral ICs is simplified by setting the bus width for external area to 8-bit.

The data width is always 16 bits when accessing the internal memory area regardless of the BYTE pin level.

(2) Wait bit

The wait bit (bit 2 at address $5E_{16}$) is used to simplify access to external memory or I/O. When the wait bit is "0", a wait for external area access is enabled (one-wait mode) and bus operation is performed at the speed of 2/3 of the bus cycle (bus cycle= $f(X_{IN})/4$) during no wait. When the wait bit is "1", bus operation becomes no wait mode and bus cycle is $f(X_{IN})/4$.

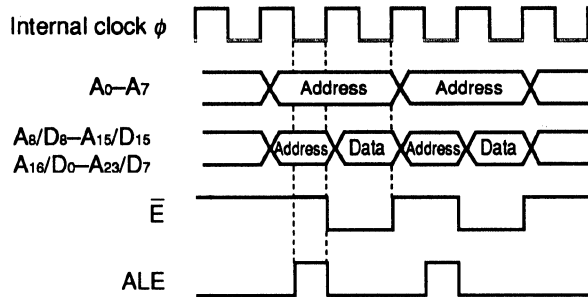
The wait bit is cleared to "0" at reset and the system starts in one-wait mode. Internal area access is always performed at no wait because this bit is ignored.

Figure 2.4.4 shows the effect of the wait bit for external area access.

FUNCTIONAL DESCRIPTION

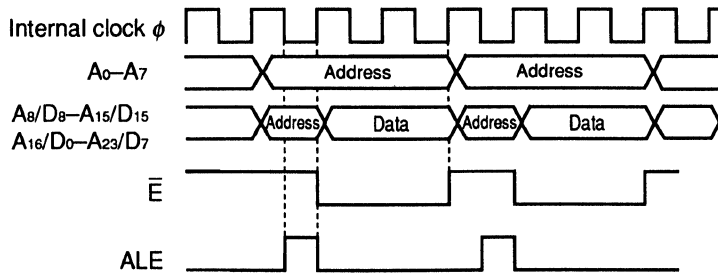
2.4 Memory allocation

(a) Waveform when the external memory area is accessed with the wait bit is "1".



※ This waveform is always used when accessing the internal area.

(b) Waveform when the external memory area is accessed with the wait bit is "0".



※ Waveform (a) is used for internal area access even when the wait bit is "0".

Fig. 2.4.4 Effect of wait bit for external area access (when BYTE pin="L" level)

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

2.5 Input/Output pins

The M37732 group has 37 programmable I/O pins (ports P4–P8). Ports P5–P8 also function as I/O pins for internal peripheral devices.

Programmable I/O ports and the pin functions are described below.

2.5.1 Programmable I/O ports

Each of the programmable I/O ports (ports P4–P8) has a direction register and a port register. The direction registers and the port registers are allocated in the SFR area of bank 0₁₆. The direction register is used to select the input/output mode by one bit. Input level of port is read from the pin selecting input mode by reading the port register. The data written to the port register is output from the pin selecting output mode. Figure 2.5.1 shows the memory allocation of the direction registers and the port registers.

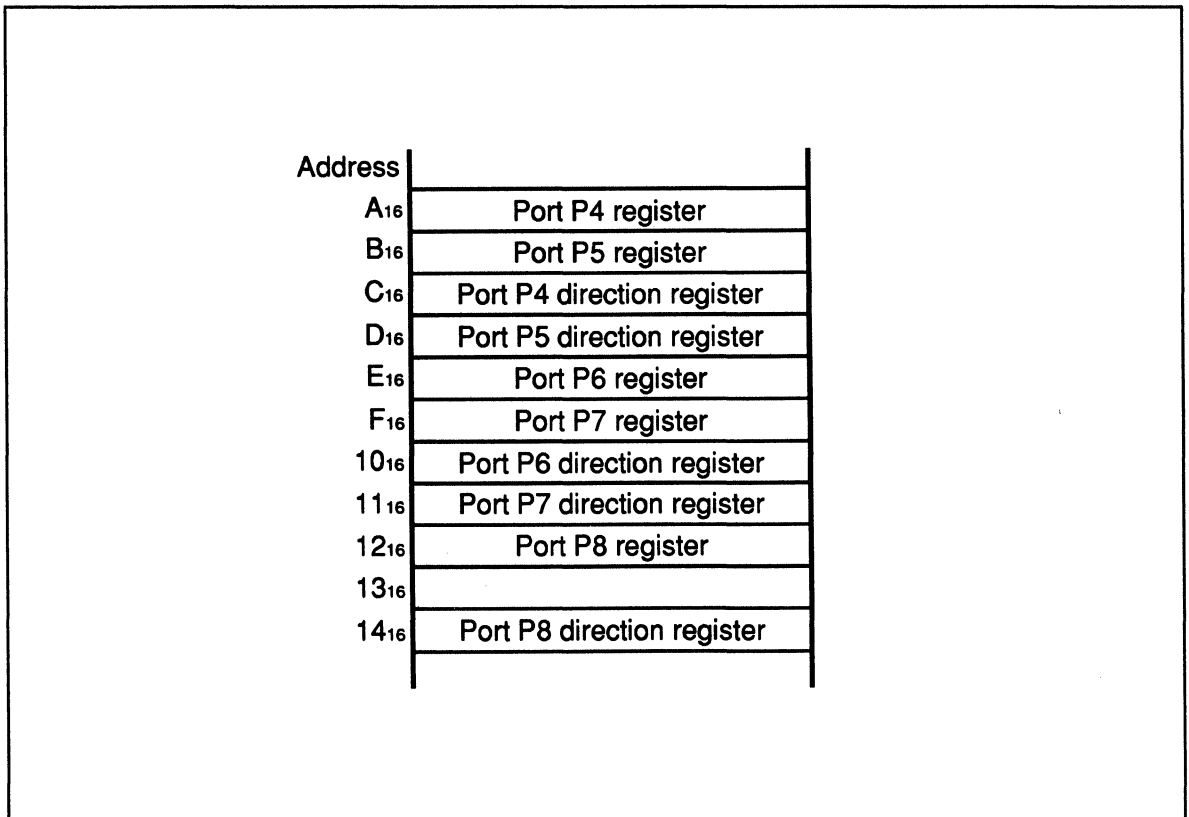


Fig. 2.5.1 Memory allocation of direction registers and port registers

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

(1) Direction register

Direction of each port can be selected by setting the direction registers. Each bit of the direction register corresponds to a pin. Figure 2.5.2 shows the structure of the port P_i (i=4 to 8) direction registers.

The port is set to input mode (input pin) when the corresponding bit is "0", and to output mode (output pin) when the corresponding bit is "1".

The direction registers are cleared to "00₁₆" at reset. Therefore, I/O ports are set to input mode.

If I/O ports are not used as output pins, set the corresponding direction register bit to "0" for input mode.

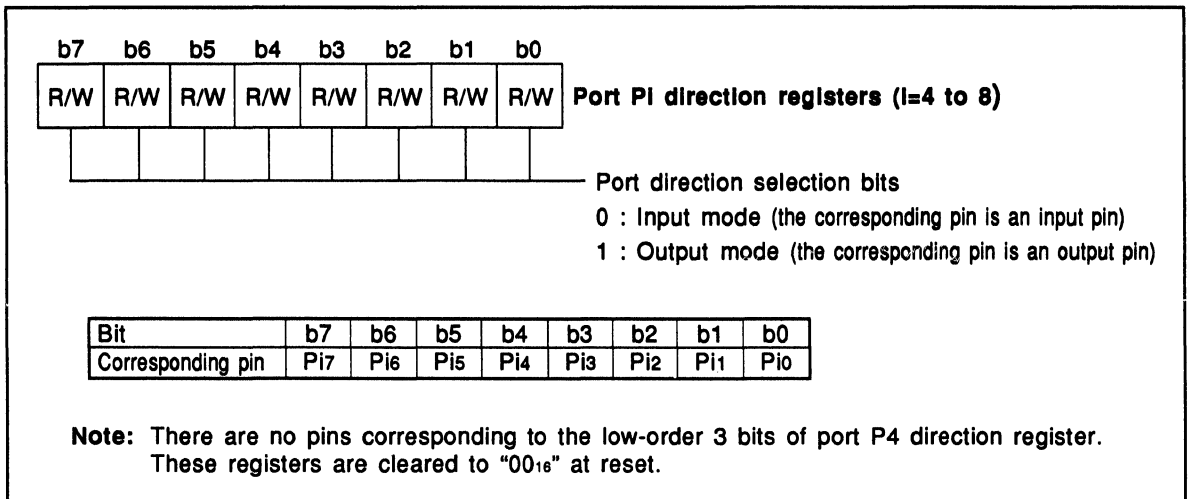


Fig. 2.5.2 Port P_i (i=4 to 8) direction registers structure

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

(2) Port register

The port register is used to transfer data with external devices through the I/O ports. Figure 2.5.3 shows the relationship between the port registers and the pins.

To output data from a port set to output mode, the data must be written to the corresponding bits of the port register. This data is written in the port latch and is output from the port set to output mode. If a bit of the port register corresponding to a pin programmed for output is read, the state of the output pin is not read but the contents of the port latch are read. Therefore, the previously output value can be read correctly even if the output "H" voltage drops or "L" voltage rises due to external load (refer to "Figure 2.5.4 Port peripheral circuits").

A port programmed for input is floated and the value input to the pin can be read by reading the corresponding bit of the port register. If a value is written to a bit of the port register corresponding to a port programmed for input, it is written in the port latch and the pin remains floating.

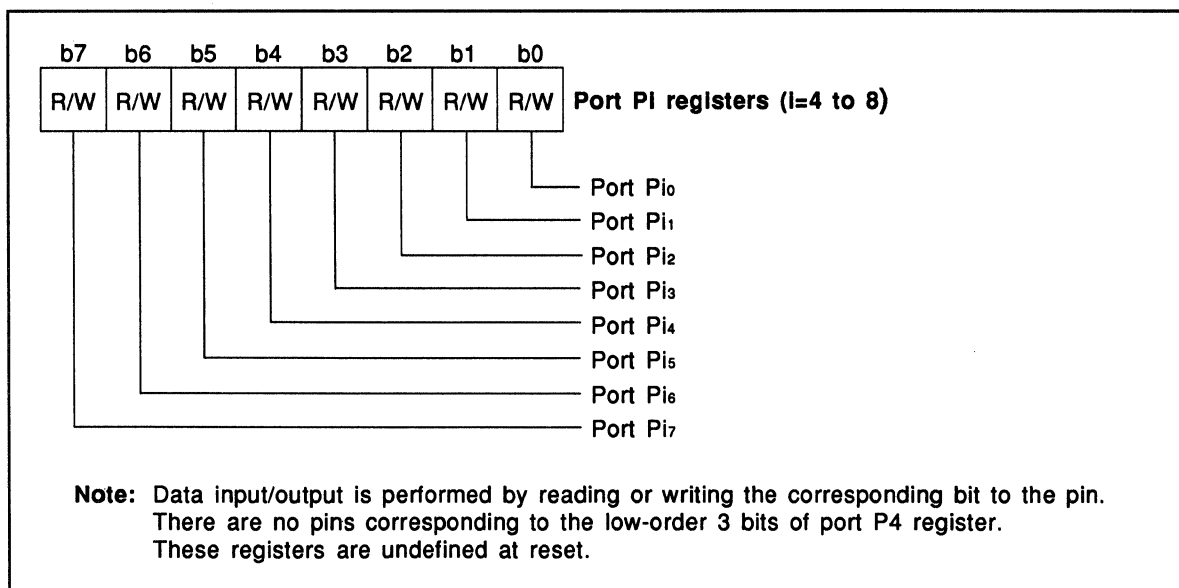


Fig. 2.5.3 Relationship between port registers and pins

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

2.5.2 Ports P4–P8 functions

Ports P5–P8 (32 pins) have the special functions such as I/O pins for external interrupt, timer, A-D converter and serial I/O as well as programmable I/O port function. Port P4 functions as only a programmable I/O port. Refer to the section on each function for ports P5–P8 function as I/O pins for peripheral devices. Ports P4–P8 function as programmable I/O ports after removing reset.

Figure 2.5.4 shows the port peripheral circuits. Table 2.5.1 shows the function of ports P4–P8. Refer to section "2.5.1 Programmable I/O ports" for information concerning functions of the programmable I/O ports. Ports P5–P8 are forced to output mode when used as output pin for internal peripheral devices, but when used as input pin for internal peripheral devices, the direction register must be set to input mode because the contents of the port direction register has higher priority.

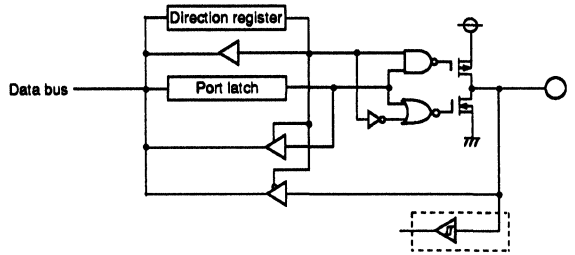
Table 2.5.1 Function of ports P4–P8

Pin	Function 1	Function 2
Port P4 (P4 ₃ –P4 ₇)	Programmable I/O port	
Port P5 (P5 ₀ –P5 ₇)	Programmable I/O port	TA0 _{OUT} /RTP0 ₀ (P5 ₀):Timer A0 I/O pin / RTP0 ₀ output pin TA0 _{IN} /RTP0 ₁ (P5 ₁) :Timer A0 input pin / RTP0 ₁ output pin TA1 _{OUT} /RTP0 ₂ (P5 ₂):Timer A1 I/O pin / RTP0 ₂ output pin TA1 _{IN} /RTP0 ₃ (P5 ₃) :Timer A1 input pin / RTP0 ₃ output pin * RTP0 _n (n=0 to 3) output pin is available when RTP0 function is selected. TA2 _{OUT} /RTP1 ₀ (P5 ₄):Timer A2 I/O pin / RTP1 ₀ output pin TA2 _{IN} /RTP1 ₁ (P5 ₅) :Timer A2 input pin / RTP1 ₁ output pin TA3 _{OUT} /RTP1 ₂ (P5 ₆):Timer A3 I/O pin / RTP1 ₂ output pin TA3 _{IN} /RTP1 ₃ (P5 ₇) :Timer A3 input pin / RTP1 ₃ output pin * RTP1 _n (n=0 to 3) output pin is available when RTP1 function is selected.
Port P6 (P6 ₀ –P6 ₇)	Programmable I/O port	TA4 _{OUT} (P6 ₀) :Timer A4 I/O pin TA4 _{IN} (P6 ₁) :Timer A4 input pin INT ₀ (P6 ₂) :INT ₀ interrupt signal input pin INT ₁ (P6 ₃) :INT ₁ interrupt signal input pin INT ₂ (P6 ₄) :INT ₂ interrupt signal input pin TB0 _{IN} (P6 ₅) :Timer B0 input pin TB1 _{IN} (P6 ₆) :Timer B1 input pin TB2 _{IN} (P6 ₇) :Timer B2 input pin
Port P7 (P7 ₀ –P7 ₇)	Programmable I/O port	AN ₀ –AN ₆ (P7 ₀ –P7 ₆):Analog input pin AN ₇ /AD _{TRG} (P7 ₇) :Analog input pin when AN ₇ function is selected External trigger input pin when AD _{TRG} function is selected
Port P8 (P8 ₀ –P8 ₇)	Programmable I/O port	CTS ₀ /RTS ₀ (P8 ₀) :UART0 transmit enable signal input pin when CTS ₀ function is selected UART0 receive enable signal output pin when RTS ₀ function is selected CLK ₀ (P8 ₁) :UART0 transmit/receive clock I/O pin RxDo (P8 ₂) :UART0 receive data input pin TxDo (P8 ₃) :UART0 transmit data output pin CTS ₁ /RTS ₁ (P8 ₄) :UART1 transmit enable signal input pin when CTS ₁ function is selected UART1 receive enable signal output pin when RTS ₁ function is selected CLK ₁ (P8 ₅) :UART1 transmit/receive clock I/O pin RxD ₁ (P8 ₆) :UART1 receive data input pin TxD ₁ (P8 ₇) :UART1 transmit data output pin

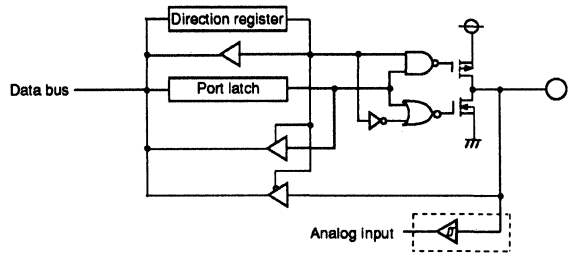
FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

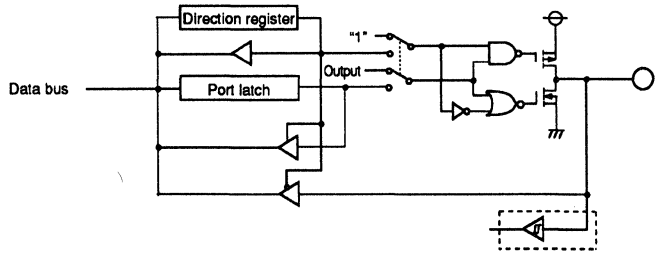
Ports P4₃–P4₆
(dotted area not included)



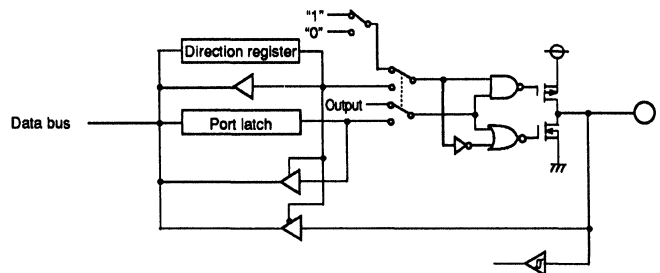
Ports P4₇, P6₁–P6₇, P8₂, P8₆
(dotted area included, however no hysteresis for P8₂ and P8₆)



Ports P7₀–P7₆ (dotted area not included)
Port P7₇ (dotted area included)



Ports P8₃, P8₇
(dotted area not included)
Ports P5₀–P5₇, P6₀
(dotted area included)



Ports P8₀, P8₁, P8₄, P8₅

\bar{E} output pin

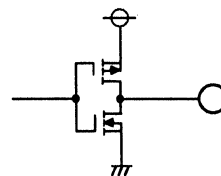


Fig. 2.5.4 Port peripheral circuits

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

2.5.3 Pin functions

The functions of each pin are described below.

(1) Address bus (pins A₀–A₇, pins A₈/D₈–A₁₅/D₁₅, pins A₁₆/D₀–A₂₃/D₇)

Pins A₀–A₇ output the low-order 8 bits of the address signal.

Pins A₈/D₈–A₁₅/D₁₅ output the middle-order 8 bits of the address signal. Pins A₁₆/D₀–A₂₃/D₇ output the high-order 8 bits of the address signal.

Pins A₈/D₈–A₁₅/D₁₅ and pins A₁₆/D₀–A₂₃/D₇ function as data I/O pins as the same time. Therefore, pins A₈/D₈–A₁₅/D₁₅ and pins A₁₆/D₀–A₂₃/D₇ perform time division multiplexing of address output and data input/output.

The M37732 group allows direct access to 16M-byte space from addresses 000000₁₆ to FFFFFFF₁₆. Therefore, 24 address signals are output externally.

Figure 2.5.5 shows the relationship between the address bus and the output pins.

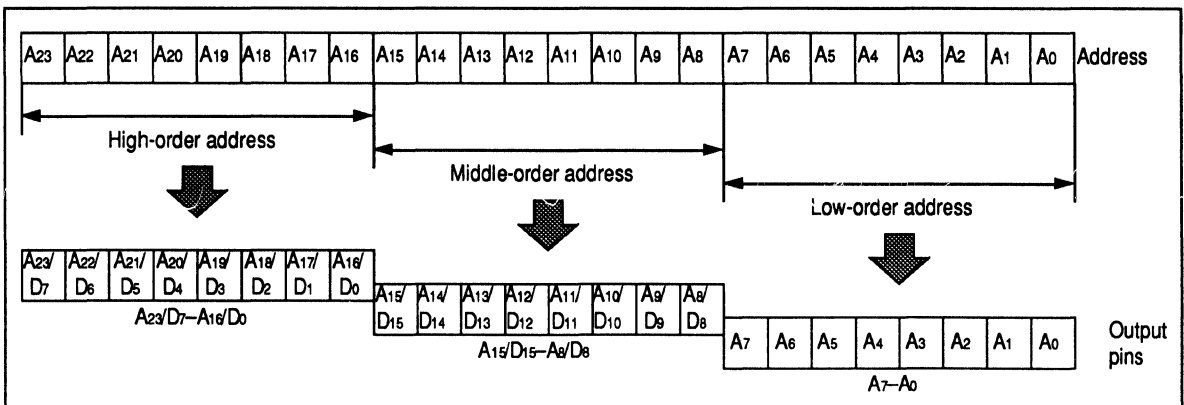


Fig. 2.5.5 Relationship between address bus and output pins

(2) Data bus (pins A₈/D₈–A₁₅/D₁₅, pins A₁₆/D₀–A₂₃/D₇)

In addition to address signal (high-order and middle-order address bus) output pin function, pins A₈/D₈–A₁₅/D₁₅ and pins A₁₆/D₀–A₂₃/D₇ also function as data I/O pins. The level of the BYTE pin can be used to select between 8-bit and 16-bit data bus width.

●When the BYTE pin is at "L" level (16-bit external bus width)

When the BYTE pin is at "L" level, the external bus width is 16 bits and even address data and odd address data are output simultaneously. Pins A₈/D₈–A₁₅/D₁₅ and pins A₁₆/D₀–A₂₃/D₇ are used as address bus and data bus, and multiplex signals (address signal and data signal) are output from these pins.

Pins A₈/D₈–A₁₅/D₁₅ perform time division multiplexing of address (A₈–A₁₅) output and data input/output in odd address (high-order byte of 16-bit data). Middle-order address is output while \bar{E} signal is at "H" level, and data input/output in odd address is performed while \bar{E} signal is at "L" level.

Similarly, pins A₁₆/D₀–A₂₃/D₇ perform time division multiplexing of address (A₁₆–A₂₃) output and data input/output in even address (low-order byte of 16-bit data). High-order address is output while \bar{E} signal is at "H" level, and data input/output in even address is performed while \bar{E} signal is at "L" level. Figure 2.5.6 shows the relationship between the data bus and the I/O pins (when BYTE pin="L" level). Figure 2.5.7 shows the bus timing when external bus width is 16 bits.

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

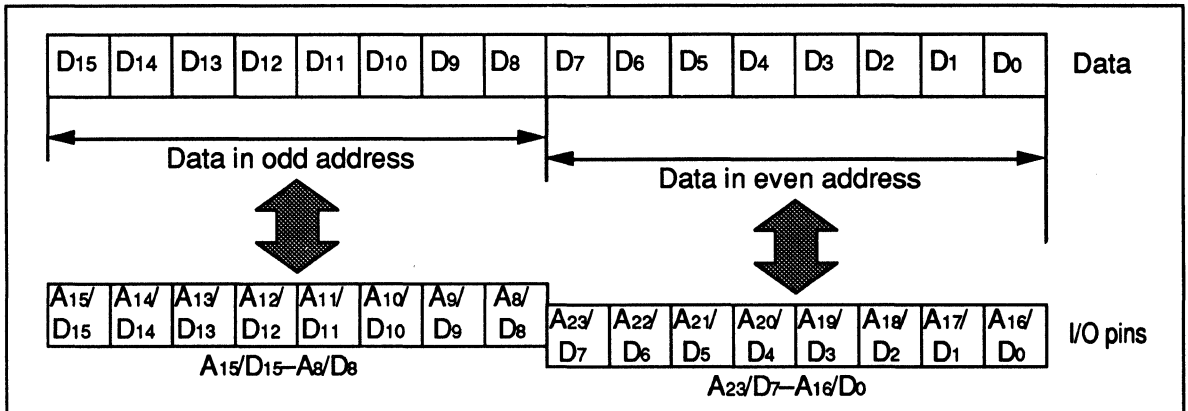


Fig. 2.5.6 Relationship between data bus and I/O pins (when BYTE pin="L" level)

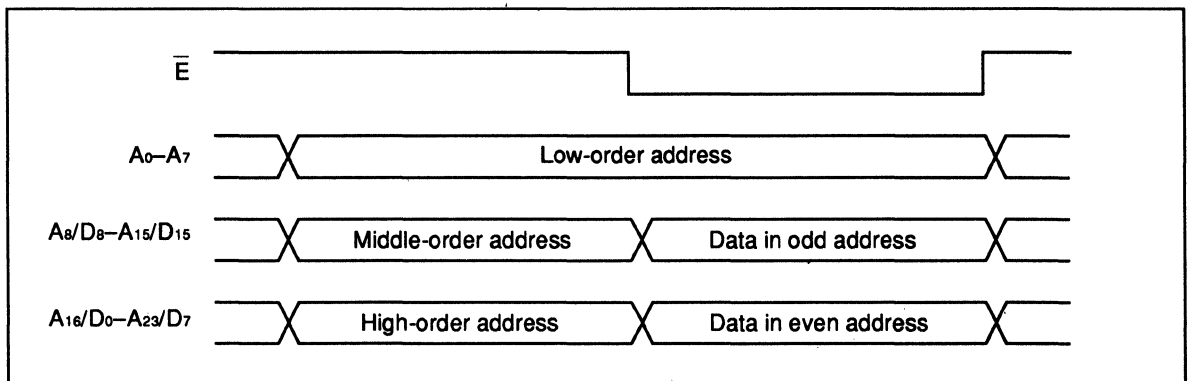


Fig. 2.5.7 Bus timing when external bus width is 16 bits (when BYTE pin="L" level)

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

●When the BYTE pin is at "H" level (8-bit external bus width)

When the BYTE pin is at "H" level, the external bus width is 8 bits and pins A₁₆/D₀–A₂₃/D₇ perform time division multiplexing of address (A₁₆–A₂₃) output and data input/output. Address is output while \bar{E} signal is at "H" level, and 8-bit data is input/output when \bar{E} signal is at "L" level.

Figure 2.5.8 shows the relationship between the data bus and the I/O pins (when BYTE pin="H" level). Figure 2.5.9 shows the bus timing when external bus width is 8 bits.

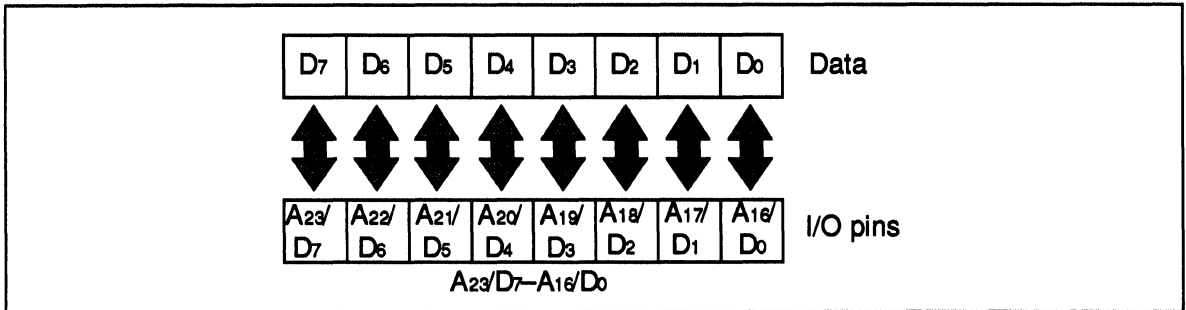


Fig. 2.5.8 Relationship between data bus and I/O pins (when BYTE pin="H" level)

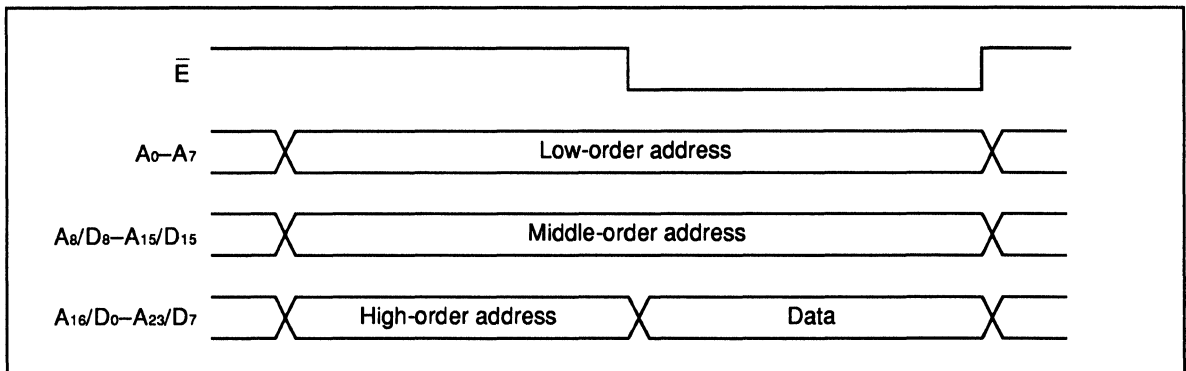


Fig. 2.5.9 Bus timing when external bus width is 8 bits (when BYTE pin="H" level)

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

(3) V_{CC} , V_{SS} pins

These are microcomputer power voltage supply pins. Supply $5V \pm 10\%$ to V_{CC} pin and GND (0V) to V_{SS} pin.

(4) CNV_{SS} pin

Connect to V_{CC} pin.

(5) AV_{CC} , AV_{SS} pins

These are A-D converter power voltage input pins. Connect AV_{CC} pin to V_{CC} pin and AV_{SS} pin to V_{SS} pin.

(6) V_{REF} Input pin

This pin is used to input the reference voltage during A-D conversion. A-D conversion can be performed for analog input voltage from V_{SS} level up to the level of this pin. It is available to supply the optional voltage up to the V_{CC} level.

(7) X_{IN} , X_{OUT} pins

These are clock input and output pins for the internal oscillator circuit. Connect a ceramic resonator or a crystal resonator between X_{IN} and X_{OUT} pins. When using an external clock, connect the clock source to X_{IN} pin and leave X_{OUT} pin open. The maximum clock input frequency of the M37732S4FP, M37732S4AFP, and M37732S4BFP is 8MHz, 16MHz, and 25MHz respectively.

(8) \overline{RESET} Input pin

This pin is used to input the reset signal. The microcomputer is reset state when "L" level is input to this pin. Reset state is removed and a program starts from the address set in the reset vector when this pin is returned to "H" level. Refer to "CHAPTER 3. RESET" for details.

(9) R/\overline{W} output pin

This pin is used to output the read/write signal indicating the data bus state. This pin is at "H" level when the data bus is read, and it is at "L" level when data is written to the data bus. This signal is used for external memory input/output requests.

(10) \overline{BHE} output pin

This pin is used to output the byte high enable signal. This pin is at "L" level when an odd address is accessed. This signal is used to connect the 8-bit memory and I/O when the external bus is used at 16-bit width.

Refer to section "CHAPTER 9. APPLICATION" for memory and I/O connection method.

(11) ALE output pin

This pin outputs the ALE signal to obtain only address signal from the multiplex signals of pins A_8/D_8 – A_{15}/D_{15} and pins A_{16}/D_0 – A_{23}/D_7 . A latch is opened externally when the ALE signal is at "H" level to obtain the address data, and the latched contents are held while the ALE signal is at "L" level.

(12) \overline{HOLD} Input pin

This pin is used to input the Hold request signal. The microcomputer becomes Hold state while this pin is at "L" level. When this pin is returned to "H" level, the Hold state is removed.

Refer to section "2.13 Hold function" for details.

(13) \overline{HLDA} output pin

This pin is used to output the Hold acknowledge signal. The Hold acknowledge signal externally indicates that "L" level is input to the \overline{HOLD} pin and the microcomputer is in Hold state. "L" level is output from this pin while the microcomputer is in Hold state.

FUNCTIONAL DESCRIPTION

2.5 Input/Output pins

(14) \overline{RDY} Input pin

This pin is used to input the Ready signal. The bus cycle \overline{E} can be stopped (Ready state) when "L" level is input to this pin. The port and bus state at inputting "L" level to the \overline{RDY} pin is maintained in Ready state.

Refer to section "2.14 Ready function" for details.

(15) \overline{E} output pin

This pin is used to output the enable signal. Data input/output is performed when the output of this pin is at "L" level. This signal controls the time division multiplexing of address and data.

(16) BYTE pin

This pin is used to select the external bus width. The input level to this pin determines whether the external memory is used with 16-bit data width or 8-bit. When the BYTE pin input level is at "L", the data width is 16 bits. When the BYTE pin input level is at "H", the data width is 8 bits. Refer to section "2.5.3 (2) Data bus" for details. However, the data width is always 16 bits regardless of the BYTE pin level when accessing an internal area.

(17) Clock ϕ_1 output pin

This pin outputs the clock ϕ_1 which is divided the clock to X_{IN} pin by 2.

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6 Interrupts

The suspension of the current operation in order to perform another operation due to a certain event is referred to as an "interrupt". Interrupt is used when there is a request to execute a higher priority routine or when an operation must be performed at a certain timing.

2.6.1 Interrupt functions

The M37732 group has 19 different sources of interrupts. When an interrupt is occurred, a branch is made to the address (branch address) corresponding to the source. The branch address must be stored in the interrupt vector table. The interrupt vector table is allocated at addresses $FFD6_{16}$ to $FFFF_{16}$ in bank 0_{16} . When writing programs, branch must be made to the address in the interrupt vector table corresponding to each interrupt (interrupt vector address). The branch address is the start address of the interrupt processing routines (interrupt service routine). Figure 2.6.1 shows the interrupt mechanism.

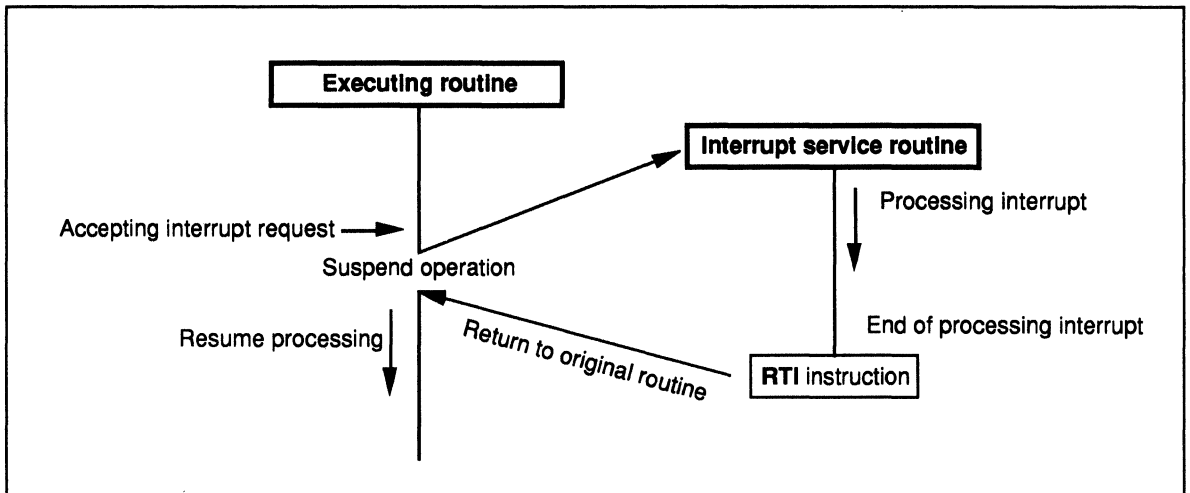


Fig. 2.6.1 Interrupt mechanism

FUNCTIONAL DESCRIPTION

2.6 Interrupts

When an interrupt request is accepted, the contents of the following registers before the interrupt are stored automatically to the stack area in the order ①→②→③.

- ① Program bank register (PG)
- ② Program counter (PC_L, PC_H)
- ③ Processor status register (PS_L, PS_H)

The procedure of storing these registers depends on whether the contents of the stack pointer S are even or odd. When the contents of the stack pointer S are even, the contents of the program counter PC and processor status register PS are stored in 16-bit unit. When the contents of the stack pointer S are odd, they are stored in 8-bit unit. Figure 2.6.2 shows the state of the stack area when an interrupt request is accepted.

When interrupt processing completes, the control must be returned to the original routine to resume processing. Therefore, the RTI instruction is used to return to the original routine and the above registers stored to the stack area are restored in the order of ③→②→① to resume operation.

Only the above registers ① to ③ are stored automatically when an interrupt request is accepted. The user is responsible for storing other necessary registers by program.

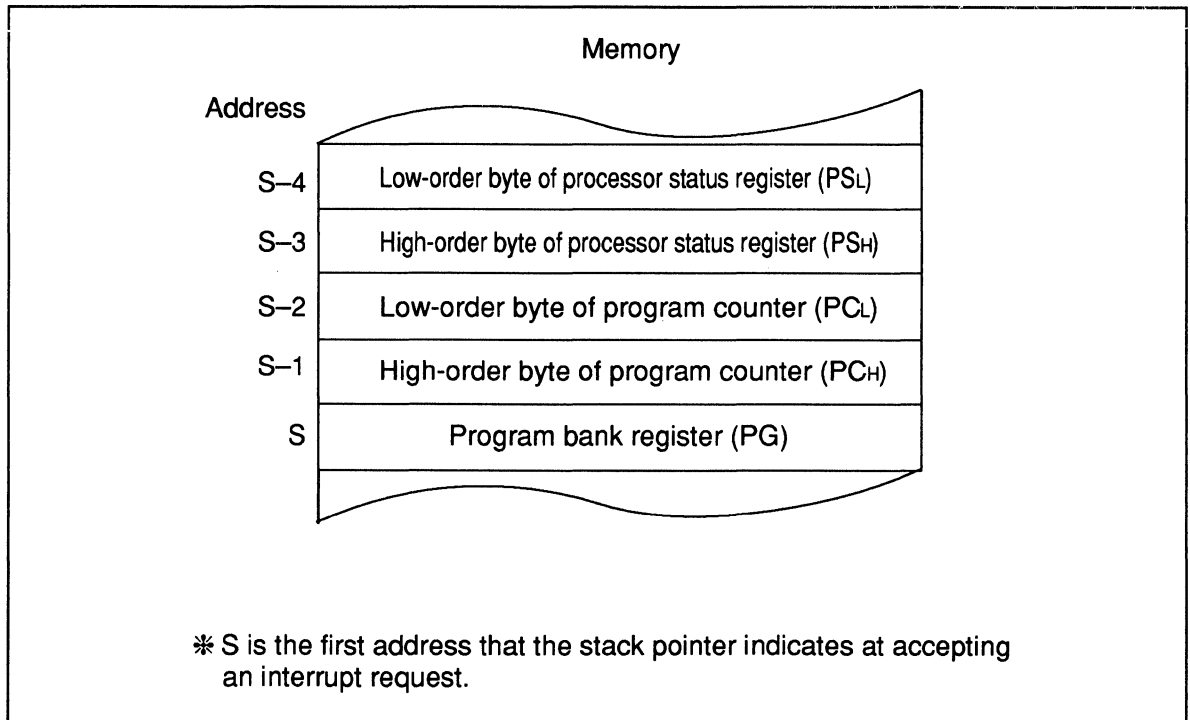


Fig. 2.6.2 State of the stack area when an interrupt request is accepted

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.2 Sources of interrupts

Table 2.6.1 shows the sources of interrupts and the corresponding vector address. Store the start address of the interrupt service routine at the vector address shown in this table by program.

Table 2.6.1 Interrupt sources and vector address

Interrupt source	Vector address		Remarks
	High-order address	Low-order address	
Reset (Note 1)	00FFFF ₁₆	00FFFE ₁₆	Non-maskable
Zero division	00FFFD ₁₆	00FFFC ₁₆	Non-maskable software interrupt
BRK instruction	00FFFB ₁₆	00FFFA ₁₆	Non-maskable software interrupt
DBC (Note 2)	00FFF9 ₁₆	00FFF8 ₁₆	Not used normally
Watchdog timer	00FFF7 ₁₆	00FFF6 ₁₆	Non-maskable interrupt
INT ₀	00FFF5 ₁₆	00FFF4 ₁₆	External interrupt due to INT ₀ pin input signal
INT ₁	00FFF3 ₁₆	00FFF2 ₁₆	External interrupt due to INT ₁ pin input signal
INT ₂	00FFF1 ₁₆	00FFF0 ₁₆	External interrupt due to INT ₂ pin input signal
Timer A0	00FFEF ₁₆	00FFEE ₁₆	Timer A0 internal interrupt
Timer A1	00FFED ₁₆	00FFEC ₁₆	Timer A1 internal interrupt
Timer A2	00FFEB ₁₆	00FFEA ₁₆	Timer A2 internal interrupt
Timer A3	00FFE9 ₁₆	00FFE8 ₁₆	Timer A3 internal interrupt
Timer A4	00FFE7 ₁₆	00FFE6 ₁₆	Timer A4 internal interrupt
Timer B0	00FFE5 ₁₆	00FFE4 ₁₆	Timer B0 internal interrupt
Timer B1	00FFE3 ₁₆	00FFE2 ₁₆	Timer B1 internal interrupt
Timer B2	00FFE1 ₁₆	00FFE0 ₁₆	Timer B2 internal interrupt
UART0 reception	00FFDF ₁₆	00FFDE ₁₆	Valid only when the UART0 function is selected.
UART0 transmission	00FFDD ₁₆	00FFDC ₁₆	
UART1 reception	00FFDB ₁₆	00FFDA ₁₆	Valid only when the UART1 function is selected.
UART1 transmission	00FFD9 ₁₆	00FFD8 ₁₆	
A-D conversion	00FFD7 ₁₆	00FFD6 ₁₆	Internal interrupt due to completion of A-D conversion

Note 1: Reset is included in this table.

Note 2: The DBC is an interrupt for the exclusive use of the debug control interrupt and is not used, usually.

FUNCTIONAL DESCRIPTION

2.6 Interrupts

Each interrupt source is described below.

(1) Internal Interrupt

Table 2.6.2 shows the sources of internal interrupt request.

Table 2.6.2 Internal Interrupt request sources

Interrupt	Interrupt request source
Zero division	Occurs when 0 is specified as the divisor for the DIV instruction. (Refer to "MELPS 7700 SOFTWARE MANUAL")
BRK instruction	Occurs when the BRK instruction is executed. (Refer to "MELPS 7700 SOFTWARE MANUAL")
Watchdog timer	Occurs when the topmost bit of the 12-bit watchdog timer becomes "0". (Refer to section "2.12 Watchdog timer")
Timer Ai	Occurs when timer Ai (i=0 to 4) underflows or overflows. (Refer to section "2.7 Timer A")
Timer Bi	Occurs when timer Bi (i=0 to 2) underflows or overflows. (Refer to section "2.8 Timer B")
UARTi reception	Occurs during UARTi (i=0,1) receive. (Refer to section "2.10 Serial I/O")
UARTi transmission	Occurs during UARTi (i=0,1) transmit. (Refer to section "2.10 Serial I/O")
A-D conversion	Occurs when A-D conversion completes. (Refer to section "2.11 A-D Converter")

(2) External Interrupt ($\overline{\text{INT}}_0$ to $\overline{\text{INT}}_2$ Interrupts)

M37732 group has three external interrupts ($\overline{\text{INT}}_0$ to $\overline{\text{INT}}_2$). These interrupt requests are occurred by input level or input edge to pins $\overline{\text{INT}}_0$ – $\overline{\text{INT}}_2$. The interrupt request sources can be selected by using bits 4 and 5 of the $\overline{\text{INT}}_i$ (i=0 to 2) interrupt control register shown in Figure 2.6.4. Table 2.6.3 shows the sources of $\overline{\text{INT}}_i$ interrupt requests.

Pins $\overline{\text{INT}}_0$ – $\overline{\text{INT}}_2$ are shared with ports P6₂–P6₄. Therefore, the corresponding bit in the port P6 direction register must be cleared to "0" in order to use these pins as external interrupt input pins. If the $\overline{\text{INT}}_i$ interrupts are not used, the $\overline{\text{INT}}_i$ interrupt priority level (refer to next section) should be set to 0 because the $\overline{\text{INT}}_i$ interrupts always monitor the state of ports P6₂–P6₄ to raise interrupt requests.

The input signal to the $\overline{\text{INT}}_i$ pins must have pulse width greater than 250ns at "H" level or "L" level regardless of the external clock input frequency $f(X_{IN})$.

Table 2.6.3 $\overline{\text{INT}}_i$ Interrupt request sources

b5	b4	Interrupt request source
0	0	Interrupt request occurs when falling edge of the signal input to the $\overline{\text{INT}}_i$ pin.
0	1	Interrupt request occurs when rising edge of the signal input to the $\overline{\text{INT}}_i$ pin.
1	0	Interrupt request occurs when the $\overline{\text{INT}}_i$ pin level becomes "H".
1	1	Interrupt request occurs when the $\overline{\text{INT}}_i$ pin level becomes "L".

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.3 Interrupt control

The enabling and disabling of interrupts are controlled by the interrupt request bit, interrupt priority level, processor interrupt priority level (IPL), and interrupt disable flag (I) (excluding non-maskable interrupts). The interrupt disable flag and the processor interrupt priority level are assigned to the processor status register (PS). The interrupt request bit and the interrupt priority level selection bits are assigned to the interrupt control register of the respective interrupt. Figure 2.6.3 shows the memory map of the interrupt control register and Figure 2.6.4 shows the structure. However, there is no interrupt control register for non-maskable interrupts such as zero division interrupt, the BRK instruction interrupt, and watchdog timer interrupt.

- Non-maskable interrupt: An interrupt that branches to the interrupt service routine regardless of the interrupt disable flags.
- Maskable interrupt: An interrupt that can be disabled with the interrupt control flags.

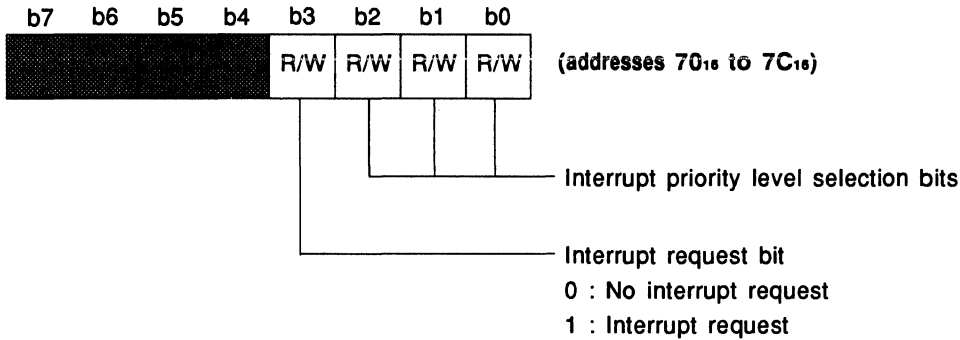
Address	
70 ₁₆	A-D conversion interrupt control register
71 ₁₆	UART0 transmission interrupt control register
72 ₁₆	UART0 receive interrupt control register
73 ₁₆	UART1 transmission interrupt control register
74 ₁₆	UART1 receive interrupt control register
75 ₁₆	Timer A0 interrupt control register
76 ₁₆	Timer A1 interrupt control register
77 ₁₆	Timer A2 interrupt control register
78 ₁₆	Timer A3 interrupt control register
79 ₁₆	Timer A4 interrupt control register
7A ₁₆	Timer B0 interrupt control register
7B ₁₆	Timer B1 interrupt control register
7C ₁₆	Timer B2 interrupt control register
7D ₁₆	$\overline{\text{INT}}_0$ interrupt control register
7E ₁₆	$\overline{\text{INT}}_1$ interrupt control register
7F ₁₆	$\overline{\text{INT}}_2$ interrupt control register

Fig. 2.6.3 Interrupt control register memory map

FUNCTIONAL DESCRIPTION

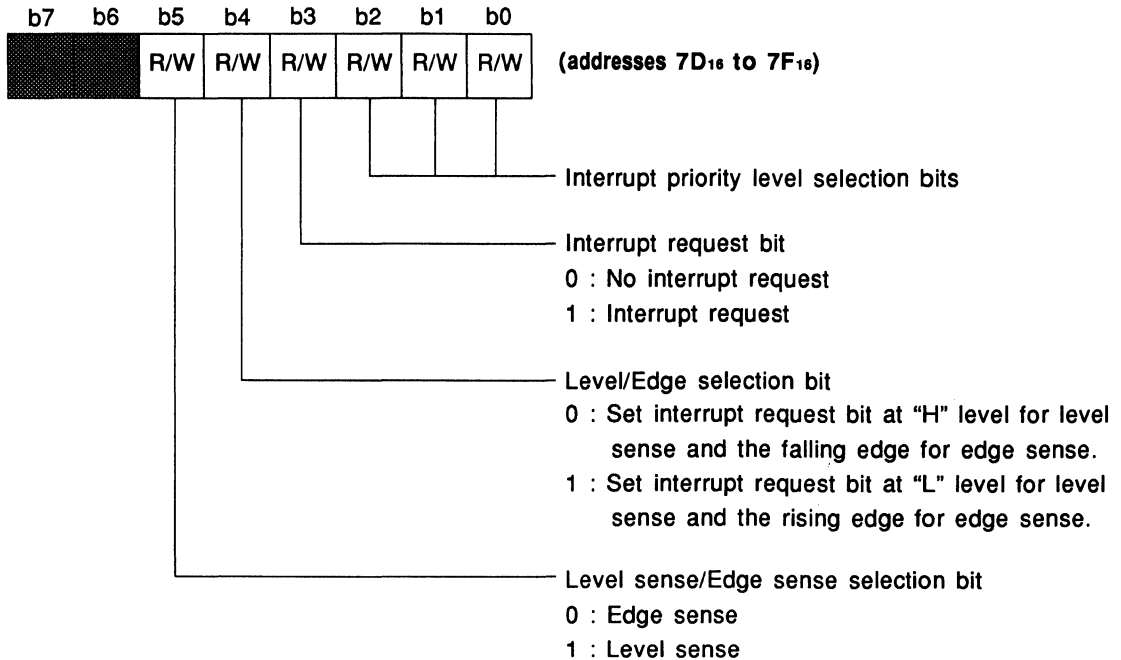
2.6 Interrupts

● A-D conversion, UART0 and 1 transmission, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers



Note: Bits 4 to 7 are undefined at reading.
 Bits 0 to 3 are cleared to "0" at reset.
 Use the **SEB** and **CLB** instructions when setting each interrupt control register.

● INT₀ to INT₂ interrupt control registers



Note: Bits 6 and 7 are undefined at reading.
 Bits 0 to 5 are cleared to "0" at reset.
 Use the **SEB** and **CLB** instructions when setting each interrupt control register.

Fig. 2.6.4 Interrupt control register structure

The interrupt control bits are described below.

(1) Interrupt disable flag (I flag)

The interrupt disable flag (I flag) is assigned to the processor status register bit 2. This flag can be used to disable all maskable interrupts. All maskable interrupts are disabled when the I flag is set to "1" and enabled when it is cleared to "0". This flag is set to "1" at reset and must be cleared to "0" if interrupts are to be enabled.

(2) Interrupt request bit

When an interrupt request occurs, the interrupt request bit which is bit 3 of the corresponding interrupt control register is set to "1". The interrupt request bit remains set until the interrupt request is accepted, and is cleared to "0" when the interrupt request is accepted.

This bit is used to indicate that an interrupt request has occurred. This bit can also be set or cleared by program. Use the **SEB** and **CLB** instructions when setting interrupt request bit.

The $\overline{\text{INTI}}$ interrupt request bit's function is not available when used in level sense mode.

(3) Interrupt priority level and processor interrupt priority level (IPL)

An interrupt priority level between 0 and 7 can be assigned to each interrupt by using the interrupt priority level selection bits which are assigned to bits 0 to 2 of each interrupt control register. Use the **SEB** and **CLB** instructions when setting interrupt priority level selection bits. When an interrupt request occurs, this priority level is compared with the processor interrupt priority level in the processor status register.

Interrupt priority level > Processor interrupt priority level (IPL)

An interrupt is enabled when the above condition is satisfied. Therefore, an interrupt can be disabled by setting its priority level to 0.

The interrupt disable flag, interrupt request bit, interrupt priority level, and IPL are independent of each other and do not affect each other. An interrupt is occurred only when all of these bits are properly set. These bits can be used to control interrupt priorities in a variety of ways by program.

Table 2.6.4 shows the setting of interrupt priority levels and Table 2.6.5 shows the interrupt enable levels corresponding to IPL setting.

FUNCTIONAL DESCRIPTION

2.6 Interrupts

Table 2.6.4 Setting of Interrupt priority levels

Interrupt control register			Interrupt priority level	Priority
b2	b1	b0		
0	0	0	Level 0 (Interrupt disabled)	—
0	0	1	Level 1	Low ↓ High
0	1	0	Level 2	
0	1	1	Level 3	
1	0	0	Level 4	
1	0	1	Level 5	
1	1	0	Level 6	
1	1	1	Level 7	

Table 2.6.5 Interrupt enable levels corresponding to IPL setting

IPL ₂	IPL ₁	IPL ₀	Enabled interrupt priority level
0	0	0	Enable level 1 and above interrupts.
0	0	1	Enable level 2 and above interrupts.
0	1	0	Enable level 3 and above interrupts.
0	1	1	Enable level 4 and above interrupts.
1	0	0	Enable level 5 and above interrupts.
1	0	1	Enable level 6 and above interrupts.
1	1	0	Enable level 7 interrupts.
1	1	1	Disable all maskable interrupts.

IPL₀: Processor status register bit 8
 IPL₁: Processor status register bit 9
 IPL₂: Processor status register bit 10

2.6.4 Interrupt priority level

All interrupts have an assigned priority level. When more than one interrupt request occurs during the same sampling timing (timing in which interrupt requests are checked) while accepting all interrupt requests are enabled, the one with the highest priority level is accepted.

The 16 interrupt sources of all 19 sources except for software interrupts (zero division and the **BRK** instruction interrupt) and watchdog timer interrupt can be set by program using the interrupt priority level selection bits in the interrupt control register. Reset (highest priority level) and watchdog timer interrupt priority levels are set by the hardware. Figure 2.6.5 shows the hardware controlled interrupt priority levels.

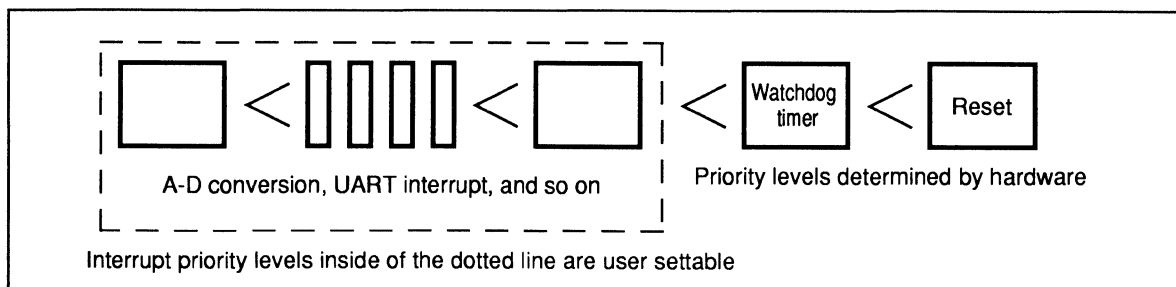


Fig. 2.6.5 Hardware controlled interrupt priority levels

2.6.5 Interrupt priority level detection circuit

The M37732 group is equipped with an interrupt priority level detection circuit to select the highest priority level when more than one interrupt request occurs during the same sampling timing. Figure 2.6.6 shows the interrupt priority level detection circuit.

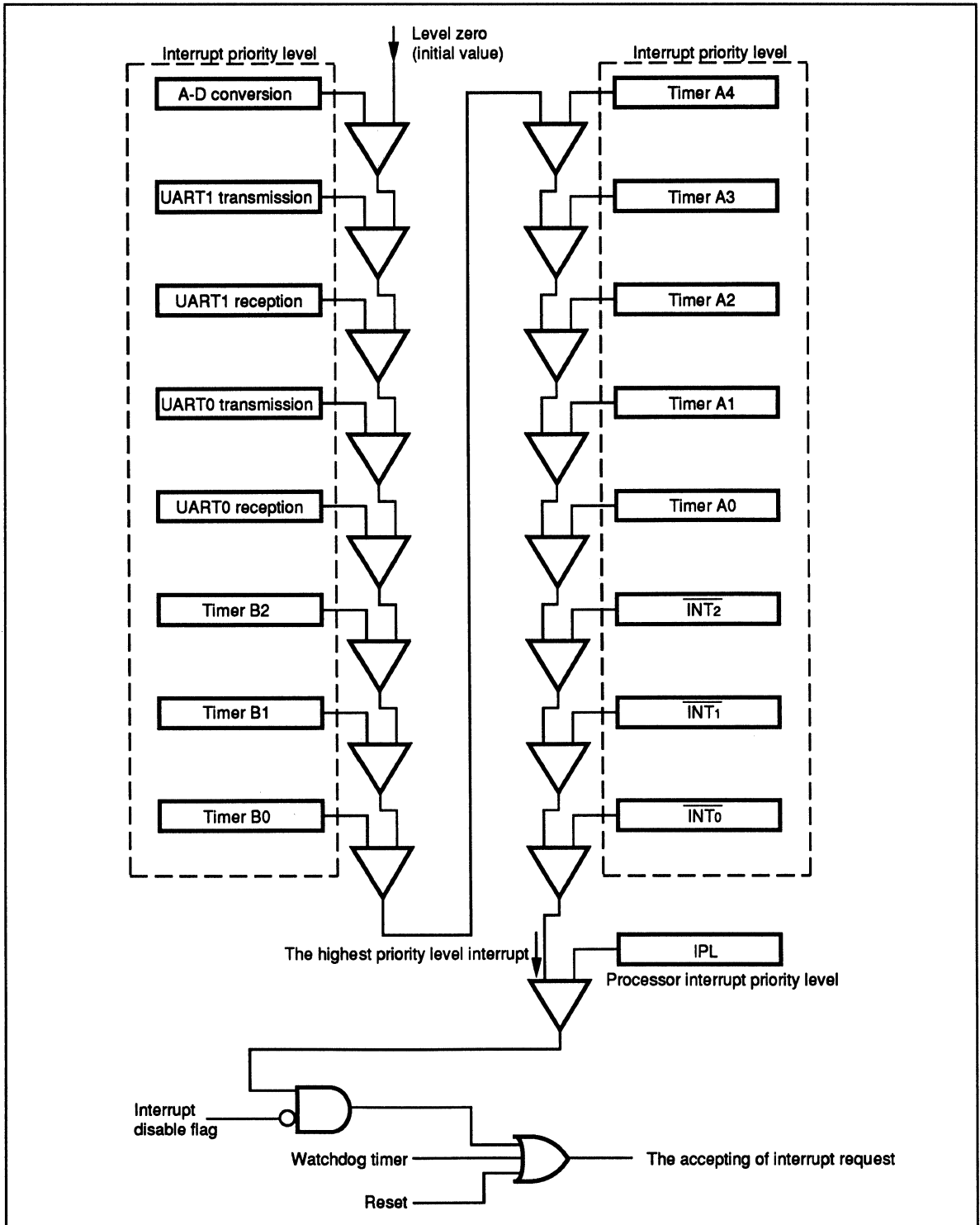


Fig. 2.6.6 Interrupt priority level detection circuit

FUNCTIONAL DESCRIPTION

2.6 Interrupts

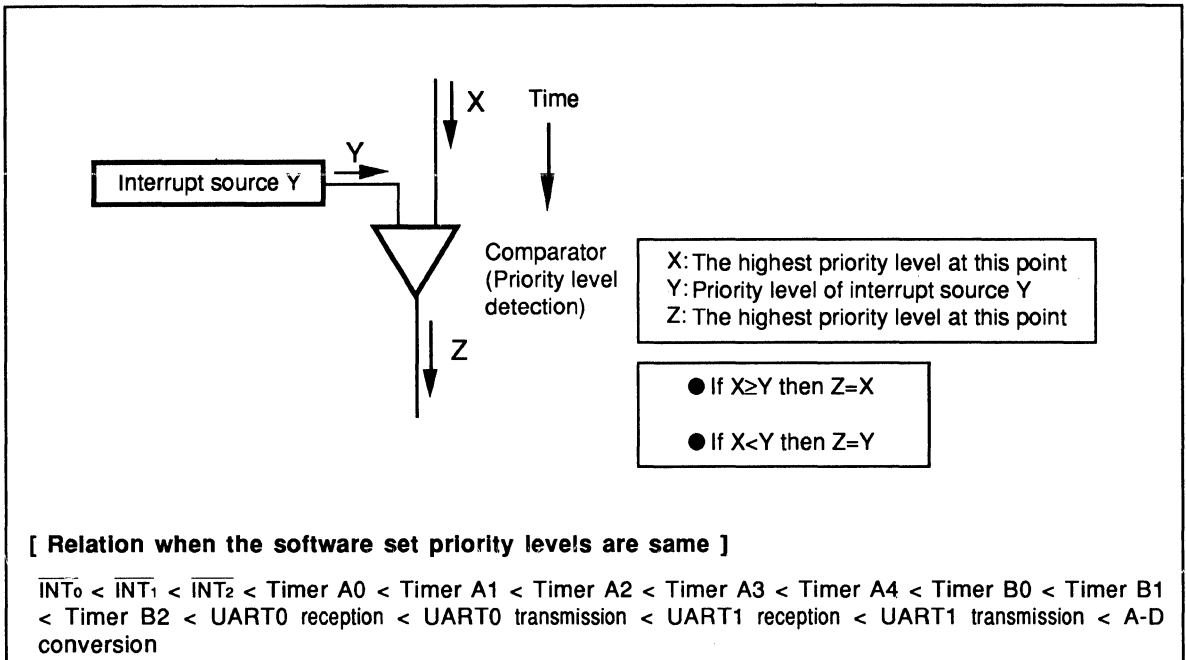


Fig. 2.6.7 Interrupt priority level detection model

The operation of the interrupt priority level detection circuit shown in Figure 2.6.6 is described with interrupt priority level detection model shown in Figure 2.6.7.

The priority level of the requested interrupt (Y in Figure 2.6.7) is compared with the priority level of the upstream interrupt (X in Figure 2.6.7) (initial priority level is 0) in the order shown in Figure 2.6.6 and the higher level interrupt is sent downstream for comparison with next interrupt (Z in Figure 2.6.7). Unrequested interrupts are not compared and upstream interrupt is passed unchanged to downstream. If the priority levels are equal, the upstream interrupt is selected. Therefore, the following relation exists if the software set priority levels are the same.

$\overline{INT}_0 < \overline{INT}_1 < \overline{INT}_2 < \text{Timer A0} < \text{Timer A1} < \text{Timer A2} < \text{Timer A3} < \text{Timer A4} < \text{Timer B0} < \text{Timer B1} < \text{Timer B2} < \text{UART0 reception} < \text{UART0 transmission} < \text{UART1 reception} < \text{UART1 transmission} < \text{A-D conversion}$

When there are multiple interrupts during the same sampling timing, the interrupt with the highest priority level is selected as the result of this comparison. Then, that interrupt request is accepted and its interrupt service routine is executed if its interrupt priority level is higher than the processor interrupt priority level (IPL) and the interrupt disable flag is "0".

The detection of interrupt priority level is synchronized with the sampling pulse occurred during the operation code fetch cycle of CPU. While the interrupt priority level is being checked, the interrupt request bit and the interrupt priority level are latched so that they do not change. They are sampled at the first half of the operation code fetch cycle and latched from the second half to the end of the level detection. Note that while the priority level is being checked, no sampling pulse is occurred even when it is the operation code fetch cycle. (Refer to Figure 2.6.8.)

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.6 Interrupt priority level detection time

With the M37732 group, the time which it takes for the interrupt priority level detection circuit to determine the level of an interrupt can be set by program. Figure 2.6.8 shows the interrupt priority level detection time. The detection time can be set to 7, 4, or 2 cycles according to the contents of the interrupt priority detection time selection bits (bits 4, and 5 at address 5E16) in the processor mode register. The interrupt priority detection time selection bits are cleared to "00" at reset and the 7 cycles at internal clock ϕ are selected for interrupt priority level detection time. Normally, use these bits set to "10".

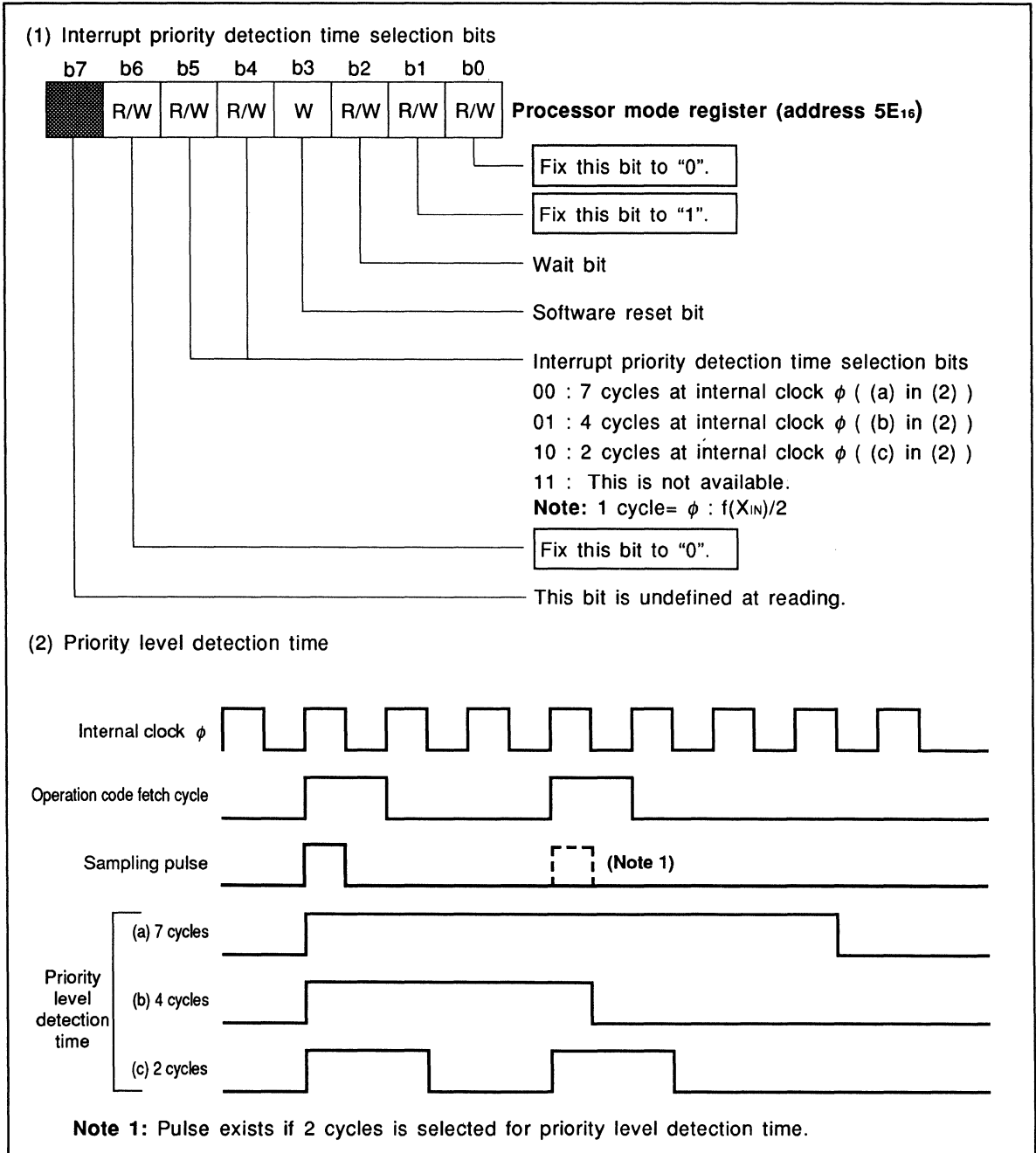


Fig. 2.6.8 Interrupt priority level detection time

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.7 Interrupt processing sequence

The sequence of events from the receiving of interrupt request in the main routine to the start of the interrupt service routine is referred to as the interrupt processing sequence.

When an interrupt request is accepted, interrupt processing starts from the next cycle of the instruction at the time of the accepting interrupt request. Figure 2.6.9 shows the interrupt processing sequence.

After execution of an instruction at the time of the accepting interrupt request completes, an INTACK (Interrupt Acknowledge) sequence is executed and control is passed to the beginning of the interrupt service routine allocated in bank 0₁₆. The INTACK sequence operates as follows.

- ① The contents of the program bank register (PG) just before the INTACK sequence are stored to stack.
- ② The contents of the program counter (PC) just before the INTACK sequence are stored to stack.
- ③ The contents of the processor status register (PS) just before the INTACK sequence are stored to stack.
- ④ The interrupt disable flag is set to "1".
- ⑤ The interrupt request bit of the accepted interrupt is cleared to "0".
- ⑥ The interrupt priority level of the accepted interrupt is set in IPL.
- ⑦ The contents of the program bank register (PG) are set to "00₁₆" and the interrupt vector address is loaded in the program counter (PC).

The INTACK sequence requires a minimum 13 cycles ($1 \text{ cycle} = \phi = f(X_{IN})/2$). Figure 2.6.10 shows the INTACK sequence timing.

The interrupt service routine is started after completing the INTACK sequence.

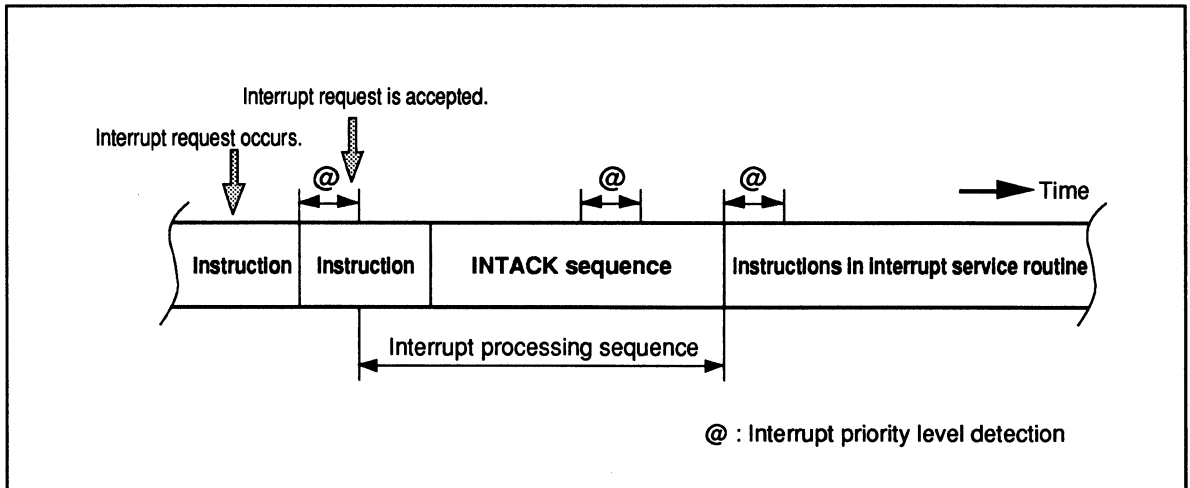


Fig. 2.6.9 Interrupt processing sequence

FUNCTIONAL DESCRIPTION

2.6 Interrupts

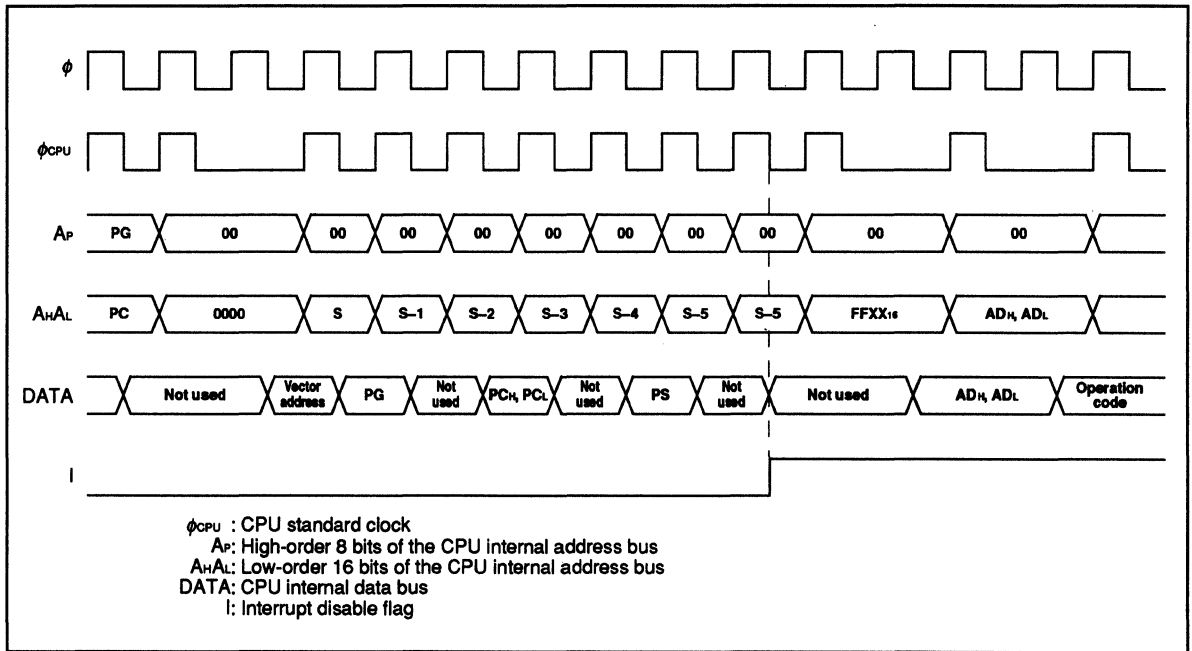


Fig. 2.6.10 INTACK sequence timing

(1) Change In IPL

When an interrupt request is accepted, the IPL in the processor status register is replaced by the interrupt priority level of the accepted interrupt.

If the interrupt is a reset, watchdog timer, or software interrupt, the value shown in Table 2.6.6 is set in the IPL. This operation is meaningful when multiple interrupts are used (refer to section "2.6.9 Multiple interrupts").

Table 2.6.6 Change In IPL

Interrupt source	Change in processor interrupt priority level
Reset	0 (000 ₂)
Watchdog timer	7 (111 ₂)
Zero division	No change
BRK instruction	No change
Other interrupts	Interrupt priority level of the accepted interrupt request

FUNCTIONAL DESCRIPTION

2.6 Interrupts

(2) Storing registers

The register storing operation performed during INTACK sequence depends on whether the contents of the stack pointer S at the time of the accepting interrupt request are even or odd.

When the contents of the stack pointer S are even, the contents of the program counter PC and processor status register PS are stored as 16-bit simultaneously at each other. When the contents of the stack pointer S are odd, they are stored in 8-bit unit. Figure 2.6.11 shows the register storing operation.

In the INTACK sequence, only the contents of the program bank register PG, program counter PC, and processor status register PS are stored to stack area. Other registers must be stored by program at the beginning of the interrupt service routine as necessary.

The M37732 group provides the PSH instruction which stores all CPU registers except for the stack pointer with a single instruction.

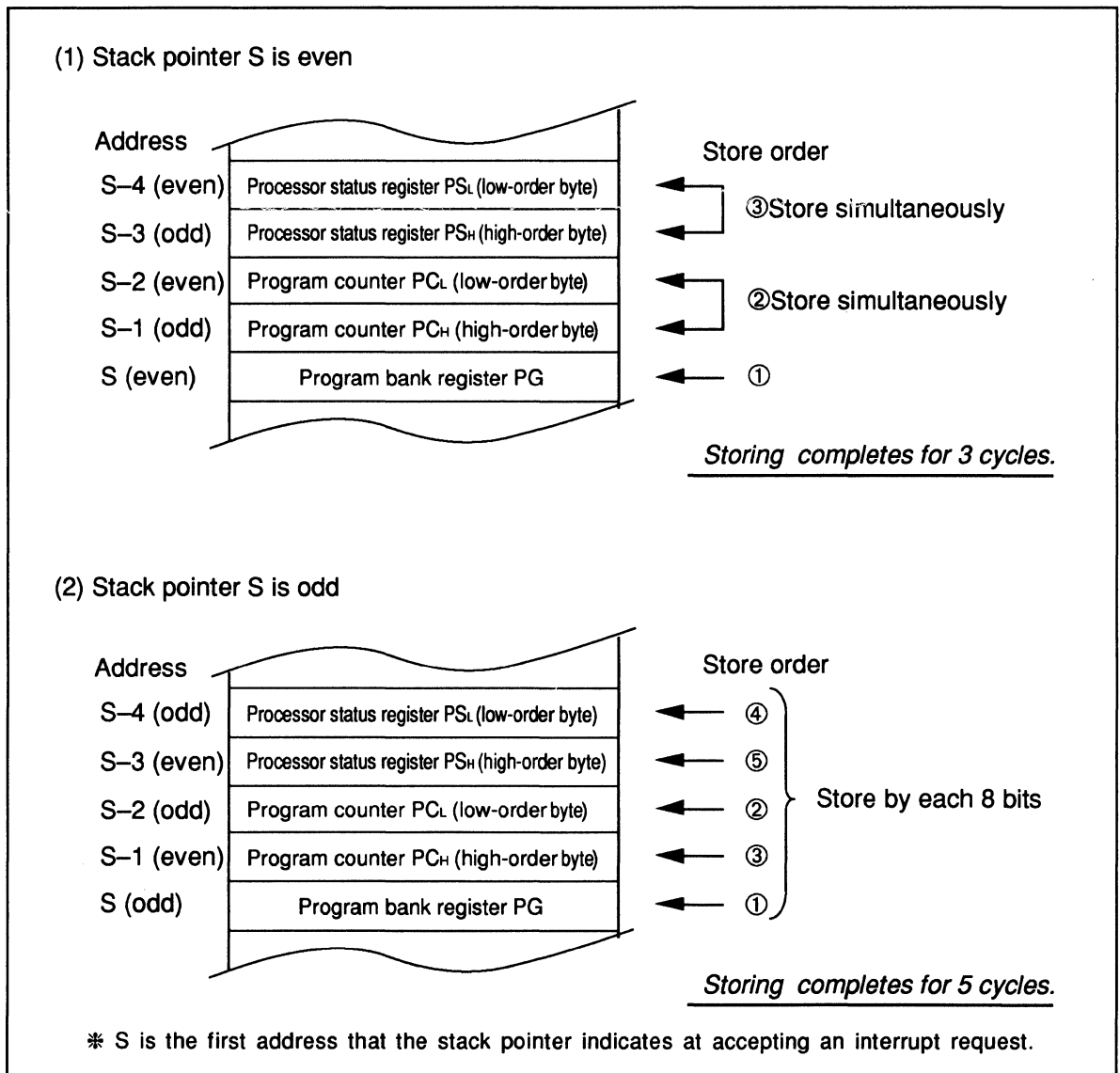


Fig. 2.6.11 Register storing operation

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.8 Returning from an interrupt service routine

The RTI instruction is used at the end of the interrupt service routine to return to the interrupted routine and continue processing. The RTI instruction restores the contents of the program bank register PG, the program counter PC, and the processor status register PS stored before entering the interrupt service routine to their original registers.

The registers stored within the interrupt service routine must be restored with the PUL instruction before executing the RTI instruction. When they are restored with the PUL instruction, use the same data/register length as they were stored. The state of other interrupt request bits are retained after branching to the interrupt service routine. Therefore, if these interrupts are to be disabled after returning, these interrupt request bits must be cleared to "0" before executing the RTI instruction.

2.6.9 Multiple Interrupts

The interrupt disable flag I is set to "1" in order to disable further interrupts and the interrupt request bit of the accepted interrupt is cleared to "0" when a branch is made to an interrupt service routine. However, if there are multiple interrupt requests, the interrupt request bit of the interrupt that was rejected by the interrupt priority level detection circuit remains set to "1".

The IPL (processor interrupt priority level) in the processor status register changes to the priority level of the accepted interrupt. Therefore, if the interrupt disable flag I is cleared to "0" in the interrupt service routine for the accepted interrupt, interrupt request with priority level higher than the current interrupt can be accepted as long as IPL is unchanged. This is referred to as multiple interrupts.

Figure 2.6.12 shows the multiple interrupt mechanism.

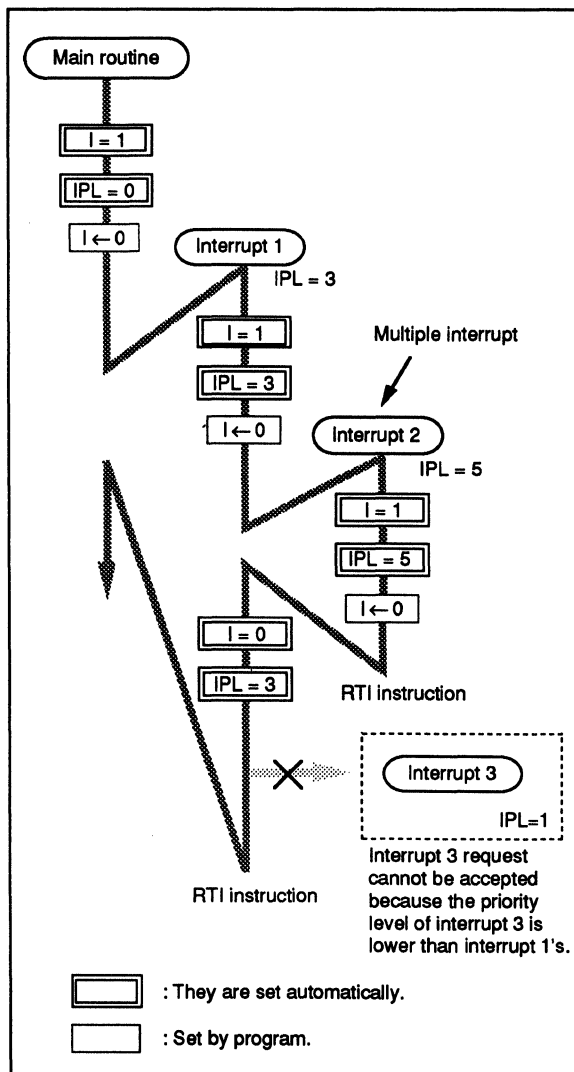


Fig. 2.6.12 Multiple Interrupt mechanism

FUNCTIONAL DESCRIPTION

2.6 Interrupts

2.6.10 Interrupt response time and interrupt delay time

When an interrupt request occurs, the priority level must be checked and the INTACK sequence must be executed before interrupt service routine can start. The interval between the interrupt request and the start of the interrupt service routine is referred to as the interrupt response time. This is shown in Figure 2.6.13. Interrupt priority level detection is performed at the beginning of each instruction and during the INTACK sequence. This is performed in parallel with the instruction execution. Therefore, the interrupt response time is the sum of ① through ③ as follows (in Figure 2.6.13):

- ① The interval from the occurrence of the interrupt request until the instruction being executed completes.
- ② The interval from the start of the next instruction (start of interrupt priority level detection) until the end of the instruction being executed at the end of priority level detection.
- ③ Time required to execute the INTACK sequence (13 cycles minimum, 15 cycles maximum).

If registers are stored at the beginning of the interrupt service routine, the time required for this operation (context switching time) must also be added. The sum of the interrupt response time and the context switching time is the interrupt delay time (refer to Figure 2.6.13). This is the time required for the interrupt processing program to start after an interrupt request occurs. The interrupt delay time is expressed by the following equation.

$$\text{Interrupt delay time} = (\text{time required to execute instruction 1}) + (\text{time required to execute instruction 2}) - 0.5\phi + (\text{INTACK sequence interval}) + (\text{context switching time})$$

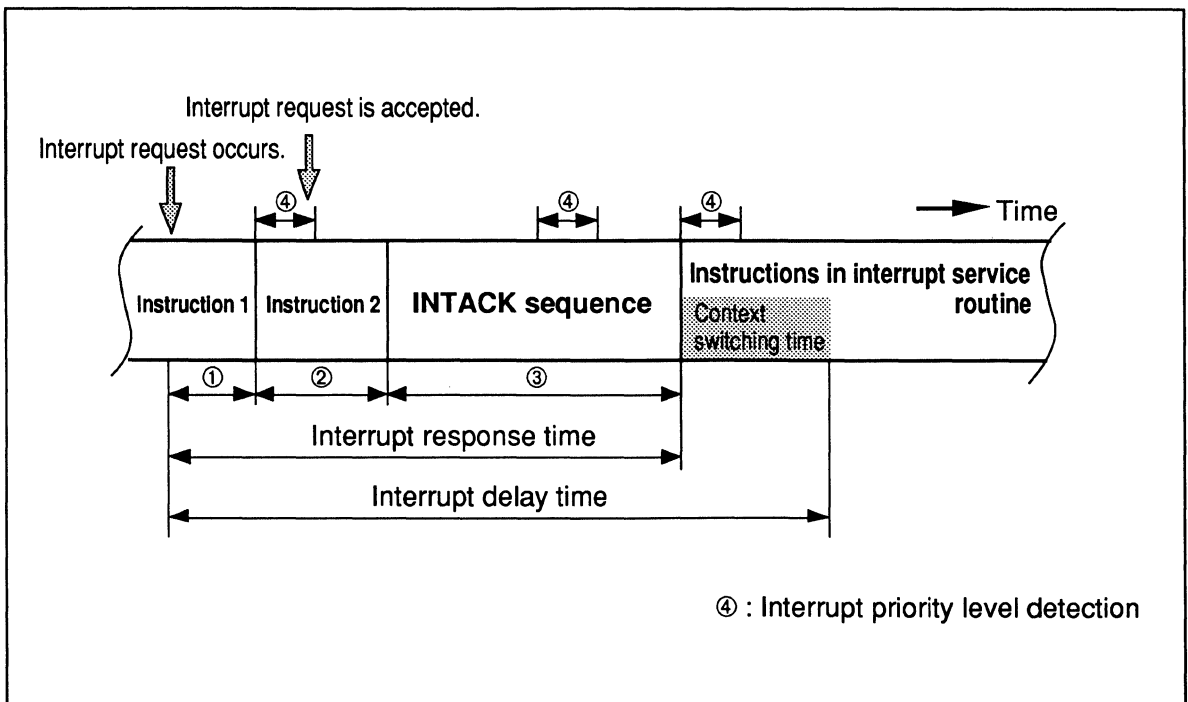


Fig. 2.6.13 Interrupt response time and interrupt delay time

FUNCTIONAL DESCRIPTION

2.6 Interrupts

[Precautions when using interrupts]

1. Use the **SEB** and **CLB** instructions for setting each interrupt control register (addresses 70₁₆ to 7F₁₆).
2. When the $\overline{\text{INT}}_i$ interrupt is used in level sense mode, the $\overline{\text{INT}}_i$ interrupt request bit's function is not available.

The interrupt request is not retained when changing from valid level to invalid level if interrupt request is not accepted during valid level. Figure 2.6.14 shows the $\overline{\text{INT}}_i$ interrupt during level sense mode. If the level of the $\overline{\text{INT}}_i$ pin is valid (did not change from valid level to invalid level) when returning to the original routine after processing an interrupt, $\overline{\text{INT}}_i$ interrupt is occurred as shown in Figure 2.6.15.

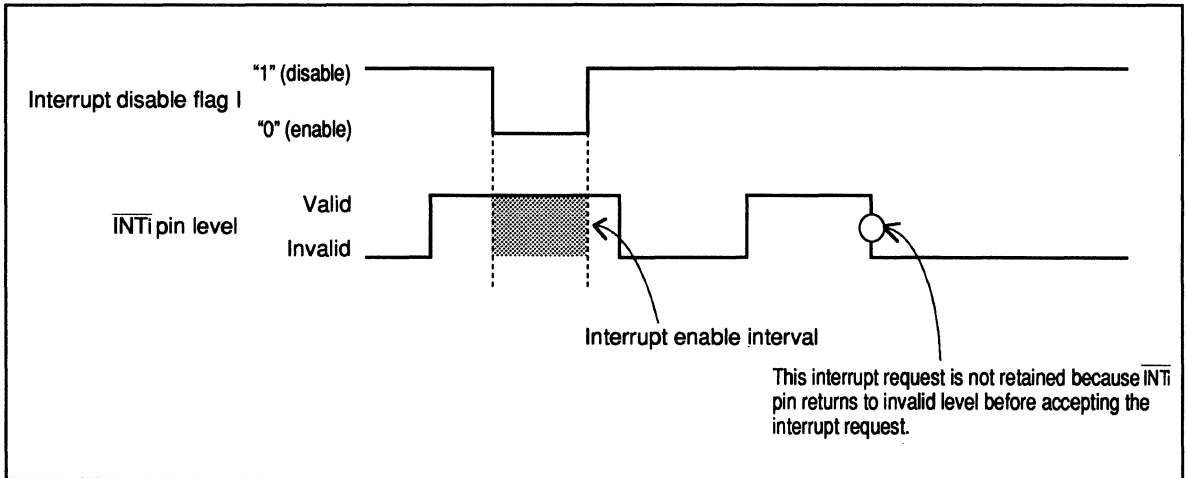


Fig. 2.6.14 $\overline{\text{INT}}_i$ Interrupt during level sense mode

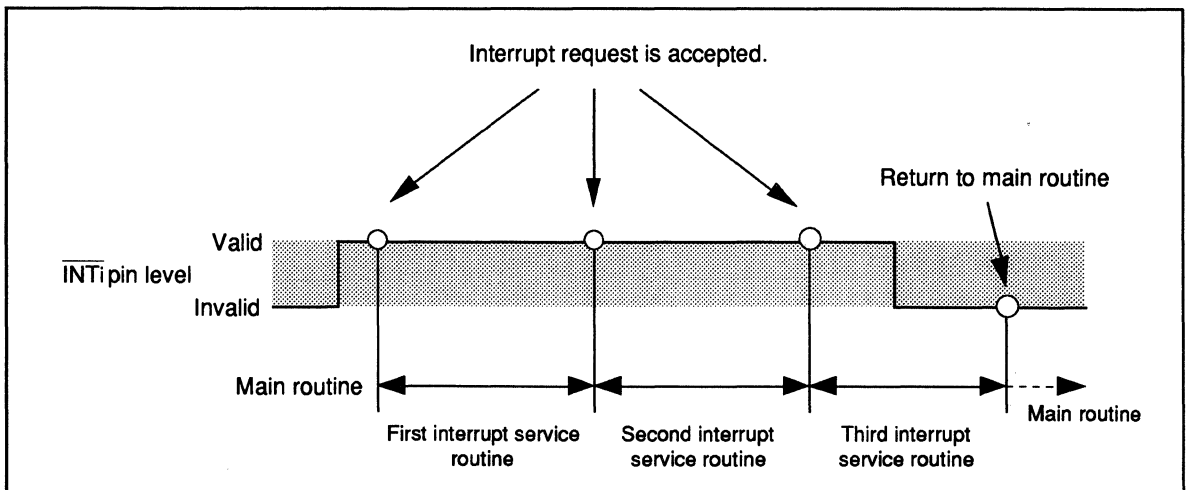


Fig. 2.6.15 Repeating $\overline{\text{INT}}_i$ interrupt (level sense)

FUNCTIONAL DESCRIPTION

2.6 Interrupts

3. To change the $\overline{\text{INT}}_i$ interrupt from level sense to edge sense, set the $\overline{\text{INT}}_i$ interrupt control register in the sequence shown in Figure 2.6.16 (1).

To change the INT_i interrupt polarity, set the $\overline{\text{INT}}_i$ interrupt control register in the sequence shown in Figure 2.6.16 (2).

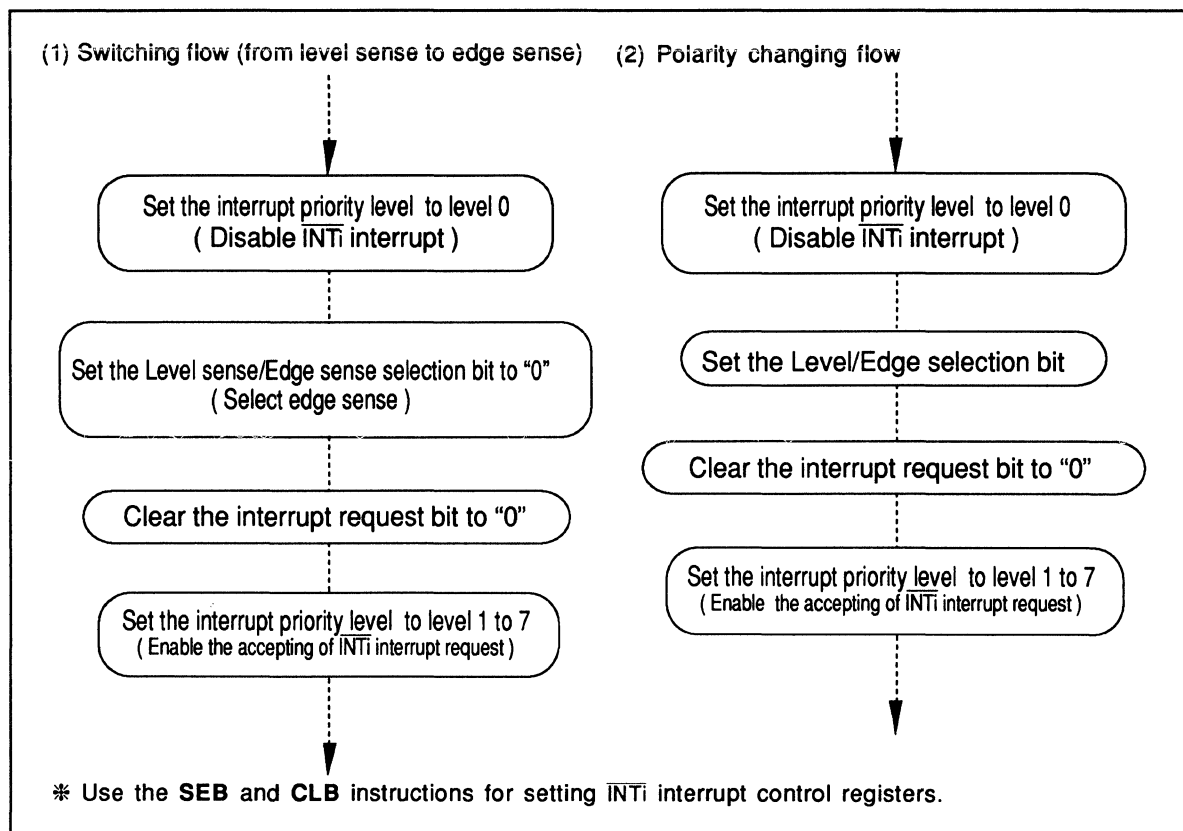


Fig. 2.6.16 INT_i interrupt control register setting flow

FUNCTIONAL DESCRIPTION

2.7 Timer A

2.7 Timer A

Timer A consists of five 16-bit timers (timers A0 to A4). External output is the main function of these timers. Timers A0 to A4 operate independently and each can operate in one of four different modes.

2.7.1 Timer A description

Timer Ai (i=0 to 4) has four operating modes as described below. These timers have identical functions except the two-phase pulse signal processing function in event counter mode.

The timer I/O pins are shared with ports P5₀–P5₇, P6₀, and P6₁.

●Timer mode

This mode counts the selected internal clock. The counter is decremented by 1 each time a count source (selected clock) is input and a timer Ai interrupt request occurs when the contents of the counter reach 0000_{16} → reloaded value "n" (hereafter referred to as underflow).

Gate function (control count operation with the input signal to the TAI_{IN} pin) and pulse output function (output from the TAI_{OUT} pin, a signal that changes polarity each time the counter underflows) are available and can be selected by program.

●Event counter mode

This mode counts the external clock input to the TAI_{IN} pin. The counter can be incremented (add 1 to the contents of the counter with each clock input) or decremented (subtract 1 from the contents of the counter with each clock input) by program or with an external signal. In case used as an increment counter*¹, a timer Ai interrupt request occurs when the counter reaches $FFFF_{16}$ → reloaded value "n" (hereafter referred to as overflow). In case used as a decrement counter*², a timer Ai interrupt request occurs when the counter underflows.

In addition, the pulse output function (output from the TAI_{OUT} pin, a signal that changes polarity each time the counter underflows or overflows) can be selected by program. Timers A2, A3, and A4 also have two-phase pulse signal processing function which controls the counter increment/decrement by inputting two-phase pulse with phase shifted by 90 degrees.

Increment counter*¹ The counter can be incremented (add 1 to the contents of the counter with each clock input)

Decrement counter*² The counter can be decremented (subtract 1 from the contents of the counter with each clock input)

●One-shot pulse mode

In this mode, the timer is driven by an internal or external trigger and "H" level is output from the TAI_{OUT} pin for an arbitrary interval.

●Pulse width modulation (PWM) mode

In this mode, an arbitrary pulse width signal is output repeatedly from the TAI_{OUT} pin. PWM output is started by an internal or external trigger.

FUNCTIONAL DESCRIPTION

2.7 Timer A

2.7.2 Block description

Figure 2.7.1 shows the block diagram of timer Ai. It is followed by the description of timer Ai related registers.

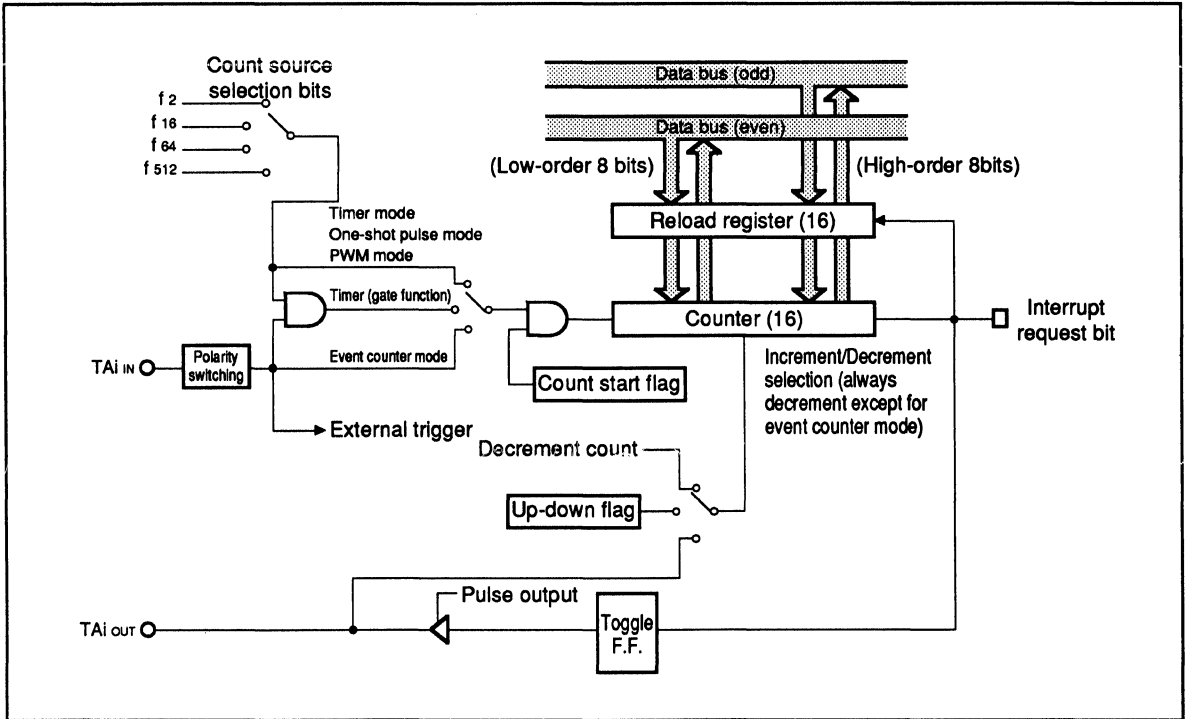


Fig. 2.7.1 Timer Ai block diagram

FUNCTIONAL DESCRIPTION

2.7 Timer A

(1) Counter and reload register (timer Ai register)

Timer Ai counter and reload register consist of 16 bits. The counter counts the selected count source and its contents are decremented by 1 each time a count source is input. In event counter mode, it can also function as an increment counter and its contents are incremented by 1 each time a count source is input. The reload register is used to store the initial value of the counter. The contents of the reload register are reloaded into the counter when the counter underflows (or when it overflows in case used as an increment counter in event counter mode).

The contents of the counter change each time a count source is input, but the contents of the reload register remain unchanged.

Values are stored in the counter and reload register by writing to the timer Ai register.

Table 2.7.1 shows the memory allocation of timer Ai register.

The value written in timer Ai register when the timer is not operating is stored in the counter and reload register. The value written in timer Ai register when the timer is operating is stored only in the reload register. In this case, the updated value is transferred to the counter during next reload. If the timer Ai register is read, the result depends on the operating mode. Table 2.7.2 shows the results of the timer Ai register read and write.

The value of the timer Ai register is undefined at reset. Therefore, at first the counter and the reload register must be initialized when using timer Ai.

Table 2.7.1 Timer Ai register memory allocation

Timer Ai register	High-order byte	Low-order byte
Timer A0 register	Address 47 ₁₆	Address 46 ₁₆
Timer A1 register	Address 49 ₁₆	Address 48 ₁₆
Timer A2 register	Address 4B ₁₆	Address 4A ₁₆
Timer A3 register	Address 4D ₁₆	Address 4C ₁₆
Timer A4 register	Address 4F ₁₆	Address 4E ₁₆

Table 2.7.2 Timer Ai register read and write

Mode	Read	Write
Timer mode	Read timer counting value	<Timer operating> Write to reload register
Event counter mode		<Timer halted> Write to reload register and counter
One-shot pulse mode	Read undefined value	
Pulse width modulation mode		

FUNCTIONAL DESCRIPTION

2.7 Timer A

(2) Count start flag

The count start flag (address 40₁₆) separately controls starting/stopping of each timer. Each bit corresponds to one of the timers.

When this flag is set to "1", a count source is input to the counter. When this flag is set to "0", a count source input to the counter is disabled.

Figure 2.7.2 shows the structure of the count start flag.

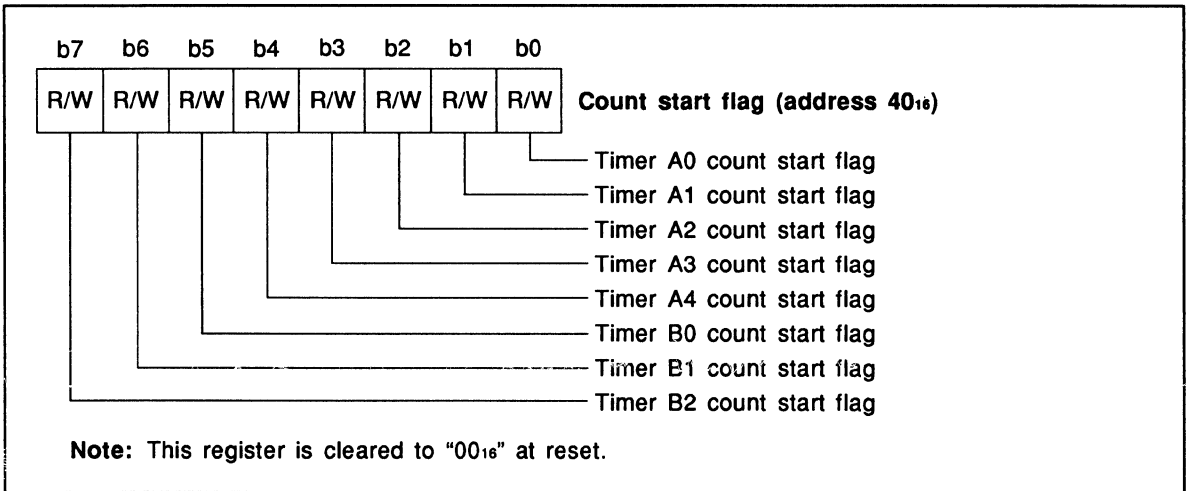


Fig. 2.7.2 Count start flag register structure

(3) One-shot start flag

The one-shot start flag (address 42₁₆) controls the occurrence of an internal trigger used in one-shot pulse mode. When a corresponding one-shot start flag to each timer is set to "1" in case internal trigger is selected, an internal trigger starting one-shot pulse output is occurred. Refer to section "2.7.5 One-shot pulse mode" for more information. One-shot start flag is a write-only flag.

Figure 2.7.3 shows the structure of the one-shot start flag.

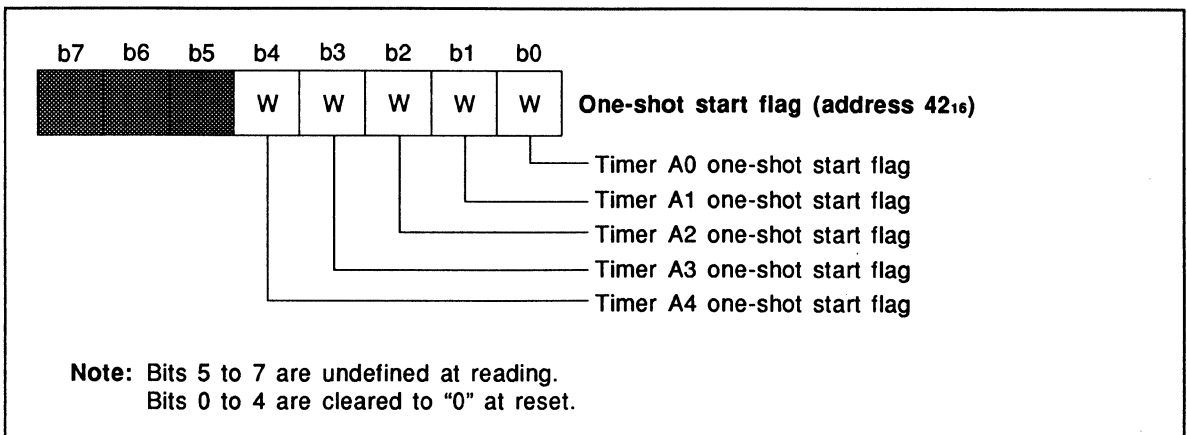


Fig. 2.7.3 One-shot start flag register structure

FUNCTIONAL DESCRIPTION

2.7 Timer A

(4) Up-down flag

The up-down flag (address 44_{16}) is a register consisting of up-down flags and two-phase signal processing selection bits used in event counter mode.

Figure 2.7.4 shows the structure of the up-down flag register followed by a description of each bit.

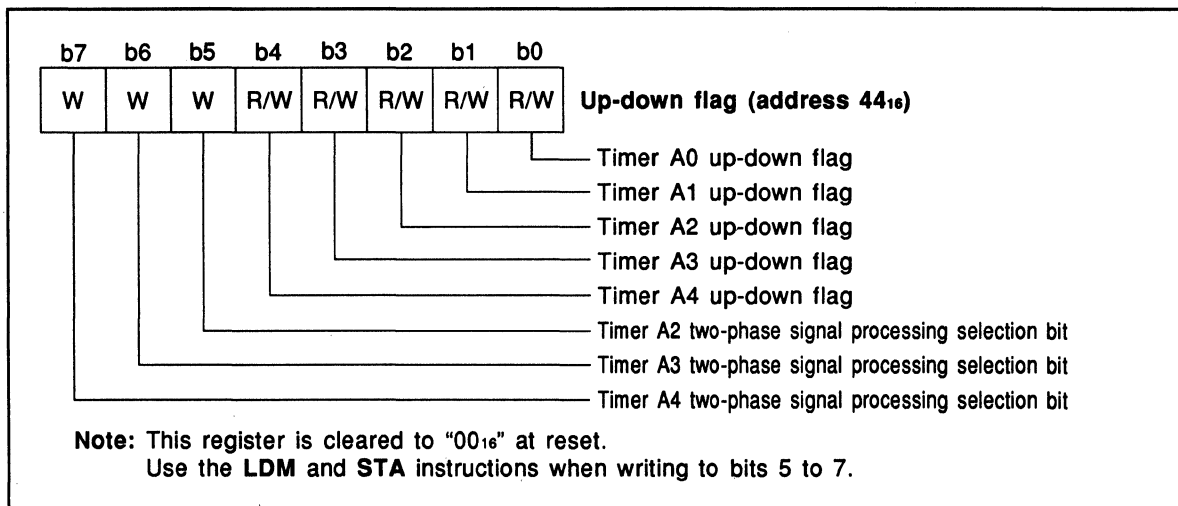


Fig. 2.7.4 Up-down flag register structure

●Timer Ai up-down flags (bits 0 to 4)

These flags are valid in event counter mode when the count up-down flag is selected for up-down switching factor of counter. When this flag is "0", the counter of the corresponding timer is decremented. When this flag is "1", the counter of the corresponding timer is incremented.

●Two-phase signal processing selection bits (bits 5 to 7)

In event counter mode, these bits select the two-phase pulse signal processing function which controls the counter using two-phase pulse with their phases shifted by 90 degrees. This is a write-only bit. The corresponding timer operates with the two-phase pulse signal processing when this bit is set to "1". This bit must be set to "0" when the two-phase pulse signal processing function is not used and in modes other than event counter mode. It is cleared to "0" at reset.

Use the LDM and STA instructions to write to this bit. Do not use the SEB and CLB instructions.

(5) Timer Ai mode register

The timer Ai mode register (addresses 56_{16} to $5A_{16}$) consists of operating mode selection bits, count source selection bits, and timer function selection bits.

Figure 2.7.5 shows the structure of the timer Ai mode register. The operating mode selection bits and count source selection bits are described below. The functions of bits 2 to 5 depend on the operating mode and are described under the description of each operating mode.

FUNCTIONAL DESCRIPTION

2.7 Timer A

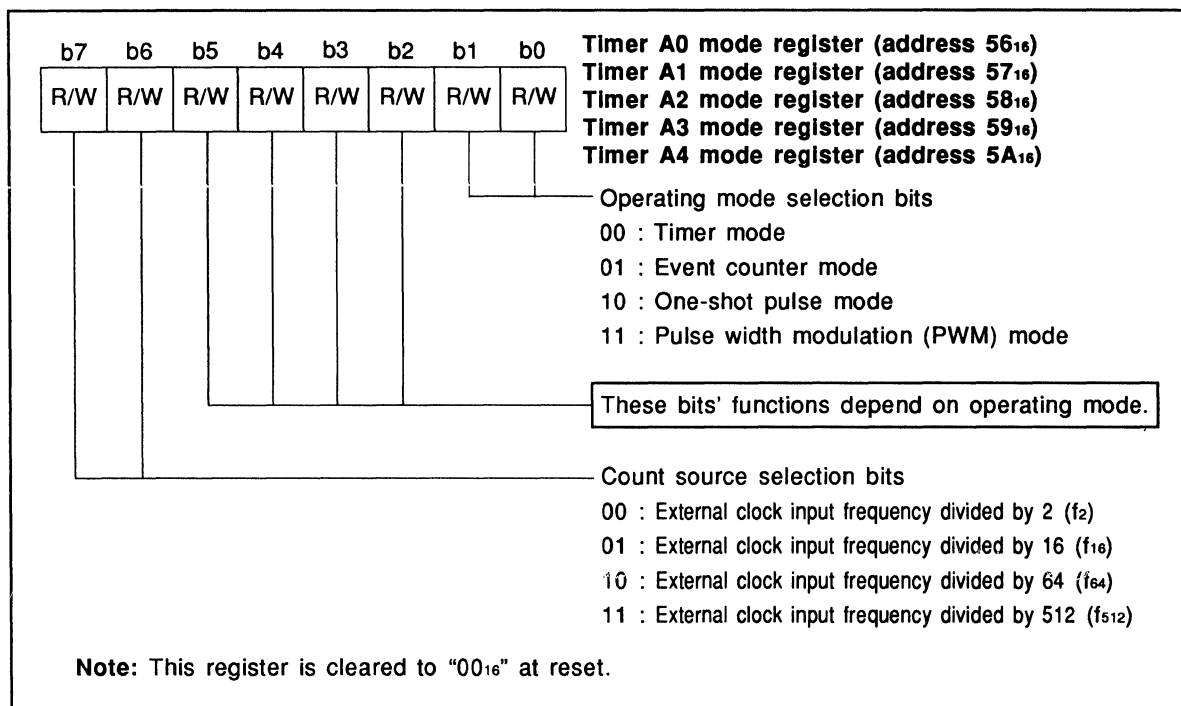


Fig. 2.7.5 Timer AI mode register structure

● Operating mode selection bits (bits 0 and 1)

The operating mode selection bits are used to select the timer operating mode. Table 2.7.3 shows the relationship between the operating mode selection bits and the timer operating modes.

Table 2.7.3 Relationship between operating mode selection bits and operating modes

b1	b0	Operating mode
0	0	Timer mode
0	1	Event counter mode
1	0	One-shot pulse mode
1	1	Pulse width modulation (PWM) mode

● Count source selection bits (bits 6 and 7)

The count source selection bits are used to select the count source. Table 2.7.4 shows the relationship between the count source selection bits and the timer count sources.

These bits are ignored in event counter mode.

Table 2.7.4 Relationship between count source selection bits and count sources

b7	b6	Timer count source	Input clock to the counter		
			$f(X_{IN})=8\text{MHz}$	$f(X_{IN})=16\text{MHz}$	$f(X_{IN})=25\text{MHz}$
0	0	External clock input frequency divided by 2 (f_2)	4MHz	8MHz	12.5MHz
0	1	External clock input frequency divided by 16 (f_{16})	500kHz	1MHz	1.5625MHz
1	0	External clock input frequency divided by 64 (f_{64})	125kHz	250kHz	390.625kHz
1	1	External clock input frequency divided by 512 (f_{512})	15625Hz	31250Hz	48.8281kHz

$f(X_{IN})$: External clock input frequency

FUNCTIONAL DESCRIPTION

2.7 Timer A

(6) Timer Ai Interrupt control register

The timer Ai interrupt control register (addresses 75₁₆ to 79₁₆) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.7.6 shows the structure of the timer Ai interrupt control register. The function of each bit is described below. Use the **SEB** and **CLB** instructions when setting the timer Ai interrupt control register. Refer to section "2.6 Interrupts" for more information.

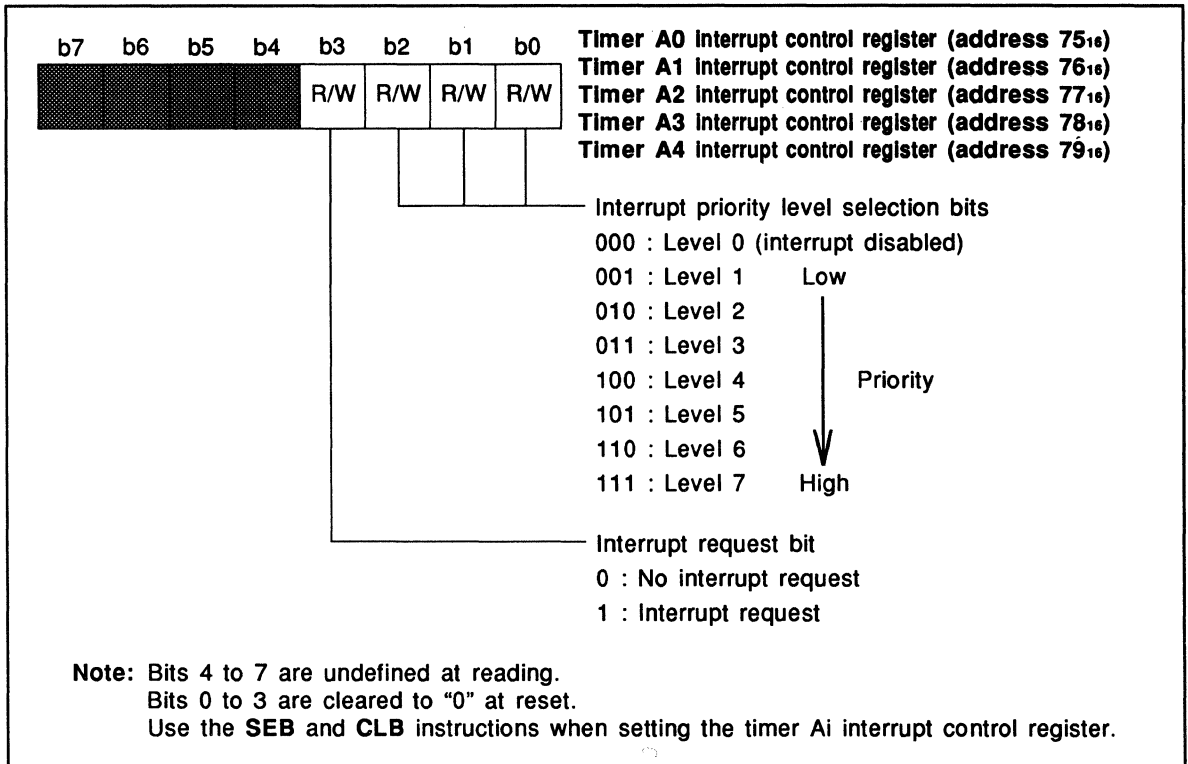


Fig. 2.7.6 Timer Ai Interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using a timer Ai interrupt. When an interrupt request occurs, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is higher than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable only timer Ai interrupt.

●Interrupt request bit (bit 3)

This bit is set to "1" when a timer Ai interrupt request occurs. The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted.
 This bit can be set or cleared by program.

FUNCTIONAL DESCRIPTION

2.7 Timer A

(7) Ports P5 and P6 direction registers

The I/O pins of timers A0 to A3 are shared with port P5 and the I/O pins of timer A4 are shared with port P6. When using these ports as timer input pins, the corresponding bit in the direction register must be set to "0" (input mode). When using these ports as timer output pins, these ports function as timer output pins regardless of the contents of the direction register.

Figure 2.7.7 shows the relationship between port P5 direction register (address 0D₁₆) and port P6 direction register (address 10₁₆) and timer pins.

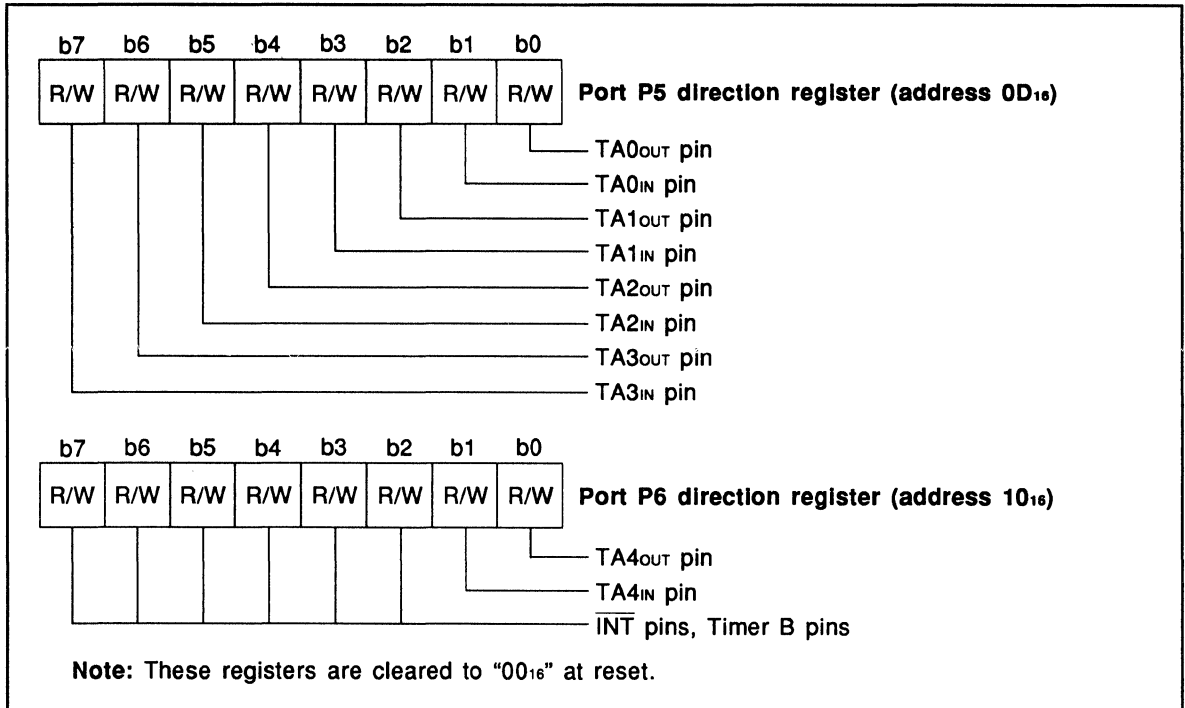


Fig. 2.7.7 Relationship between port P5 and P6 direction registers and timer pins

FUNCTIONAL DESCRIPTION

2.7 Timer A

2.7.3 Timer mode [timer Ai mode register bits 1, 0="00"]

The timer mode is selected by setting the timer Ai mode register bits 1 and 0 to "00". When this mode is selected, bit 5 of the timer Ai mode register must be fixed to "0". Figure 2.7.8 shows the structure of the timer Ai mode register in timer mode.

In timer mode, the selected internal clock is decremented and an interrupt request occurs each time the counter underflows (the contents of the counter reach $0000_{16} \rightarrow$ reloaded value "n"). The timer dividing ratio is expressed as follows:

$$\text{Timer dividing ratio} = 1/(n+1)$$

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

The following functions can be selected with the timer Ai mode register.

●Gate function

Controls count with the input signal to the TAI_{IN} pin.

●Pulse output function

Outputs from the TAI_{OUT} pin, a signal that changes polarity each time the counter underflows.

Timer mode of timers A0 and A2 is also used in pulse output port mode. Refer to section "2.9 Pulse output function" for more information.

(1) Timer mode operation

First, select the operating mode, pulse output function, gate function, and count source with the timer Ai mode register. Next, write an arbitrary value "n" ($n=0000_{16}$ to $FFFF_{16}$) in the timer Ai register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register. When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and starts the count operation.

Figure 2.7.9 shows the setting example of the timer mode related registers.

The contents of the counter are decremented by 1 each time the count source is input. When the counter underflows, the contents of the reload register are reloaded into the counter and the interrupt request bit is set to "1".

Count operation continues in this manner. The interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, an interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program.

The contents of the counter can be read at any timing by reading the contents of the timer Ai register but the contents of the reload register can not be read. In order to change the timer dividing ratio when the timer is operating, a 16-bit update value must be written simultaneously in the timer Ai register. This value is stored in the reload register, and reloaded into the counter at the next reload timing after writing to timer Ai register.

Figure 2.7.10 shows the timer mode operation diagram (neither pulse output nor gate function).

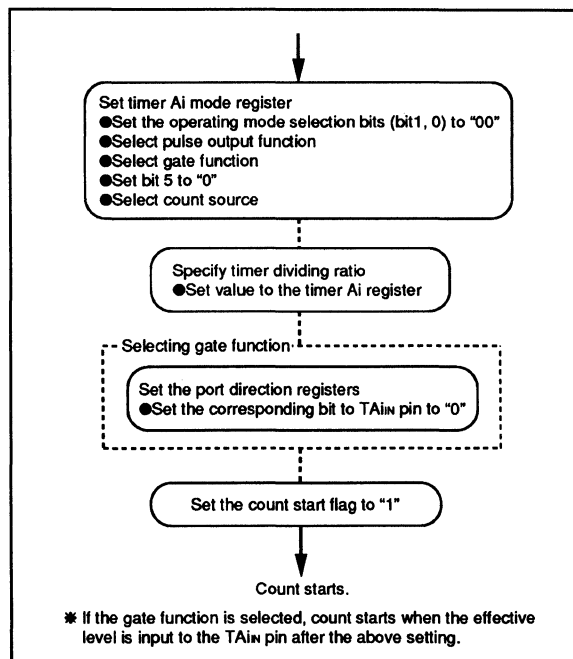


Fig. 2.7.9 Setting example of the timer mode related registers

FUNCTIONAL DESCRIPTION

2.7 Timer A

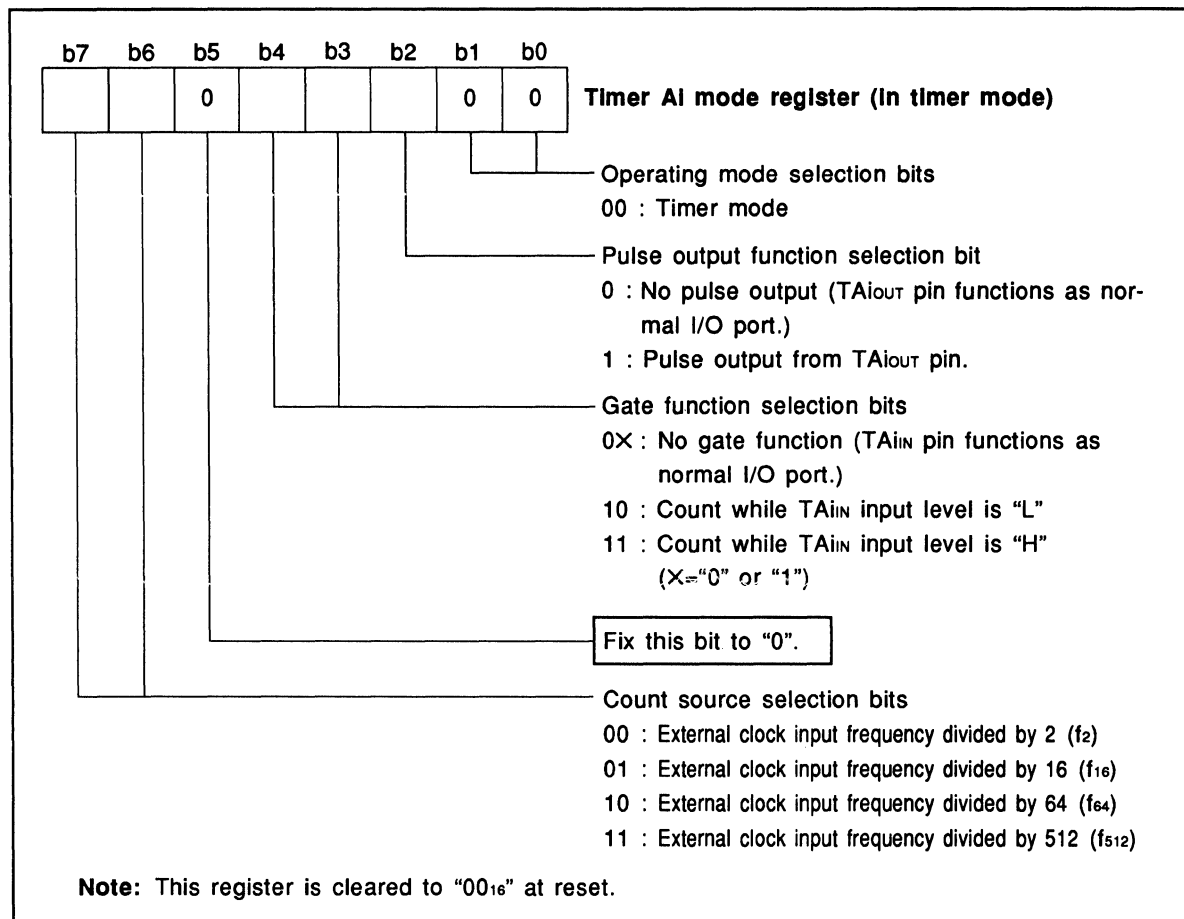


Fig. 2.7.8 Timer AI mode register structure in timer mode

FUNCTIONAL DESCRIPTION

2.7 Timer A

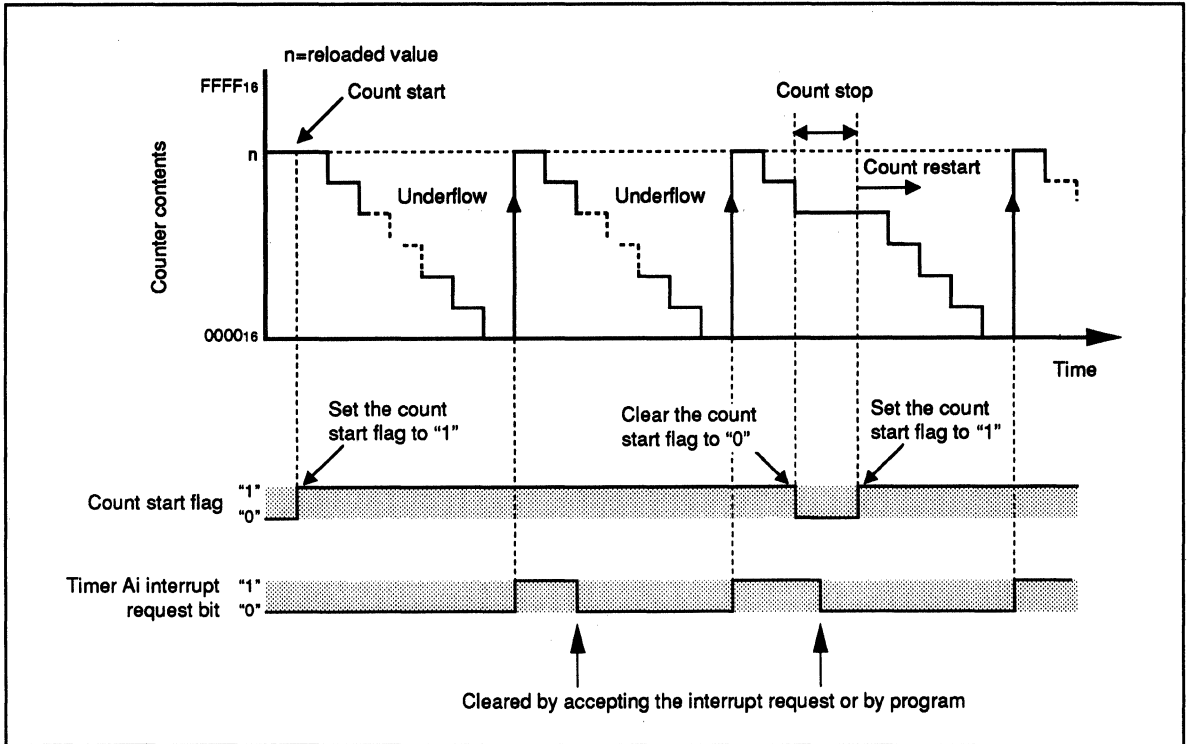


Fig. 2.7.10 Timer mode operation diagram (neither pulse output nor gate function)

[Selection function]

Program selectable gate function and pulse output function are described below. These functions can be used together.

●Gate function

The gate function is selected by setting the gate function selection bits of the timer Ai mode register to "10" or "11". The gate function controls the starting and stopping of the timer count by the level of the input signal to the TA_{iIN} pin. Table 2.7.5 shows the relationship between the gate function selection bits and the count effective level.

When using the gate function, the corresponding port direction register bit to the TA_{iIN} pin must be set to "0" for input mode.

Table 2.7.5 Relationship between gate function selection bits and effective level

Gate function selection bits		Effective level
b4	b3	
1	0	Count while the input level to the TA_{iIN} pin is "L"
1	1	Count while the input level to the TA_{iIN} pin is "H"

FUNCTIONAL DESCRIPTION

2.7 Timer A

When the gate function is selected, counting is performed when count source is input to the counter while the input level to the TAI_{IN} pin is effective and the count start flag is "1". No count source is input and the count stops while the input level is not effective. The contents of the counter are preserved when the TAI_{IN} pin input level changes from effective to non-effective. Therefore, counting can resume when the input level returns to an effective level.

The pulse width of the TAI_{IN} pin input signal during count interval and count halt interval must be at least 2 cycles of the selected count source.

Figure 2.7.11 shows the timer operation example when the gate function is selected.

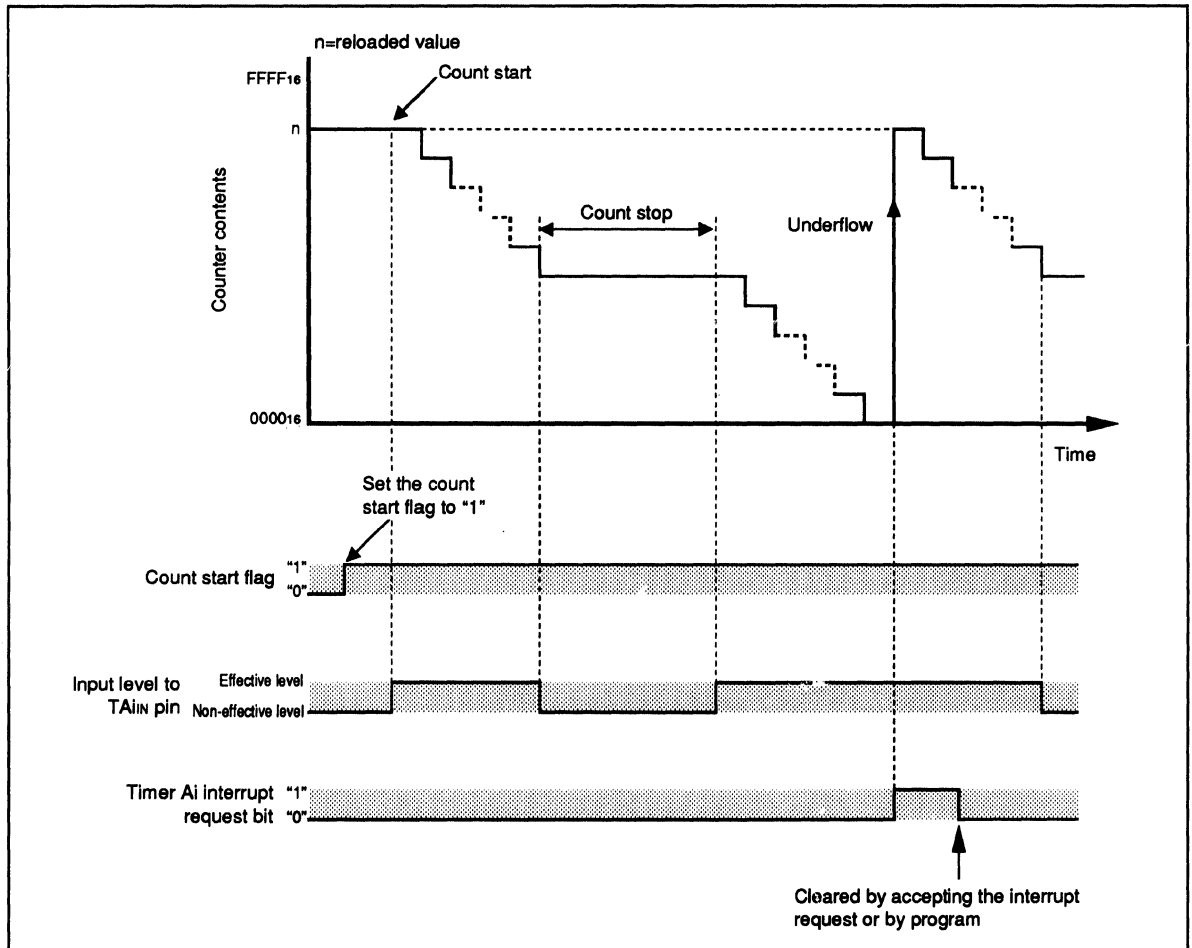


Fig. 2.7.11 Timer operation example when the gate function is selected

FUNCTIONAL DESCRIPTION

2.7 Timer A

●Pulse output function

The pulse output function is selected by setting the pulse output function selection bit to "1". When the pulse output function is selected, a signal that changes polarity each time the counter underflows is output from the TAI_{OUT} pin. The TAI_{OUT} pin is shared with ports P5 and P6. When the pulse output function is selected, the corresponding port is forced to output mode and functions as the TAI_{OUT} pin regardless of the contents of the port direction register. It can be used as a programmable I/O port once the pulse output function selection bit is set to "0".

When selecting the pulse output function, "L" level is output from the TAI_{OUT} pin while the count start flag is "0" and count halted. Therefore, the initial level of the pulse output is always "L".

Figure 2.7.12 shows the timer operation example when the pulse output function is selected.

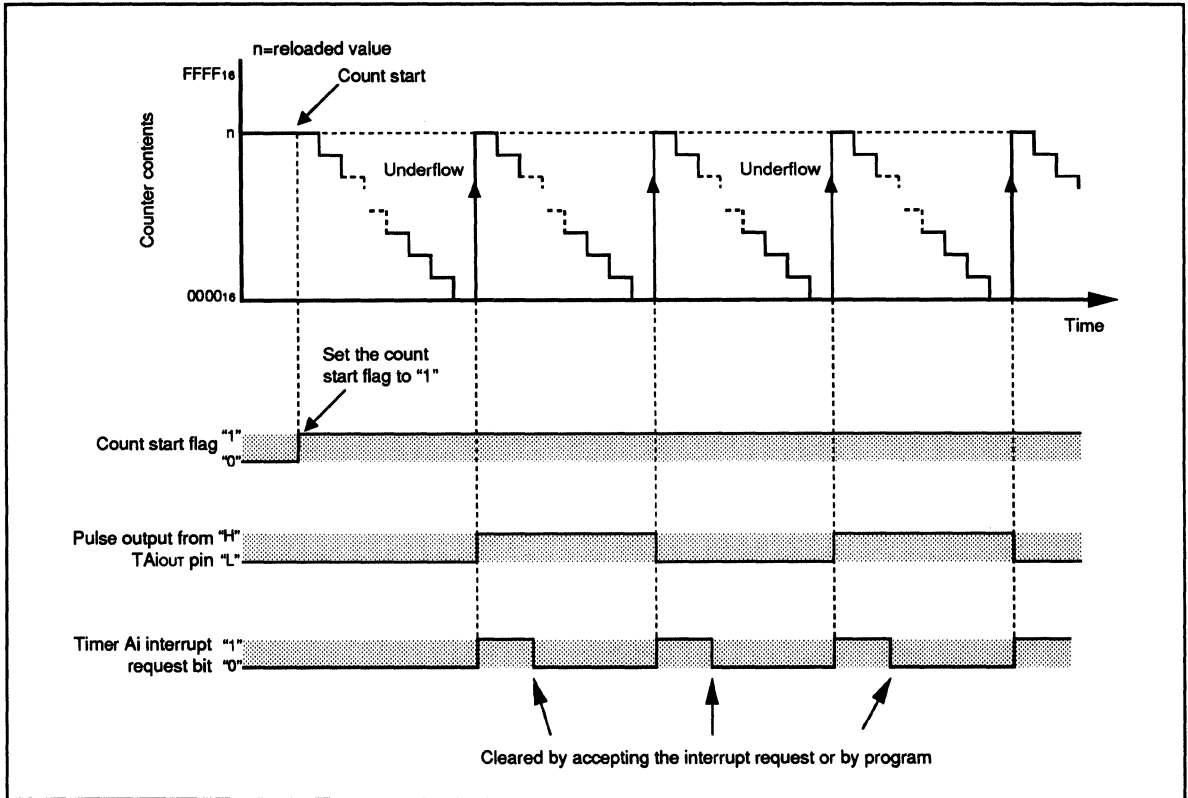


Fig. 2.7.12 Timer operation example when the pulse output function is selected

FUNCTIONAL DESCRIPTION

2.7 Timer A

[Precautions when using timer mode]

1. The value of the counter while operating can be read at any timing by reading the timer Ai register. However, if it is read at the reload timing shown in Figure 2.7.13, "FFFF₁₆" is read instead of the reloaded value. The reloaded value is read if it is read after the timer Ai register is set during timer halted and before the count source is input and count started.

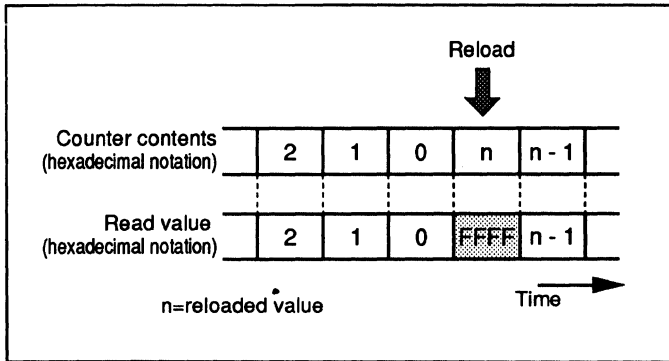


Fig. 2.7.13 Reading the timer Ai register

2.7.4 Event counter mode [timer Ai mode register bits 1, 0="01"]

The event counter mode is selected by setting the timer Ai mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, bit 5 of the timer Ai mode register must be fixed to "0". Figure 2.7.14 shows the structure of the timer Ai mode register in event counter mode.

In event counter mode, the external clock input to the TAI_{IN} pin is counted. The counter can be either an increment counter or a decrement counter according to the up-down flag or the input level to the TAI_{OUT} pin. Figure 2.7.15 shows the structure of the up-down flag register.

In increment counter, an interrupt request occurs when the counter overflows (the contents of the counter reach $FFFF_{16} \rightarrow$ reloaded value "n"). In decrement counter, an interrupt request occurs when the counter underflows (the contents of the counter reach $0000_{16} \rightarrow$ reloaded value "n"). The timer dividing ratio is expressed as follows:

● **Increment counter** Timer dividing ratio = $1/(FFFF_{16}-n+1)$

● **Decrement counter** Timer dividing ratio = $1/(n+1)$

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

In addition, in this mode, the pulse output function and two-phase pulse signal processing function can be selected by program. Two-phase pulse signal processing function is available only with timers A2, A3, and A4.

● **Pulse output function**

A signal that changes polarity is output from the TAI_{OUT} pin each time the counter underflows (decrement counter) or overflows (increment counter).

● **Two-phase pulse signal processing function (timers A2 to A4)**

Whether to increment or decrement the counter is selected by inputting two-phase pulse with phase shifted 90 degrees.

FUNCTIONAL DESCRIPTION

2.7 Timer A

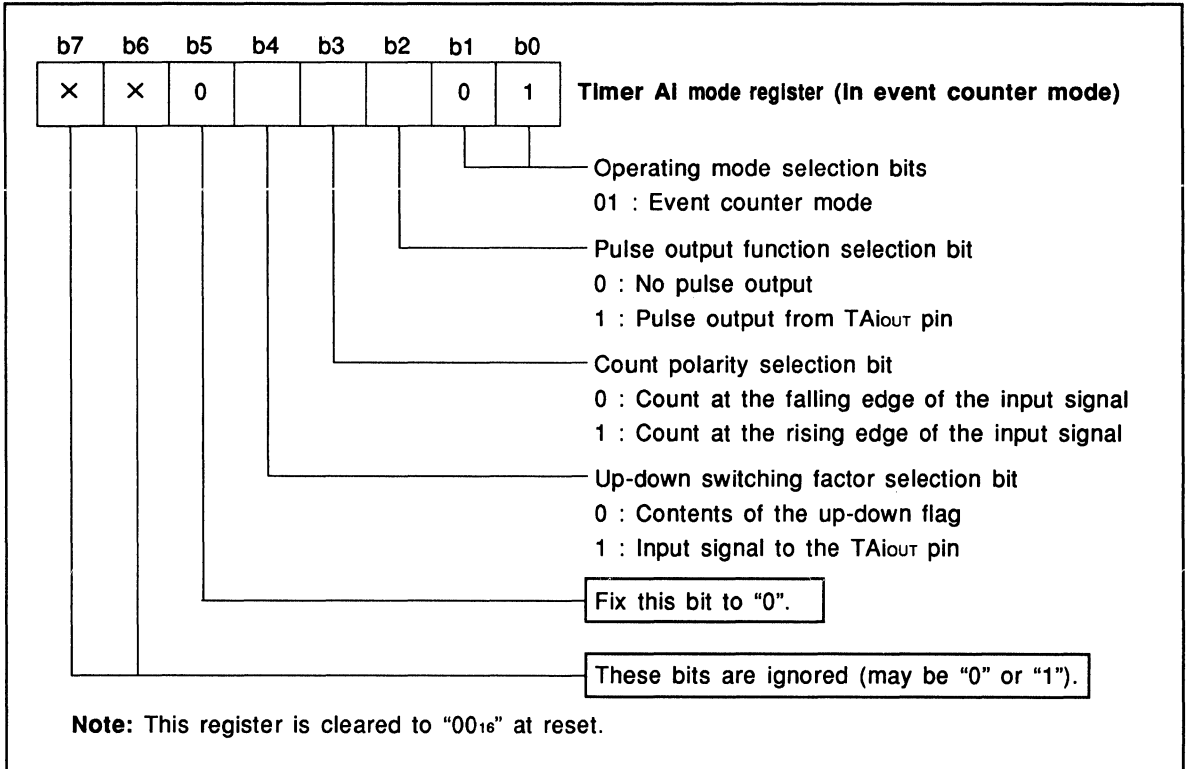


Fig. 2.7.14 Timer A1 mode register structure in event counter mode

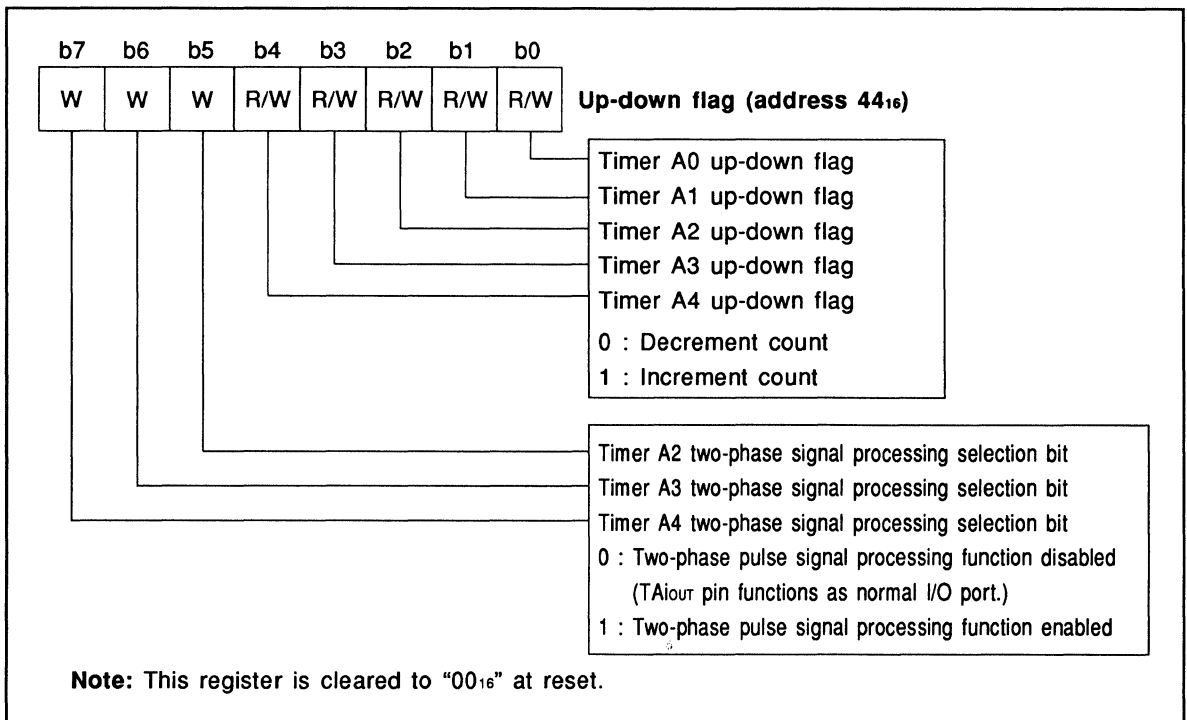


Fig. 2.7.15 Up-down flag register structure

(1) Event counter mode operation

First, select the operating mode, count polarity, pulse output function, and up-down switching factor with the timer Ai mode register. Next, write an arbitrary value "n" ("n"=0000₁₆ to FFFF₁₆) in the timer Ai register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register. Also set the port direction register bit corresponding to the TAI_{IN} pin to "0" (input mode).

When the count start flag is set to "1" (count enabled), the external clock input to the TAI_{IN} pin is input to the counter. The counter operates at the falling edge (when the count polarity selection bit is "0") or rising edge (when the count polarity selection bit is "1") of the input clock.

Whether to increment or decrement the counter can be selected with the up-down flag or the level of the input signal to the TAI_{OUT} pin. The contents of the up-down flag are used if the up-down switching factor selection bit is "0", and the level of the input signal to the TAI_{OUT} pin is used if it is "1".

Figure 2.7.16 shows the setting example of the event counter mode related registers.

●When switching with the contents of the up-down flag

When the corresponding bit to the up-down flag is set to "0", the counter is decremented. When the corresponding bit to the up-down flag is set to "1", the counter is incremented.

●When switching with the input signal to TAI_{OUT} pin

When the input level to the TAI_{OUT} pin is "L", the counter is decremented. When the input level to the TAI_{OUT} pin is "H", the counter is incremented.

The TAI_{OUT} pins are shared with port pins. Therefore, the direction register of the corresponding port pin must be set to "0" (input mode) when the input level of the TAI_{OUT} pin is used to control increment/decrement.

The pulse output function described later cannot be used when the input signal to the TAI_{OUT} pin is used to control increment/decrement.

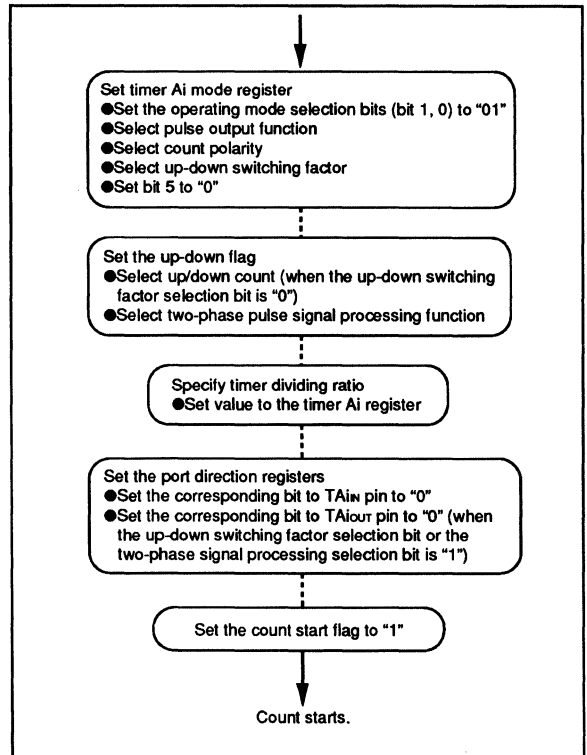


Fig. 2.7.16 Setting example of the event counter mode related registers

FUNCTIONAL DESCRIPTION

2.7 Timer A

The count direction can be changed while counting. The count direction changes when the next effective edge of the count source is input after the contents of the up-down flag change if the up-down flag is used for up-down switching factor, and after the input level to the TAI_{OUT} pin changes if the input signal to the TAI_{OUT} pin is used for up-down switching factor.

Count operation continues and the counter underflows (decrement count) or overflows (increment count) at which time an interrupt request occurs and an interrupt request bit is set to "1". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program. The contents of the counter can be read at any timing by reading the contents of timer Ai register, but the contents of the reload register cannot be read. In order to change the timer dividing ratio while the timer is operating, a 16-bit update value must be written simultaneously in the timer Ai register. This value is stored in the reload register, and reloaded into the counter at the next reload timing after writing to timer Ai register.

Figure 2.7.17 shows the event counter mode operation diagram (neither pulse output nor two-phase pulse signal processing function).

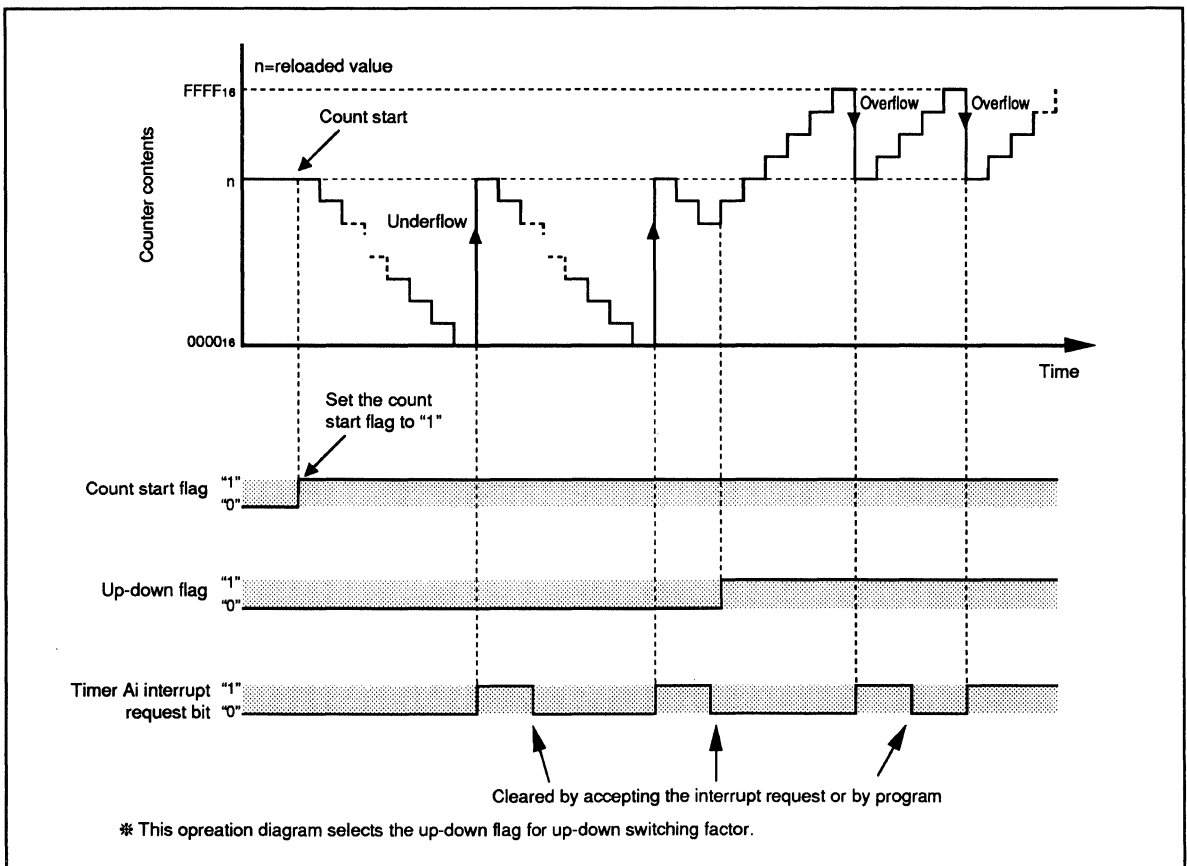


Fig. 2.7.17 Event counter mode operation diagram (when up-down switching factor selection bit is "0")

[Selection function]

Pulse output function and two-phase pulse signal processing function can be selected by program. However, only timers A2, A3, and A4 can use the two-phase pulse signal processing function. The pulse output function and the two-phase pulse signal processing function both use the TAI_{OUT} pin. Therefore, only one of these functions can be used for each timer at any one time.

●Pulse output function

Pulse output function is selected when the pulse output function selection bit is set to "1". When this function is selected, a signal that changes polarity each time the contents of the counter underflow (decrement count) or overflow (increment count) is output from the TAI_{OUT} pin.

The TAI_{OUT} pin is shared with ports P5 and P6. When the pulse output function is selected, the corresponding port is forced to output mode and functions as the TAI_{OUT} pin regardless of the contents of the port direction register. It can be used as a programmable I/O port once the pulse output function selection bit is set to "0".

"L" level is output from the TAI_{OUT} pin while the count start flag is "0" and count disabled. Therefore, the initial level of the pulse output is always "L".

●Two-phase pulse signal processing function (Timers A2 to A4)

Two-phase pulse signal processing function is selected for timers A2 to A4 when the two-phase signal processing selection bit is set to "1". When this function is selected, set the pulse output function selection bit to "0", the count polarity selection bit to "0", and the up-down switching factor selection bit to "1". Figure 2.7.18 shows the timer Aj mode register (j=2 to 4) when two-phase pulse signal processing function is selected.

The TAJ_{OUT} pin is used in input mode and the corresponding port direction register bit must be set to "0".

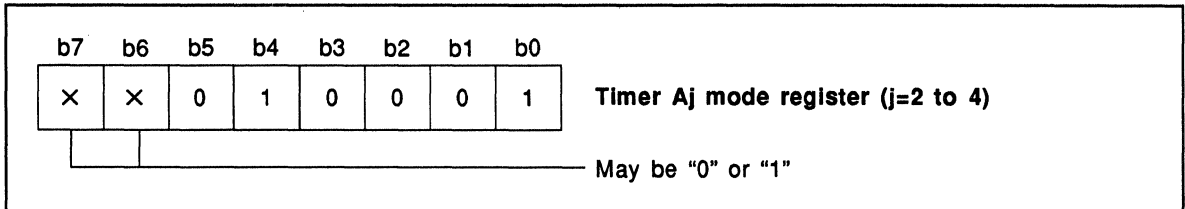


Fig. 2.7.18 Timer Aj mode register (j=2 to 4) when two-phase pulse signal processing function is selected

FUNCTIONAL DESCRIPTION

2.7 Timer A

A timer with two-phase pulse signal processing function selected controls the counter by two-phase pulse with phase shifted by 90 degrees. There are two types of two-phase pulse signal processing operation; one for timers A2 and A3 and another for timer A4 (quadruple processing).

<Timers A2 and A3>

The counter is incremented when the rising edge is input to the TA_{KIN} ($k=2, 3$) pin and decremented when the falling edge is input to the TA_{KIN} pin after the level of the TA_{KOUT} pin changes from "L" to "H". (Refer to Figure 2.7.19)

<Timer A4>

The counter is incremented at every rising and falling edge of the TA_{4OUT} and the TA_{4IN} pins when a phase related pulse with the rising edge input to the TA_{4IN} pin is input after the level of the TA_{4OUT} pin changes from "L" to "H".

The counter is decremented at every rising and falling edge of the TA_{4OUT} and the TA_{4IN} pins when a phase related pulse with the falling edge input to the TA_{4OUT} pin is input after the level of the TA_{4IN} pin changes from "H" to "L". (Refer to Figure 2.7.20)

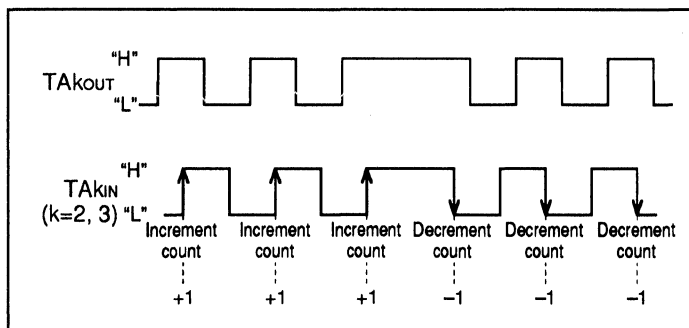


Fig. 2.7.19 Timers A2 and A3 two-phase pulse signal processing operation

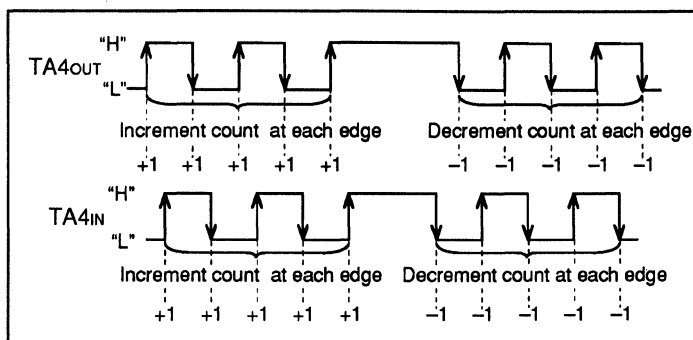


Fig. 2.7.20 Timer A4 two-phase pulse signal processing operation (quadruple processing)

FUNCTIONAL DESCRIPTION

2.7 Timer A

[Precautions when using event counter mode]

- The value of the counter while operating can be read at any timing by reading the timer Ai register. However, if it is read at the reload timing shown in Figure 2.7.21, "FFFF₁₆" is read at underflow and "0000₁₆" is read at overflow instead of the reloaded value. The reloaded value is read if it is read after the timer Ai register is set during timer halted and before the count source is input and count started.

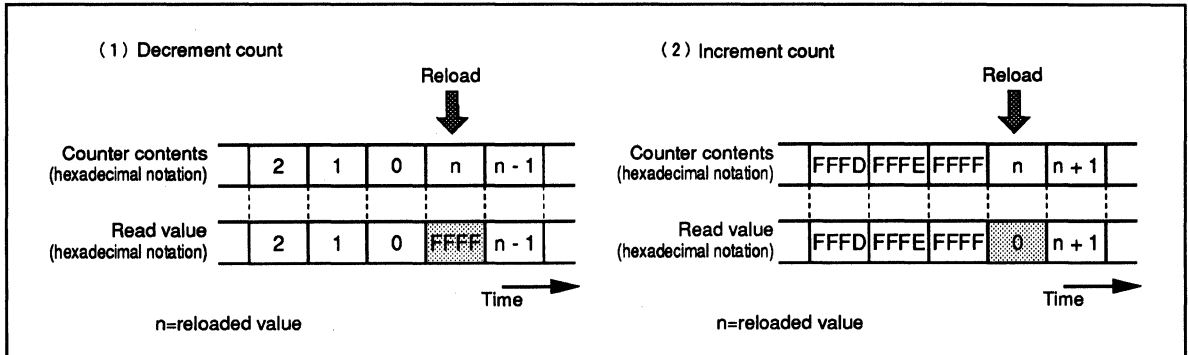


Fig. 2.7.21 Reading the timer Ai register

- All of the following functions uses the TA_{OUT} pin. Only one of these functions can be selected for each timer at any one time.
 - Count control using input signal to TA_{OUT} pin
 - Pulse output function
 - Two-phase pulse signal processing function (timers A2 to A4)

- The phase difference of the two-phase pulse used for two-phase pulse processing function (input clocks to TA_{JOUT} and TA_{JIN} pins) must be the characteristics shown in Figure 2.7.22.

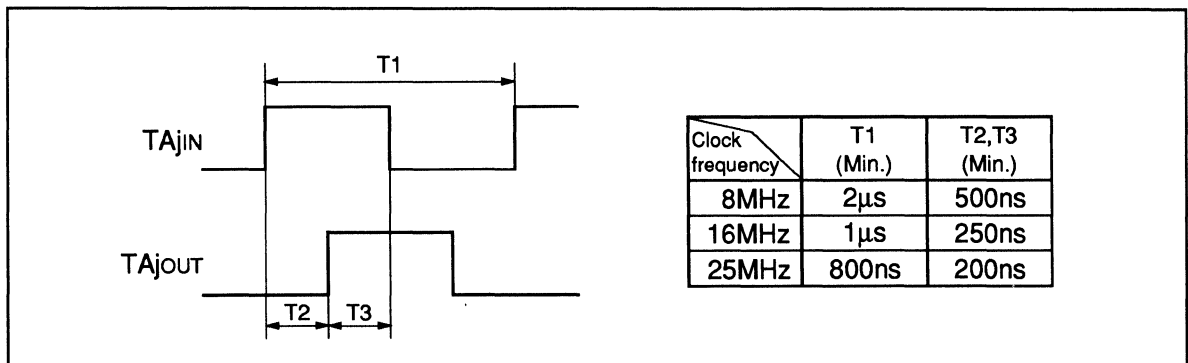


Fig. 2.7.22 The characteristics of the two-phase pulse signal

FUNCTIONAL DESCRIPTION

2.7 Timer A

2.7.5 One-shot pulse mode [timer Ai mode register bits 1, 0="10"]

The one-shot pulse mode is selected by setting the timer Ai mode register bit 1 to "1" and bit 0 to "0". When this mode is selected, the timer Ai mode register bit 5 must be fixed to "0" and bit 2 must be fixed to "1". Figure 2.7.23 shows the structure of the timer Ai mode register in one-shot pulse mode.

In one-shot pulse mode, "H" level is output for an arbitrary interval from the TAI_{OUT} pin after a trigger.

The trigger can be either an internal trigger or an external trigger. The internal trigger is occurred by writing "1" into the one-shot start flag shown in Figure 2.7.24. The external trigger is occurred by the effective edge of the input signal to the TAI_{IN} pin with count start flag set to "1".

Counting starts with a trigger and at the same time, "H" level is output from the TAI_{OUT} pin. The contents of the counter are decremented by 1 each time a count source is input. When the contents of the counter reach "0001₁₆" → reloaded value "n", the output level from the TAI_{OUT} pin changes to "L" and counting stops. At this point, an interrupt request occurs.

The width of the output pulse ("H" level width) can be expressed as follows:

"H" level width of pulse output = n/f_i [s]

f_i: Selected count source frequency

n: Value set in counter

(0000₁₆ to FFFF₁₆ during count halted)

(0001₁₆ to FFFF₁₆ during count operating)

FUNCTIONAL DESCRIPTION

2.7 Timer A

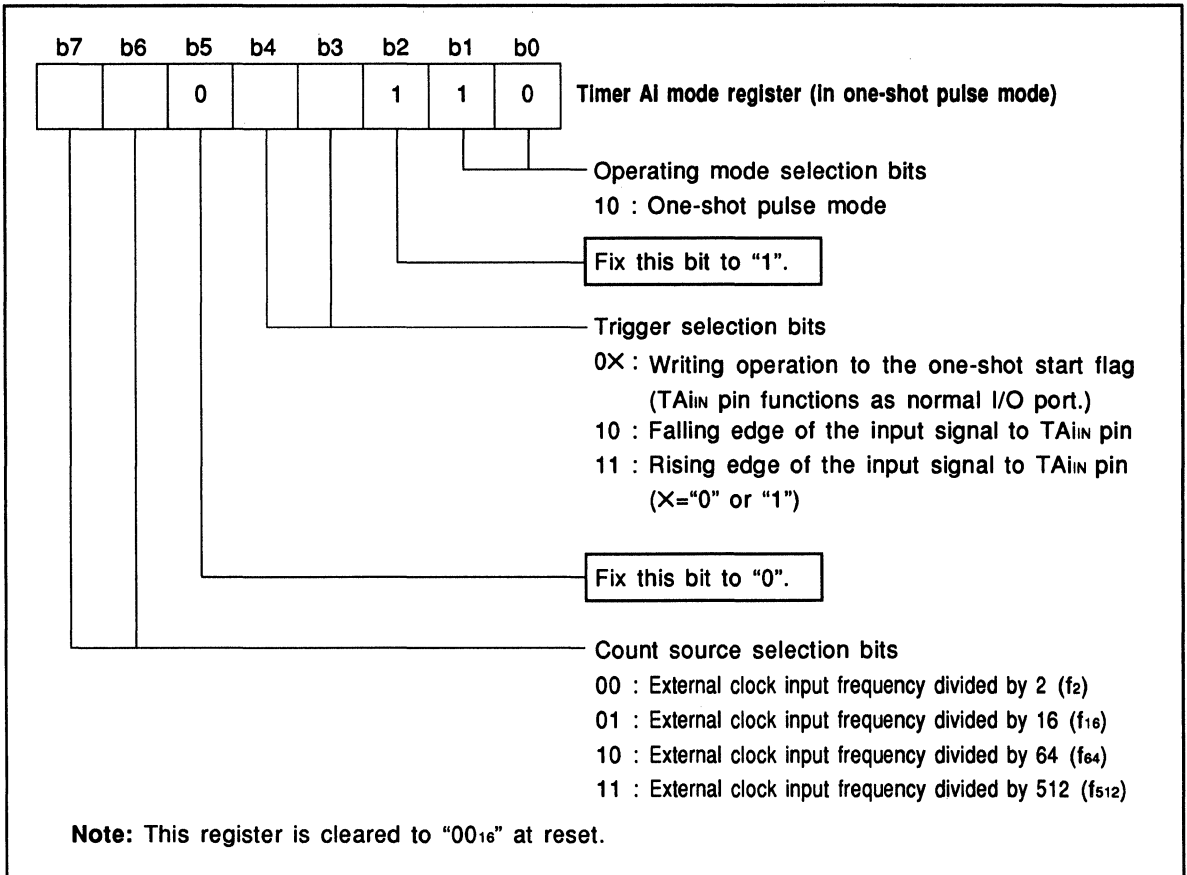


Fig. 2.7.23 Timer A1 mode register structure in one-shot pulse mode

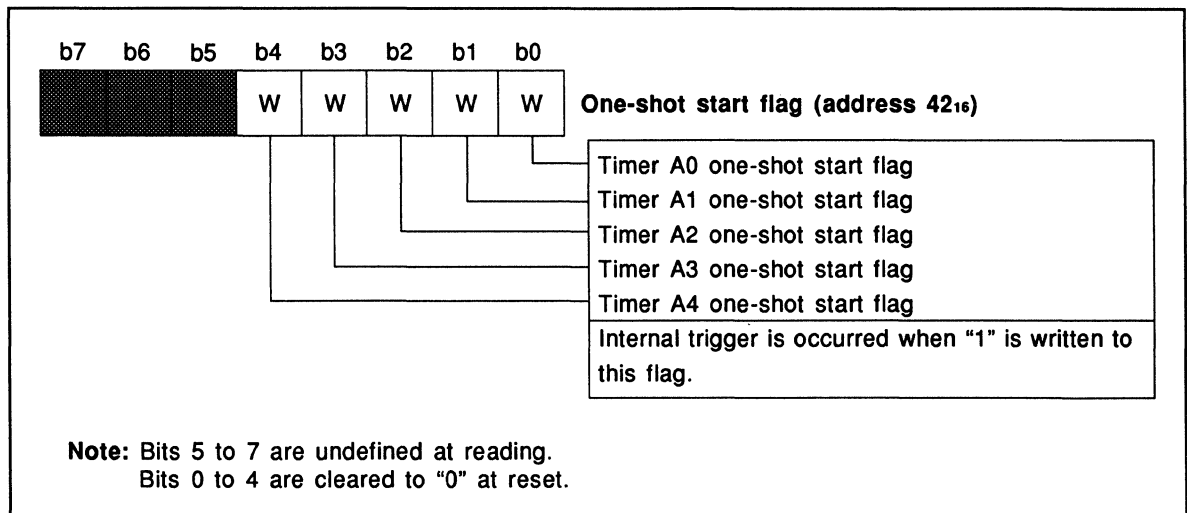


Fig. 2.7.24 One-shot start flag register structure

FUNCTIONAL DESCRIPTION

2.7 Timer A

(1) One-shot pulse mode operation

First select the operating mode, trigger occurrence factor, and count source with the timer Ai mode register. The TAI_{OUT} pin starts to output "L" level when one-shot pulse mode is selected with the operating mode selection bits. Next, write an arbitrary value "n" ("n"=0000₁₆ to FFFF₁₆) in the timer Ai register to set the pulse output width. At this point, "n" is stored in the counter and the reload register. Count is enabled when the count start flag is set to "1", and then count starts when a trigger is occurred. If bit 4 of the timer Ai mode register is "0", an internal trigger is occurred by setting the corresponding bit to each timer in the one-shot start flag to "1". If bit 4 of the timer Ai mode register is "1", an external trigger is selected. When selecting an external trigger, if bit 3 is "0", a trigger is occurred at the falling edge of the TAI_{IN} pin input signal and if bit 3 is "1", a trigger is occurred at the rising edge of the TAI_{IN} pin input signal. When using an external trigger, the corresponding port direction register bit to the TAI_{IN} pin must be set to "0" (input mode).

Figure 2.7.25 shows the setting example of the one-shot pulse mode related registers.

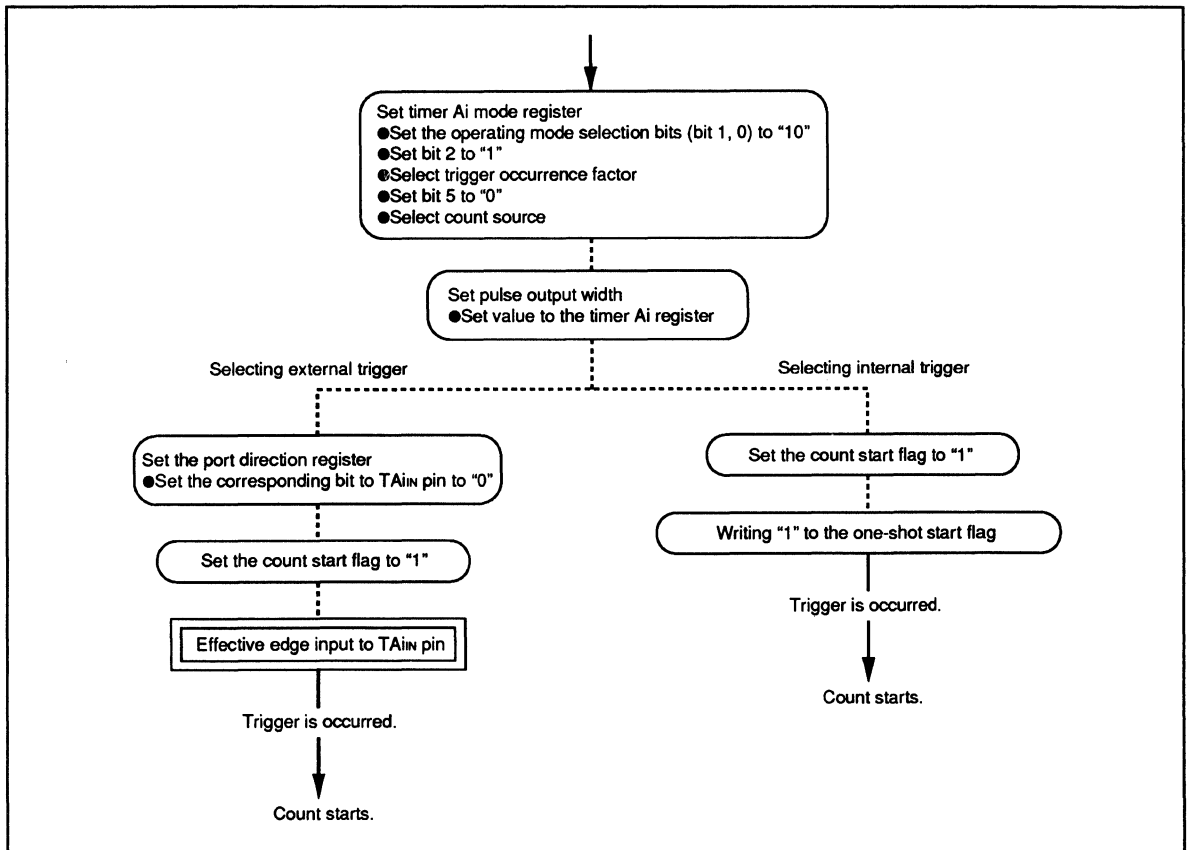


Fig. 2.7.25 Setting example of the one-shot pulse mode related registers

FUNCTIONAL DESCRIPTION

2.7 Timer A

When triggered, counting starts and "H" level is output from the TAI_{OUT} pin. (However, if the timer Ai register contains "0000₁₆", the TAI_{OUT} pin level remains at "L" and counting does not start.) The counter is decremented by 1 each time a count source is input. When the contents of the counter reach 0001₁₆ → "n", the TAI_{OUT} pin level changes to "L" and counting stops. The "H" width of the output pulse from the TAI_{OUT} pin is (count source cycle) × n.

An interrupt request occurs and the interrupt request bit is set to "1" when the TAI_{OUT} pin level changes from "H" to "L". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program.

If a value is written in the timer Ai register while the counter is operating ("H" level is output from the TAI_{OUT} pin), this value is only set in the reload register. It is reloaded into the counter at the next reload timing. Values other than 0000₁₆ can be written while the counter is operating.

If the value of timer Ai register is read, the result is undefined.

Figure 2.7.26 shows the one-shot pulse mode operation diagram (when an external trigger is selected).

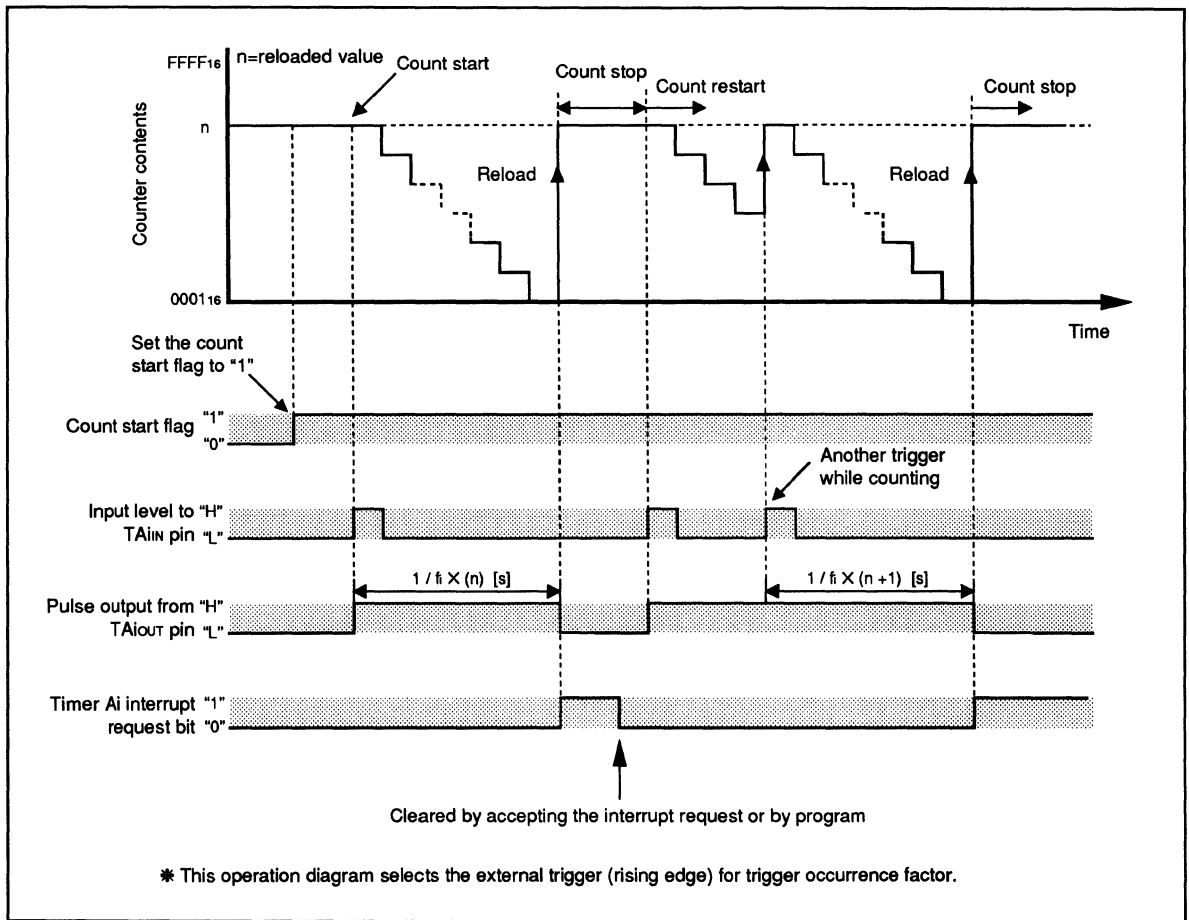


Fig. 2.7.26 One-shot pulse mode operation (when external trigger is selected)

FUNCTIONAL DESCRIPTION

2.7 Timer A

The counter stops after completing one cycle of output, and repeats the same operation when the next trigger is occurred. When the count start flag is "0" (count disabled), "L" level is output from the TAI_{OUT} pin.

If another trigger is occurred while counting, the contents of the reload register are transferred to the counter and decrement continues from that value. In this case, the TAI_{OUT} pin level becomes "L" at n+1 count after the trigger. The contents of the reload register are transferred to the counter only when a trigger is occurred while counting. At least one cycle of the timer count source should elapse before retriggering.

[Precautions when using one-shot pulse mode]

1. If the count start flag is set to "0" while counting, counting stops and the output level of the TAI_{OUT} pin becomes "L". At the same time an interrupt request occurs and the interrupt request bit is set to "1".
2. Values between 0001₁₆ and FFFF₁₆ can be written in the timer Ai register while the counter is operating (While "H" level is output from the TAI_{OUT} pin).
3. When an external trigger is selected, there may be a time lag between the time the effective edge is input to TAI_{IN} pin and the time one-shot pulse is output. This is because the output of one-shot pulse is synchronized with the internal operating clock.
4. When the operating mode is switched described below, the interrupt request bit in timer Ai interrupt control register is set to "1".

- When one-shot pulse mode is selected after removing reset.
- When the operating mode is switched from timer mode to one-shot pulse mode.
- When the operating mode is switched from event counter mode to one-shot pulse mode.

Therefore, the interrupt request bit must be cleared to "0" after above switching modes when using timer Ai interrupt or the interrupt request bit.

FUNCTIONAL DESCRIPTION

2.7 Timer A

2.7.6 Pulse width modulation (PWM) mode [timer Ai mode register bits 1, 0="11"]

Pulse width modulation mode (hereafter referred to as PWM) is selected by setting the timer Ai mode register bits 1 and 0 to "11" and timer Ai functions as a pulse width modulator. When this mode is selected, bit 2 of the timer Ai mode register must be fixed to "1".

Figure 2.7.27 shows the structure of the timer Ai mode register in PWM mode.

In PWM mode, when a trigger occurs, an arbitrary pulse width signal is output continuously from the TAI_{OUT} pin. A 16-bit PWM mode or an 8-bit PWM mode can be selected by program.

●16-bit PWM mode

The counter functions as a 16-bit pulse width modulator.

●8-bit PWM mode

The reload register and the counter are both divided into 8-bit halves. The high-order 8 bits of the counter function as a pulse width modulator and the low-order 8 bits function as a prescaler.

PWM mode of timers A1 and A3 is also used when modulating the pulse width in pulse output port mode. Refer to "2.9 Pulse output function" for more information.

The trigger is either an internal trigger occurred when the count start flag shown in Figure 2.7.28 is set to "1" or an external trigger occurred when the effective edge signal is input to the TAI_{IN} pin with the count start flag set to "1". When a trigger occurs, the pulse width modulator starts and pulses are output from the TAI_{OUT} pin.

An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the pulse output changes from "H" to "L".

When the pulse width modulator starts operation with a trigger, the next trigger is not accepted (pulses are output continuously).

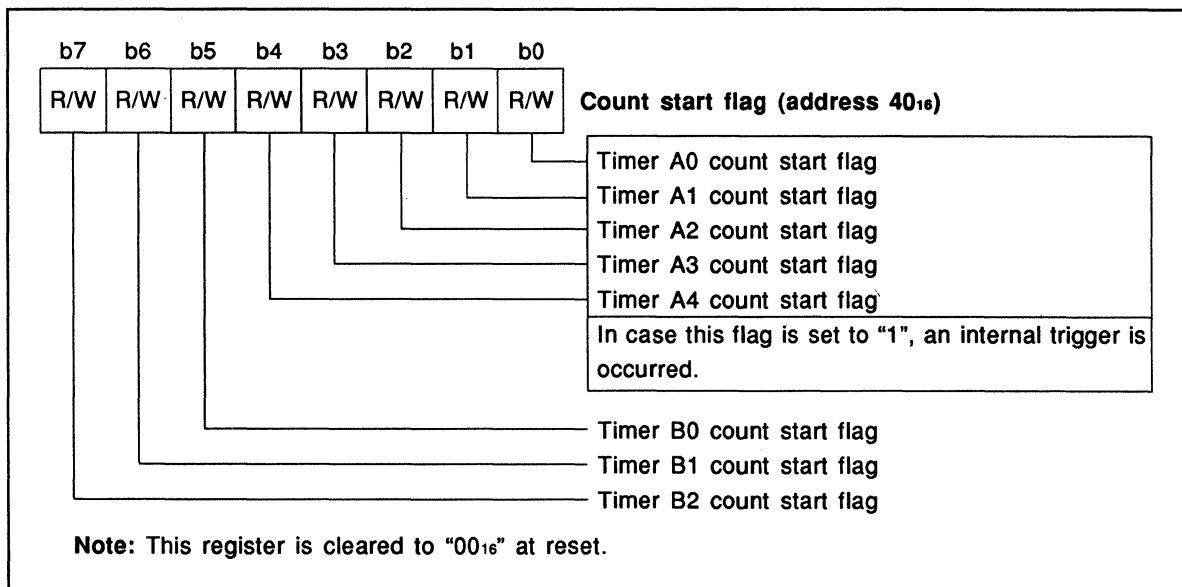


Fig. 2.7.28 Count start flag register structure

FUNCTIONAL DESCRIPTION

2.7 Timer A

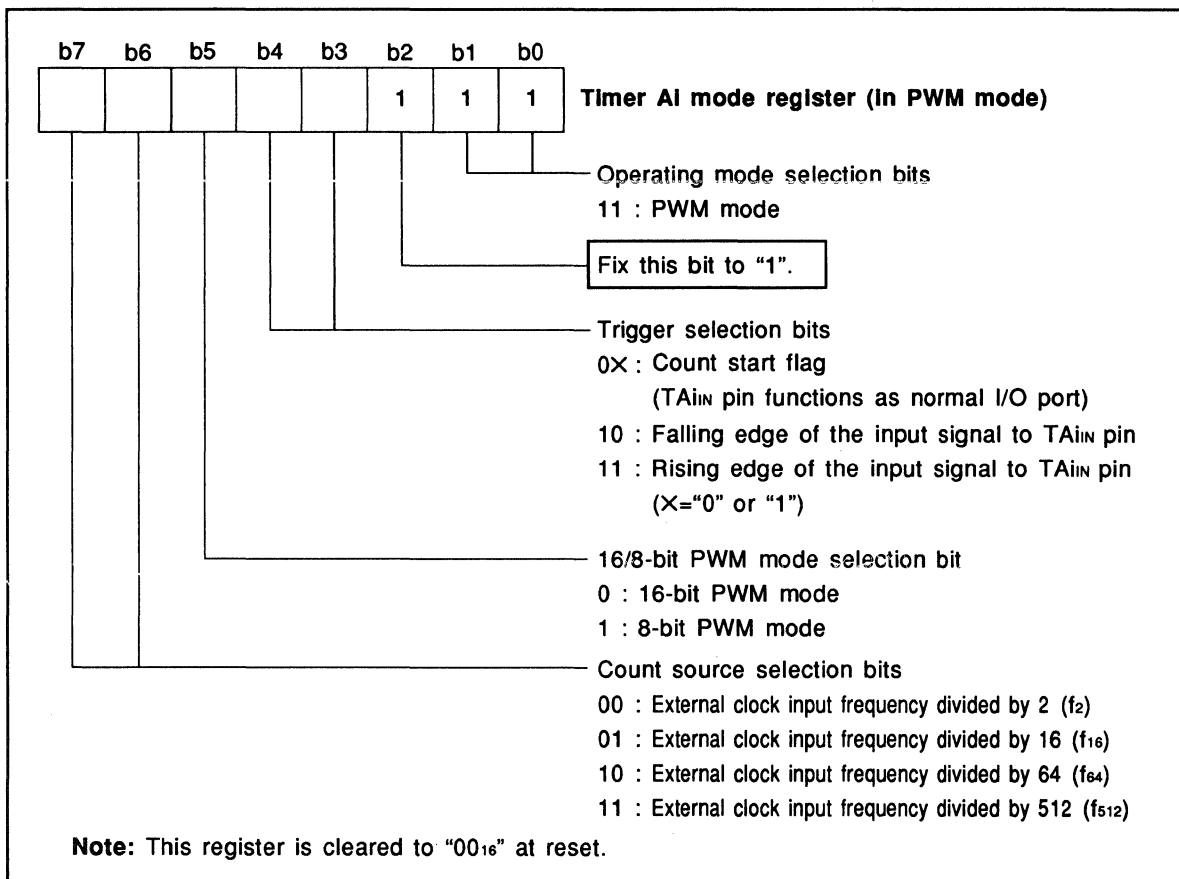


Fig. 2.7.27 Timer AI mode register structure in PWM mode

(1) PWM mode operation

First, select the operating mode, trigger occurrence factor, 16/8-bit PWM mode, and count source with the timer Ai mode register. The TAI_{OUT} pin outputs "L" level when PWM mode is selected with the operating mode selection bits. Next, write an arbitrary value "n" in the timer Ai register to set the pulse output width. At this point, "n" is set in the counter and the reload register. Then when a trigger is occurred, the pulse width modulator starts and pulses are output from the TAI_{OUT} pin.

When bit 4 of the timer Ai mode register is "0", an internal trigger is occurred each time the corresponding bit to each timer in the count start flag is set to "1". If bit 4 of the timer Ai mode register is "1", an external trigger is selected. When selecting an external trigger, if bit 3 is "0", the trigger is occurred at the falling edge of the input signal to the TAI_{IN} pin with count start flag set to "1" and if bit 3 is "1", the trigger is occurred at the rising edge of the input signal to the TAI_{IN} pin with count start flag set to "1". When using an external trigger, set the corresponding port direction register bit to the TAI_{IN} pin to "0" (input mode).

A pulse width modulator continuously outputs pulses when it starts operation. An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the pulse output changes from "H" to "L". Interrupts must be enabled before they can be used. Refer to "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until an interrupt request is accepted or it is cleared by program.

If a value is written in the timer Ai register while the counter is operating, the value is set only in the reload register. It is reloaded into the counter next time the level of the pulse output changes from "L" to "H".

If the value of the timer Ai register is read, the result is undefined.

In order to stop the pulse width modulator, count start flag must be cleared to "0". At the same time, the output level of TAI_{OUT} pin becomes "L".

The 16-bit PWM mode and 8-bit PWM mode operations are described below.

FUNCTIONAL DESCRIPTION

2.7 Timer A

[16-bit PWM mode]

16-bit PWM mode is selected when the 16/8-bit PWM mode selection bit is set to "0". In this mode, the period and width of the pulse output from the TAI_{OUT} pin can be expressed as follows:

$$\text{Pulse output period} = (1/f_i) \times (2^{16}-1) \text{ [s]}$$

$$\text{Pulse output "H" width} = (1/f_i) \times n \text{ [s]}$$

f_i : Frequency of selected count source [Hz]

n : Value set in counter

(value between 0000₁₆ and FFFE₁₆)

Figure 2.7.29 shows an example of an output waveform in 16-bit PWM mode. An interrupt request occurs and the interrupt request bit is set to "1" each time the level of the pulse output signal changes from "H" to "L".

The "H" level width of the pulse output can be changed while the pulse width modulator is operating (outputting the pulse signal) by writing a value in the timer Ai register. This value is written in the reload register and reloaded into the counter next time the level of the pulse output changes from "L" to "H". Therefore, the pulse width changes from the pulse period following the pulse period being output when the value was written.

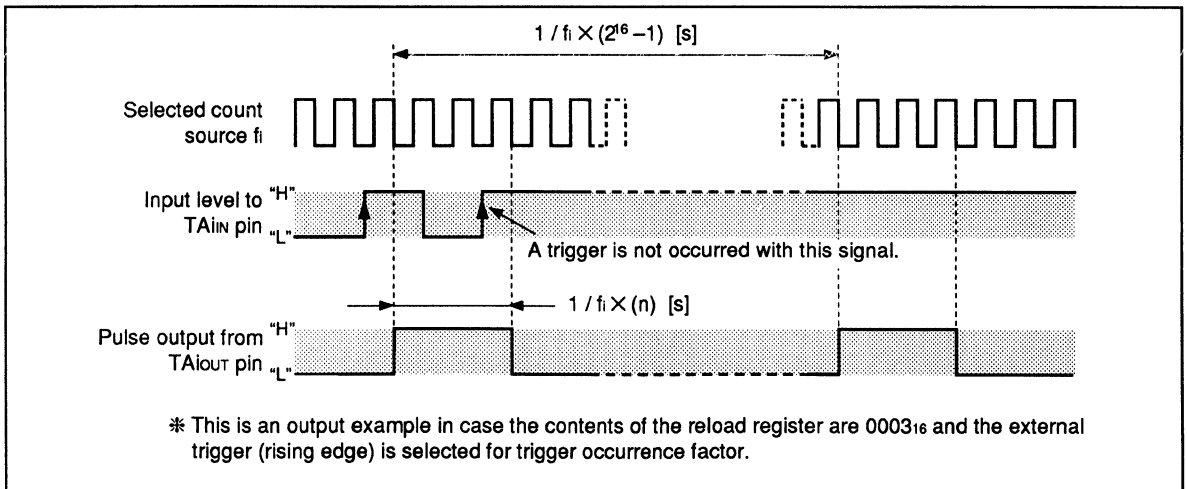


Fig. 2.7.29 16-bit PWM mode output waveform example

[8-bit PWM mode]

8-bit PWM mode is selected when the 16/8-bit PWM mode selection bit is set to "1". In this mode, the reload register and the counter are divided into 8-bit halves. The high-order 8 bits of the counter function as an 8-bit pulse width modulator and the low-order 8 bits function as a prescaler.

The prescaler counts the clock selected with the count source selection bit. The prescaler is decremented and an underflow signal is occurred when its contents reach 00₁₆ → "m" ("m" = value set in low-order 8 bits of the reload register). The contents of the reload register are reloaded into the prescaler and counting continues. The pulse width modulator (high-order 8 bits of the counter) counts the underflow signal occurred by the prescaler. The period and width of the pulse output from the TAI_{OUT} pin can be expressed as follows:

$$\text{Pulse output period} = (1/f_i) \times (m+1) \times (2^8-1) \text{ [s]}$$

$$\text{Pulse output "H" width} = (1/f_i) \times (m+1) \times n \text{ [s]}$$

f_i : Frequency of selected count source [Hz]

n : Value in the high-order 8 bits of the counter
(Value between 00₁₆ and FE₁₆)

m : Value in the low-order 8 bits of the counter
(Value between 00₁₆ and FF₁₆)

FUNCTIONAL DESCRIPTION

2.7 Timer A

Figure 2.7.30 shows an output waveform example in 8-bit PWM mode. In 8-bit PWM mode, when a trigger occurs, pulse output starts at the set pulse period after “L” level with the width equal to the “H” level of the set pulse is output.

An interrupt request occurs and the interrupt request bit is set to “1” each time the level of the pulse output changes from “H” to “L”.

The “H” level width of the pulse output can be changed while the pulse width modulator is operating (outputting the pulse signal) by writing a value in the timer Ai register. This value is written in the reload register and reloaded into the counter next time the level of the pulse output changes from “L” to “H”. Therefore, the pulse width changes from the pulse period following the pulse period being output when the value was written.

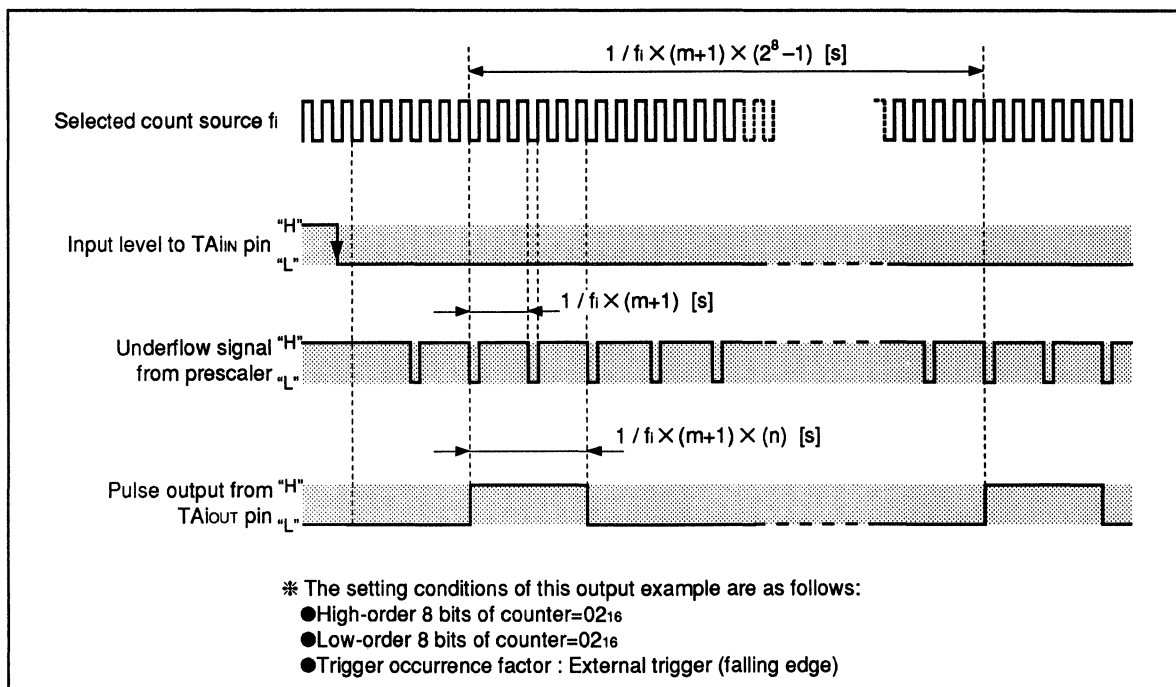


Fig. 2.7.30 8-bit PWM mode output waveform example

[Precautions when using PWM mode]

1. If the count start flag is set to “0” while counting (pulse width modulator is operating), counting stops and the output level of the TA_{out} pin becomes “L”. At the same time an interrupt request occurs and the interrupt request bit is set to “1”.

2. When the operating mode is switched described below, the interrupt request bit in timer Ai interrupt control register is set to “1”.

- When PWM mode is selected after removing reset.
- When the operating mode is switched from timer mode to PWM mode.
- When the operating mode is switched from event counter mode to PWM mode.

Therefore, the interrupt request bit must be cleared to “0” after above switching modes when using timer Ai interrupt or the interrupt request bit.

FUNCTIONAL DESCRIPTION

2.8 Timer B

2.8 Timer B

Timer B consists of three 16-bit timers (timers B0 to B2). Timers B0 to B2 operate independently and each can operate in one of three different modes.

2.8.1 Timer B description

Timer Bi ($i=0$ to 2) has three operating modes as described below. These timers have identical functions. The input pins (TBi_{IN}) of these timers are shared with ports P6₅–P6₇.

●Timer mode

This mode counts the selected internal clock. The counter is decremented by 1 each time a count source (selected clock) is input and a timer Bi interrupt request occurs when the contents of the counter reach $0000_{16} \rightarrow$ reloaded value "n" (hereafter referred to as underflow).

●Event counter mode

This mode counts the external clock input to the TBi_{IN} pin. The counter is decremented by 1 each time a clock is input and a timer Bi interrupt request occurs when the counter underflows.

●Pulse period/pulse width measurement mode

In this mode, the period or the pulse width of the input signal to the TBi_{IN} pin is measured.

2.8.2 Block description

Figure 2.8.1 shows the block diagram of timer Bi. It is followed by the description of the timer Bi related registers.

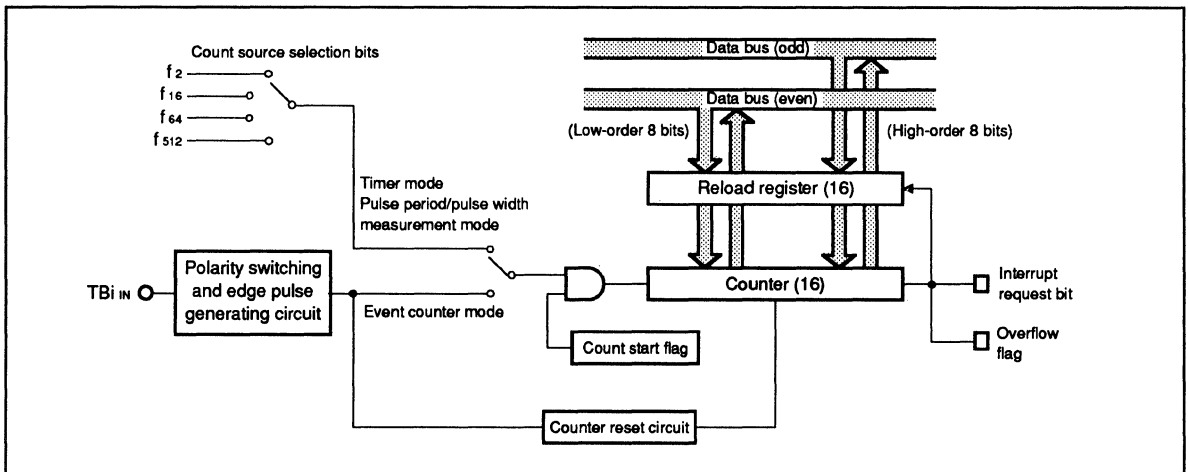


Fig. 2.8.1 Timer Bi block diagram

FUNCTIONAL DESCRIPTION

2.8 Timer B

(1) Counter and reload register (timer Bi register)

Timer Bi counter and reload register consist of 16 bits. The counter counts the selected count source. In timer mode and event counter mode, the contents of the counter are decremented by 1 each time a count source is input. The reload register is used to store the initial value of the counter. The contents of the counter change each time a count source is input, but the contents of the reload register remain unchanged. The contents of the reload register are reloaded into the counter when the counter underflows. In pulse period/pulse width measurement mode, the contents of the counter are incremented by 1 each time a count source is input to the counter. The result of measuring the pulse signal is transferred to the reload register. Values are stored in the counter and the reload register by writing to the timer Bi register.

Table 2.8.1 shows the memory allocation of the timer Bi register.

Table 2.8.1 Timer Bi register memory allocation

Timer Bi register	High-order byte	Low-order byte
Timer B0 register	Address 51 ₁₆	Address 50 ₁₆
Timer B1 register	Address 53 ₁₆	Address 52 ₁₆
Timer B2 register	Address 55 ₁₆	Address 54 ₁₆

In timer mode and event counter mode, the value written in the timer Bi register when the timer is not operating is stored in the counter and reload register. The value written in the timer Bi register when the timer is operating is stored only in the reload register. In this case, the updated value is transferred to the counter during next reload. When the timer Bi register is read, the counting value (value in the counter) is read. When the timer Bi register is read in pulse period/pulse width measurement mode, the pulse width or the pulse period measurement result is read.

The value of the timer Bi register is undefined at reset. Therefore, at first the counter and the reload register must be initialized when using timer Bi in timer mode or event counter mode.

(2) Count start flag

The count start flag (address 40₁₆) separately controls starting/stopping of each timer. Each bit corresponds to one of the timers.

When this flag is set to "1", a count source is input to the counter. When this flag is set to "0", a count source input to the counter is disabled.

Figure 2.8.2 shows the structure of the count start flag.

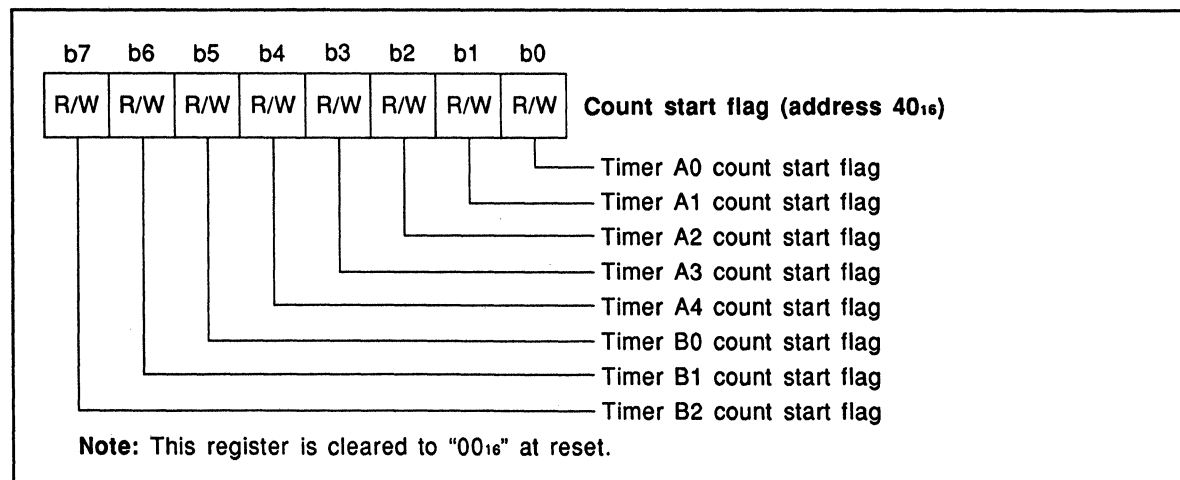


Fig. 2.8.2 Count start flag register structure

FUNCTIONAL DESCRIPTION

2.8 Timer B

(3) Timer BI mode register

The timer Bi mode register (addresses 5B₁₆ to 5D₁₆) consists of operating mode selection bits, count source selection bits and bits that have different function by operating modes.

Figure 2.8.3 shows the structure of the timer Bi mode register. The operating mode selection bits and count source selection bits are described below. The function of bits 2, 3 and 5 depend on the operating mode and are described under the description of each operating mode.

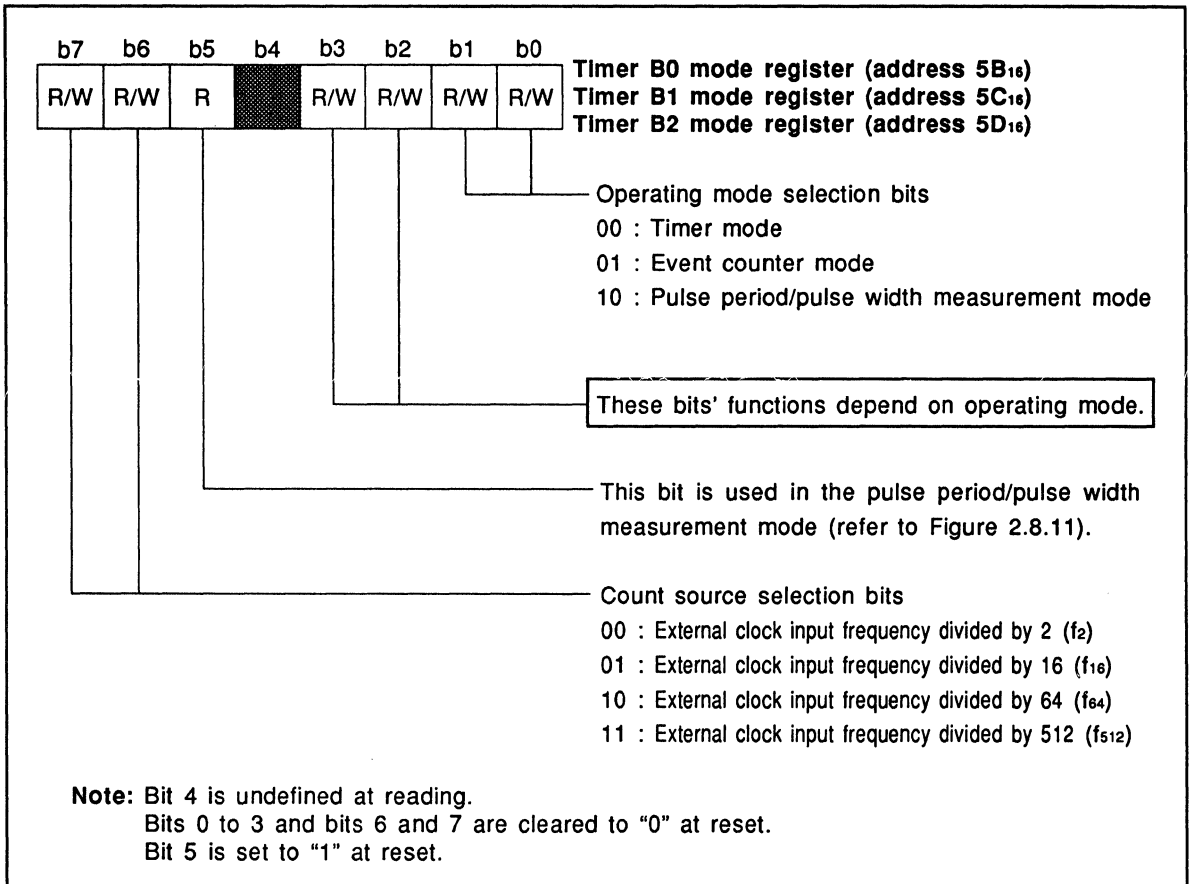


Fig. 2.8.3 Timer BI mode register structure

FUNCTIONAL DESCRIPTION

2.8 Timer B

●Operating mode selection bits (bits 0 and 1)

The operating mode selection bits are used to select the timer operating mode. Table 2.8.2 shows the relationship between the operating mode selection bits and the timer operating modes.

Table 2.8.2 Relationship between operating mode selection bits and operating modes

b1	b0	Operating mode
0	0	Timer mode
0	1	Event counter mode
1	0	Pulse period/pulse width measurement mode

●Count source selection bits (bits 6 and 7)

The count source selection bits are used to select the count source. Table 2.8.3 shows the relationship between the count source selection bits and the timer count sources.

These bits are ignored in event counter mode.

Table 2.8.3 Relationship between count source selection bits and count sources

b7	b6	Timer count source	Input clock to the counter		
			$f(X_{IN})=8\text{MHz}$	$f(X_{IN})=16\text{MHz}$	$f(X_{IN})=25\text{MHz}$
0	0	External clock input frequency divided by 2 (f_2)	4MHz	8MHz	12.5MHz
0	1	External clock input frequency divided by 16 (f_{16})	500kHz	1MHz	1.5625MHz
1	0	External clock input frequency divided by 64 (f_{64})	125kHz	250kHz	390.625kHz
1	1	External clock input frequency divided by 512 (f_{512})	15625Hz	31250Hz	48.8281kHz

$f(X_{IN})$: External clock input frequency

FUNCTIONAL DESCRIPTION

2.8 Timer B

(4) Timer Bi Interrupt control register

The timer Bi interrupt control register (addresses 7A₁₆ to 7C₁₆) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.8.4 shows the structure of the timer Bi interrupt control register. The function of each bit is described below. Use the **SEB** and **CLB** instructions when setting the timer Bi interrupt control register.

Refer to section "2.6 Interrupts" for more information.

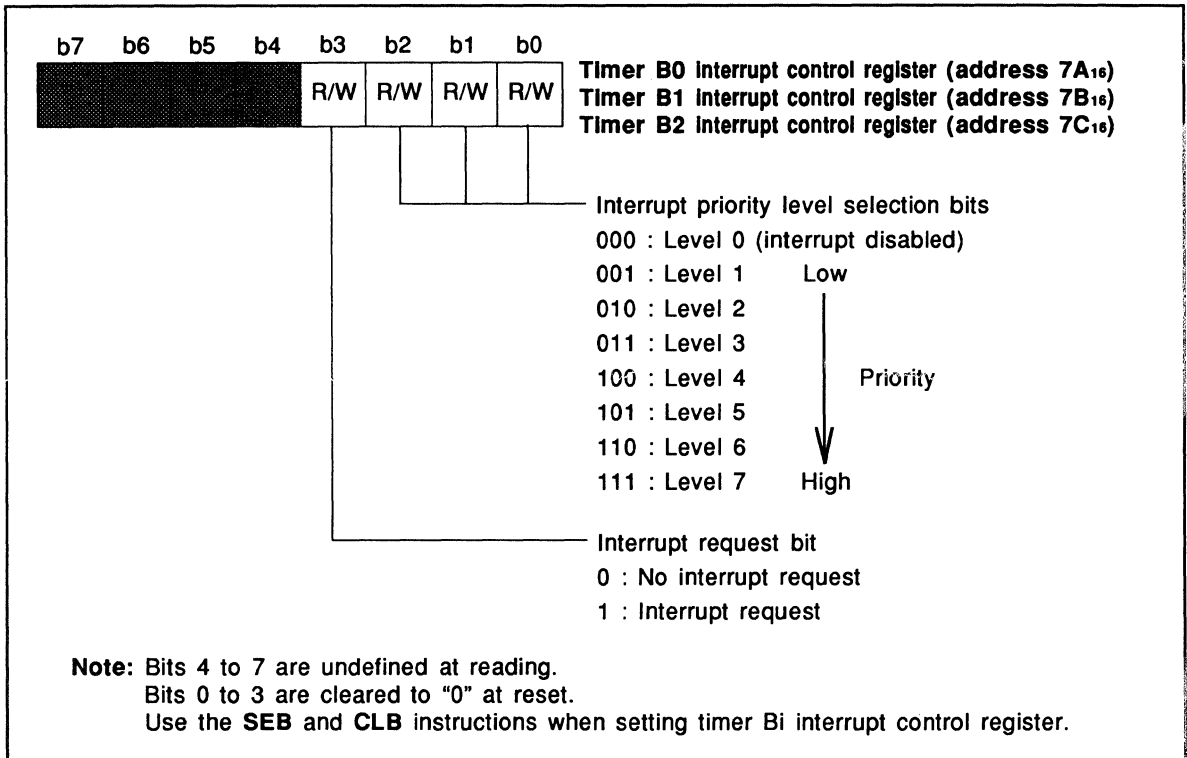


Fig. 2.8.4 Timer Bi Interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using a timer Bi interrupt. When an interrupt request occurs, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is higher than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable only timer Bi interrupt.

●Interrupt request bit (bit 3)

This bit is set to "1" when a timer Bi interrupt request occurs. The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted. This bit can be set or cleared by program.

FUNCTIONAL DESCRIPTION

2.8 Timer B

(5) Port P6 direction register

Timers B0 to B2 input pins are shared with port P6. When using these ports as timer input pins, the corresponding bit in the direction register must be set to "0" (input mode).

Figure 2.8.5 shows the relationship between the port P6 direction register (address 10₁₆) and timer pins.

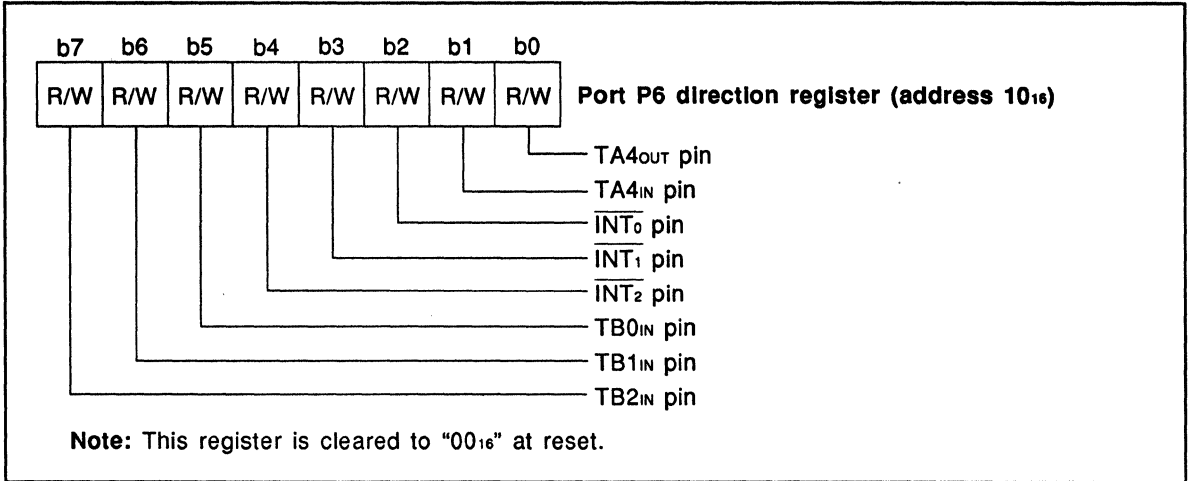


Fig. 2.8.5 Relationship between port P6 direction register and timer pins

FUNCTIONAL DESCRIPTION

2.8 Timer B

2.8.3 Timer mode [timer BI mode register bits 1, 0="00"]

Timer mode is selected by setting the timer Bi mode register bits 1 and 0 to "00". When this mode is selected, bits 2 and 3 may be "0" or "1" because they are ignored. Bit 5 is also ignored. It cannot be written and is undefined at reading. Figure 2.8.6 shows the structure of the timer Bi mode register in timer mode. In timer mode, the selected internal clock is decremented and an interrupt request occurs each time the counter underflows (the contents of the counter reach $0000_{16} \rightarrow$ reloaded value "n"). The timer dividing ratio is expressed as follows:

$$\text{Timer dividing ratio} = 1/(n+1)$$

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

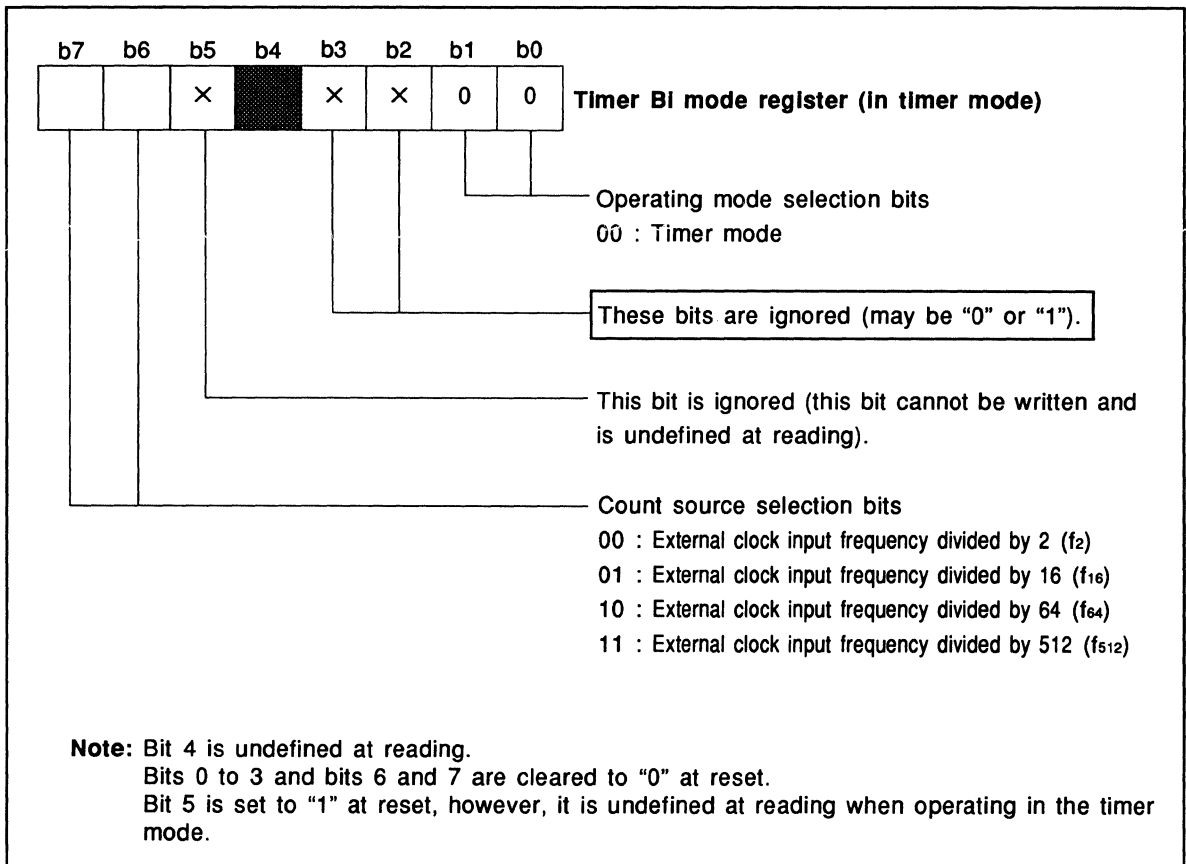


Fig. 2.8.6 Timer BI mode register structure in timer mode

(1) Timer mode operation

First, select the operating mode and count source with the timer Bi mode register. Next, write an arbitrary value "n" ("n"=0000₁₆ to FFFF₁₆) in the timer Bi register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register.

When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and starts the count operation.

Figure 2.8.7 shows the setting example of the timer mode related registers.

The contents of the counter are decremented by 1 each time the count source is input. When the counter underflows, the contents of the reload register are reloaded into the counter and the interrupt request bit is set to "1".

Count operation continues in this manner. The interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, an interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program.

The contents of the counter can be read at any timing by reading the contents of the timer Bi register, but the contents of the reload register can not be read. In order to change the timer dividing ratio when the timer is operating,

a 16-bit update value must be written simultaneously in the timer Bi register. This value is stored in the reload register, and reloaded into the counter at the next reload timing after writing to timer Bi register.

Figure 2.8.8 shows the timer mode operation diagram.

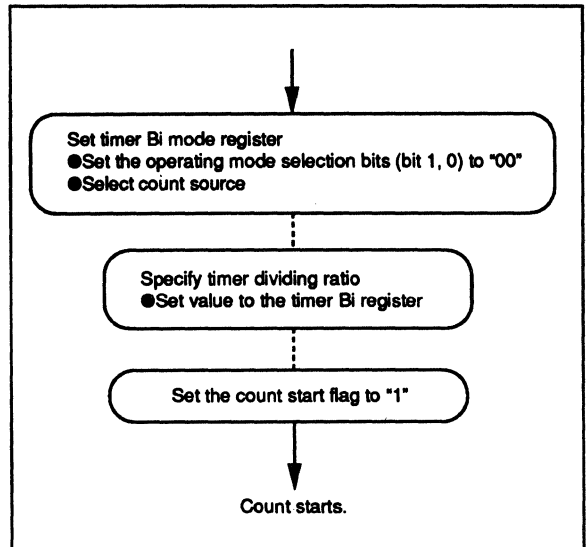


Fig. 2.8.7 Setting example of the timer mode related registers

This value is stored in the reload register, and reloaded into the counter at the next reload timing after writing to timer Bi register.

FUNCTIONAL DESCRIPTION

2.8 Timer B

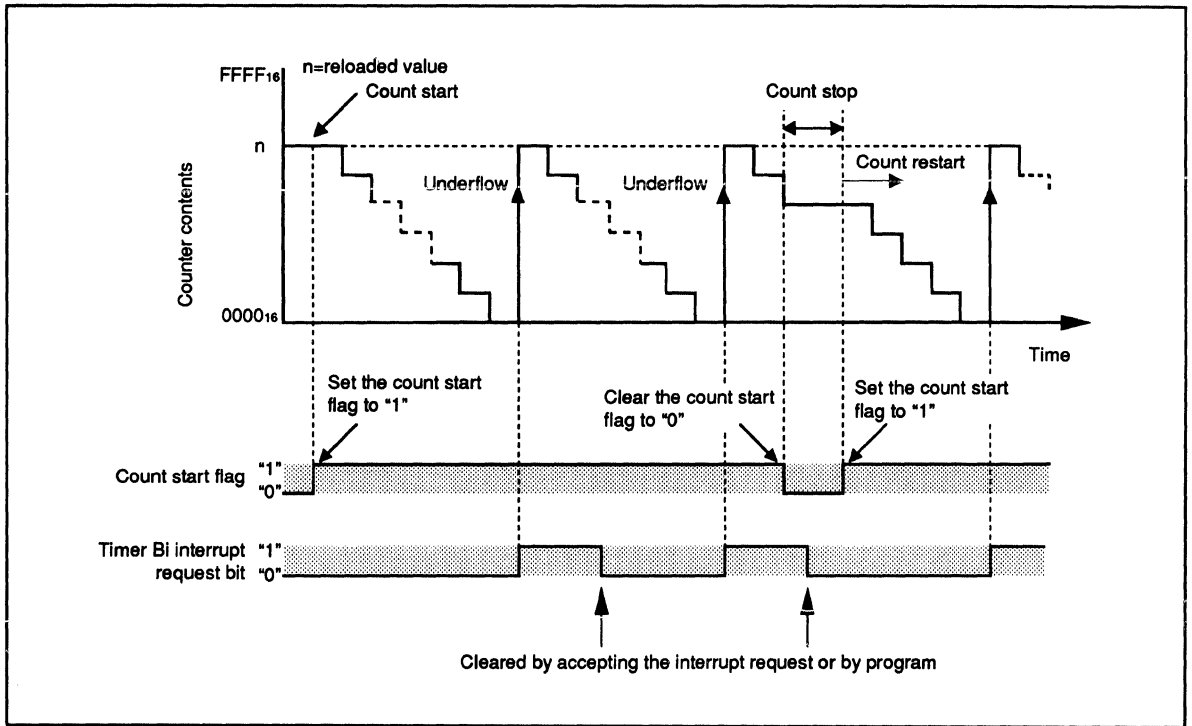


Fig. 2.8.8 Timer mode operation diagram

FUNCTIONAL DESCRIPTION

2.8 Timer B

2.8.4 Event counter mode [timer Bi mode register bits 1, 0="01"]

The event counter mode is selected by setting the timer Bi mode register bit 1 to "0" and bit 0 to "1". When this mode is selected, bits 6 and 7 may be "0" or "1" because they are ignored. Bit 5 is also ignored. It cannot be written and is undefined at reading. Figure 2.8.9 shows the structure of the timer Bi mode register in event counter mode.

In event counter mode, the external clock input to the TB_{IN} pin is counted. The counter is decremented each time an effective edge is input and an interrupt request occurs when the counter underflows (the contents of the counter reach 0000₁₆ → reloaded value "n").

The timer dividing ratio is expressed as follows:

$$\text{Timer dividing ratio} = 1/(n+1)$$

n: Value set in counter
(value between 0000₁₆ and FFFF₁₆)

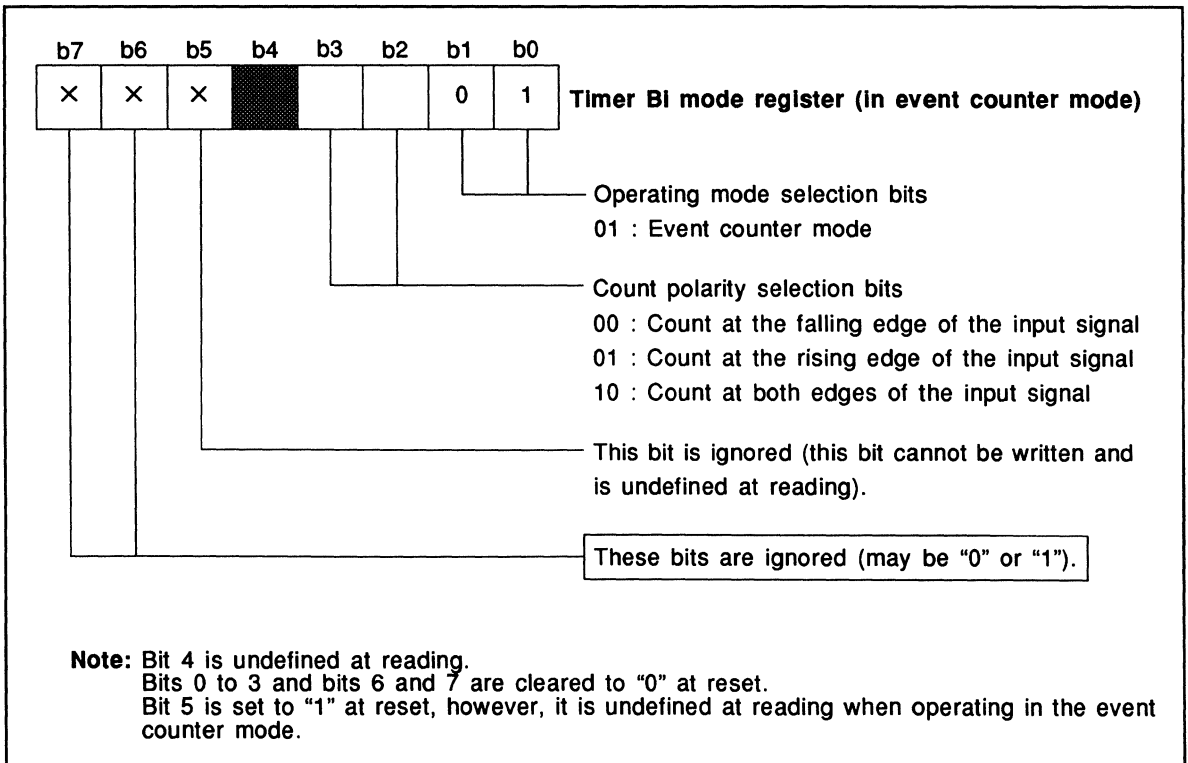


Fig. 2.8.9 Timer Bi mode register structure in event counter mode

FUNCTIONAL DESCRIPTION

2.8 Timer B

(1) Event counter mode operation

First, select the operating mode and the effective edge of the count source with the timer Bi mode register. Next, write an arbitrary value "n" ("n"=0000₁₆ to FFFF₁₆) in the timer Bi register to specify the timer dividing ratio. At the same time, the value "n" is stored in the counter and the reload register. Also, set the corresponding port P6 direction register bit to TBi_{IN} pin to "0" (input mode). When the count start flag is set to "1" (count enabled), the effective edge of the input signal to TBi_{IN} pin is detected and count operation starts.

Table 2.8.4 shows the relationship between the count polarity selection bits and the effective edge of the count. Figure 2.8.10 shows the setting example of the event counter mode related registers.

Table 2.8.4 Relationship between count polarity selection bits and effective edge

b3	b2	Count effective edge
0	0	Falling edge of input signal
0	1	Rising edge of input signal
1	0	Both edges of input signal

The contents of the counter are decremented by 1 each time an effective edge is input. When the counter underflows, the contents of the reload register are reloaded into the counter and the interrupt request bit is set to "1". The count operation continues in this manner and an interrupt request occurs and the interrupt request bit is set to "1" each time the counter underflows. Therefore, a timer Bi interrupt request occurs at every "n+1" count of the count source. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program.

The contents of the counter can be read at any timing by reading the contents of timer Bi register, but the contents of the reload register can not be read.

In order to change the timer dividing ratio when the timer is operating, a 16-bit update value must be written simultaneously in the timer Bi register. This value is stored in the reload register, and reloaded into the counter at the next reload timing after writing to timer Bi register. The operation in event counter mode is identical to that of the timer mode except that an external clock input to the TBi_{IN} pin is counted. Refer to "Figure 2.8.8 Timer mode operation diagram" about an operation diagram.

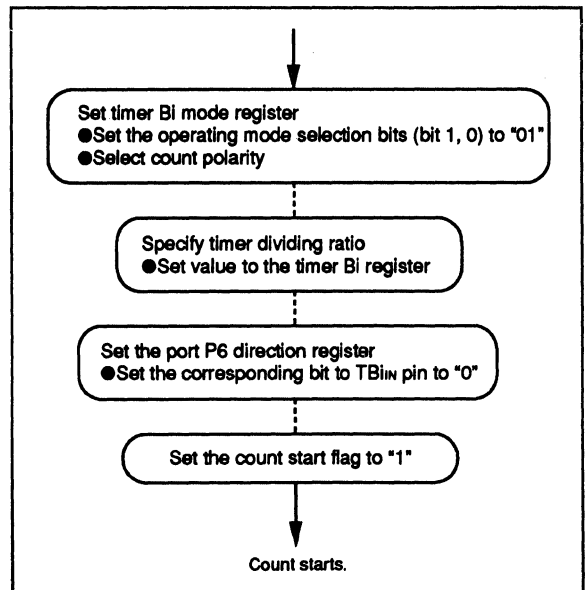


Fig. 2.8.10 Setting example of the event counter mode related registers

2.8.5 Pulse period/pulse width measurement mode [timer Bi mode register bits 1, 0="10"]

The pulse period/pulse width measurement mode is selected by setting the timer Bi mode register bit 1 to "1" and bit 0 to "0". The difference between pulse period measurement and pulse width measurement is the effective edge of the input signal determining the count term.

●Pulse period measurement

Pulse period is measured by counting during the period between the falling edge and the next falling edge or between the rising edge and the next rising edge of the input signal to the TB_{IN} pin.

●Pulse width measurement

Pulse width is measured by counting during the period between the falling edge and the next rising edge or between the rising edge and the next falling edge of the input signal to the TB_{IN} pin.

Figure 2.8.11 shows the structure of the timer Bi mode register in pulse period/pulse width measurement mode.

(1) Pulse period /pulse width measurement mode description

First, select the operating mode, whether to measure the pulse period or width, and count source with the timer Bi mode register. Set the corresponding port P6 direction register bit to TB_{IN} pin to "0" (input mode). When the count start flag is set to "1" (count enabled), the selected count source is input to the counter and count operation starts. The counter is incremented by 1 each time a count source is input.

The contents of the counter are transferred to the reload register when an effective edge is input to the TB_{IN} pin. Then the counter is cleared to 0000₁₆. The count operation continues and the contents of the counter are transferred to the reload register again when the next effective edge is input to the TB_{IN} pin. Then the counter is cleared to 0000₁₆. At the same time, an interrupt request occurs and the interrupt request bit is set to "1". The measured result can be obtained by reading the timer Bi register because the contents of the reload register are read. An interrupt request does not occur and an undefined value is transferred to the reload register at the first effective edge after setting the count start flag to "1".

The interval between the falling edge and the rising edge or between the rising edge and the falling edge of the input signal to the TB_{IN} pin must be at least 2 cycles of the selected count source. When measuring the pulse width of a signal other than 50% duty, whether the result (contents of the reload register) is "H" level width or "L" level width must be determined with program.

Table 2.8.5 shows the relationship between measurement mode selection bits and measured intervals.

Table 2.8.5 Relationship between measurement mode selection bits and measured intervals

b3	b2	Measurement mode	Measured intervals
0	0	Pulse period	Between the falling edge and the falling edge of the input signal
0	1		Between the rising edge and the rising edge of the input signal
1	0	Pulse width	Between the falling edge and the rising edge, and between the rising edge and the falling edge of the input signal

FUNCTIONAL DESCRIPTION

2.8 Timer B

In this mode, an interrupt request also occurs when the contents of the counter reach $FFFF_{16} \rightarrow 0000_{16}$ (hereafter referred to as overflow) in addition to the effective edge. The timer Bi overflow flag is used to determine the cause of interrupt in pulse period/pulse width measurement mode whether an overflow of the counter or an effective edge. When an overflow occurs, the timer Bi overflow flag is set to "1". Therefore, the cause of interrupt request must be determined in the interrupt service routine by checking the timer Bi overflow flag.

The timer Bi overflow flag set to "1" is cleared to "0" when writing to the timer Bi mode register while corresponding count start flag is "1". This flag is a read-only flag and set to "1" at reset.

The timer Bi overflow flag must be used to determine the cause of interrupt request and must not be used to detect the overflow timing of the counter. Interrupt request bit must be used to detect the overflow timing of the counter.

Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The interrupt request bit remains "1" set until the interrupt request is accepted or it is cleared by program.

Figure 2.8.12 shows the pulse period/pulse width measurement mode operation diagram in pulse period measurement mode. Figure 2.8.13 shows the pulse period/pulse width measurement mode operation diagram in pulse width measurement mode.

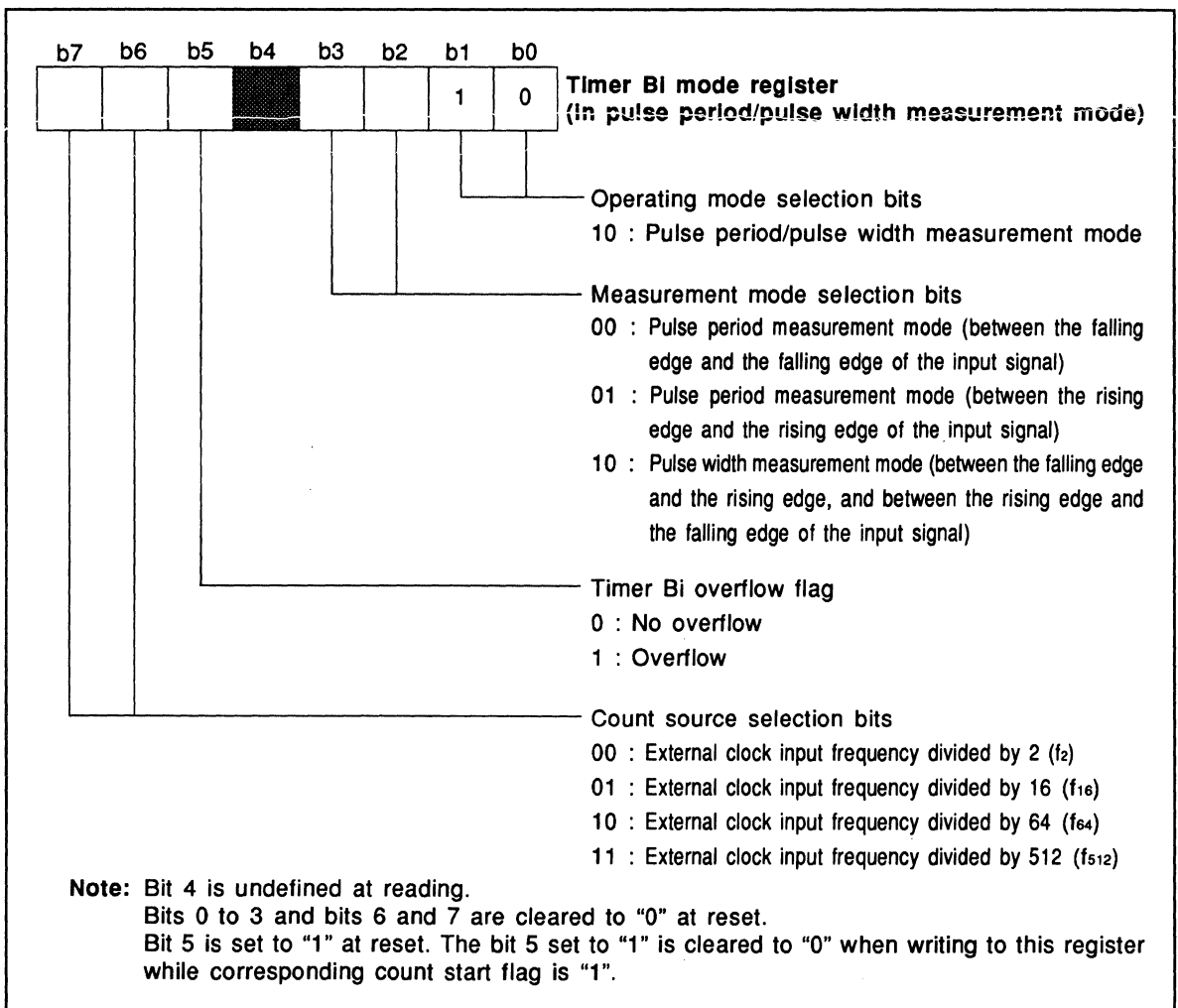


Fig. 2.8.11 Timer Bi mode register structure in pulse period/pulse width measurement mode

FUNCTIONAL DESCRIPTION

2.8 Timer B

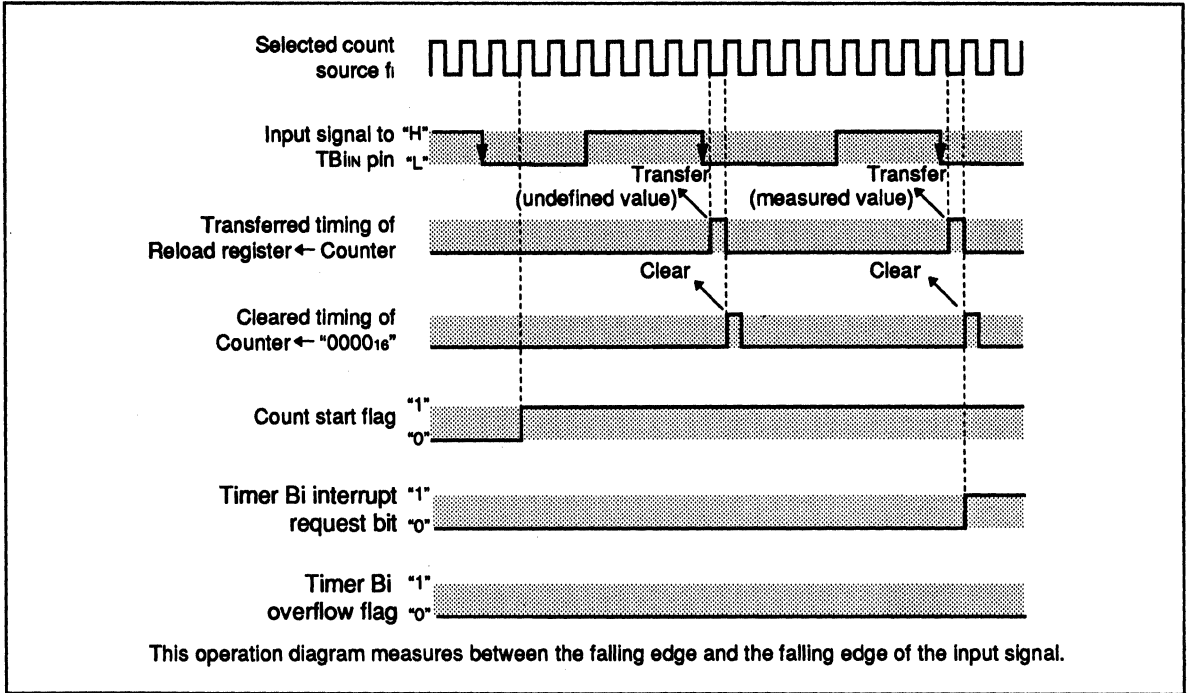


Fig. 2.8.12 Pulse period/pulse width measurement mode operation diagram in pulse period measurement mode

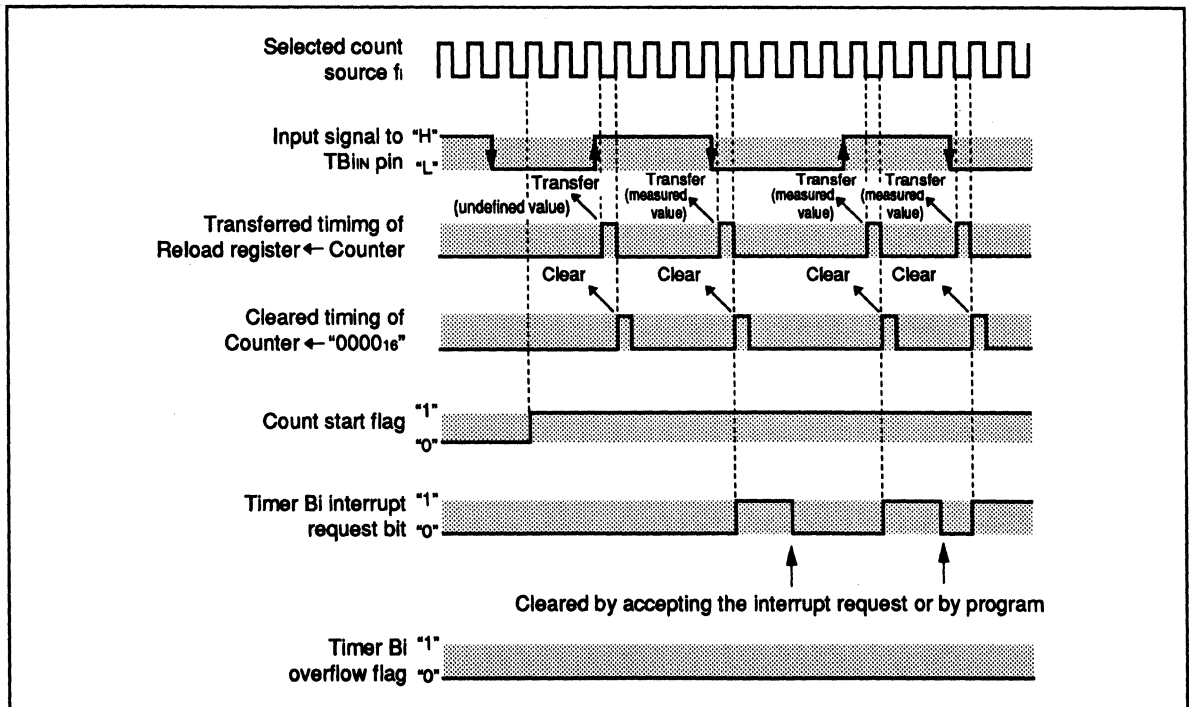


Fig. 2.8.13 Pulse period/pulse width measurement mode operation diagram in pulse width measurement mode

FUNCTIONAL DESCRIPTION

2.8 Timer B

[Precautions when using timer B]

1. The value of the counter while operating in timer mode or event counter mode can be read at any timing by reading the timer Bi register. However, if it is read at the reload timing shown in Figure 2.8.14, "FFFF₁₆" is read instead of the reloaded value. The reloaded value is read if it is read after the timer Bi register is set during timer halted and before the count source is input and count started.

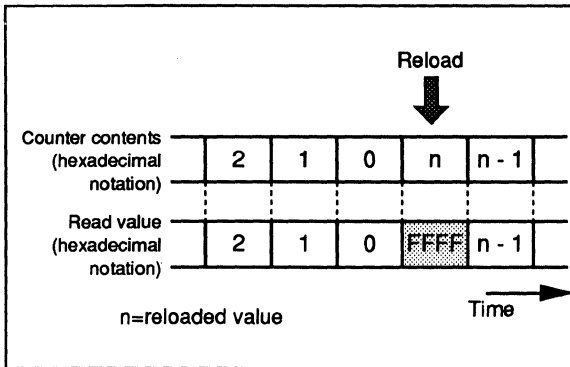


Fig. 2.8.14 Reading the timer Bi register

2. Interrupt request occurs by two causes in pulse period/pulse width measurement mode. These interrupts use the same interrupt vector address of timer Bi. Two factors are follows.
 - Completing the measurement of the pulse period/pulse width
 - Overflow of the counter
3. The timer Bi overflow flag of the timer Bi mode register in pulse period/pulse width measurement mode must be used to determine the cause of interrupt. Interrupt request bit of the timer Bi interrupt control register must be used to detect the overflow timing of the counter. The timer Bi overflow flag set to "1" is cleared to "0" when writing to the timer Bi mode register while corresponding count start flag is "1". This flag is set to "1" at reset.
4. In pulse period/pulse width measurement mode, an undefined value is transferred to the reload register at the first effective edge after setting the count start flag to "1" and count starts. An interrupt request does not occur at the first effective edge.

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

2.9 Pulse output function

The M37732 group is equipped with a pulse output function using timer A. The mode using the pulse output function is called the pulse output port mode. Pulse output port mode controls two 4-bit pulse output ports using timers A0 and A2.

The pulse output function of M37730 group is different from it of M37732 group described this section in points of the pulse output pins structure and pulse output control. Refer to "CHAPTER 8. M37730 GROUP" for more information.

2.9.1 Pulse output function overview

Pulse output port mode is divided into two waveform output modes: RTP0 which controls a 4-bit pulse output port with timer A0 and RTP1 which controls it with timer A2.

When using a pulse output port mode, timers A0 and A2 must be operated in timer mode. The pulse output pins (RTP0₀–RTP0₃) for RTP0 are shared with ports P5₀–P5₃ and the pulse output pins (RTP1₀–RTP1₃) for RTP1 are shared with ports P5₄–P5₇. Set the output data in pulse output data registers 0 and 1. When the control timers A0 and A2 underflow (contents of counter change from 0000₁₆ → reloaded value n), the contents of pulse output data registers 0 and 1 at that point are output from pulse output pins.

Pulse width modulation can be performed for RTP0 using timer A1 and for RTP1 using timer A3. In addition, for RTP0, the contents of pulse output data register 0 can be selected to output inverted (negative polarity pulse) by program.

Table 2.9.1 shows the overview of pulse output function.

Table 2.9.1 Pulse output function overview

	Pulse output port mode	
	RTP0	RTP1
Waveform output mode	RTP0	RTP1
Control timer	Timer A0	Timer A2
Pulse output pin	RTP0 ₀ –RTP0 ₃ (ports P5 ₀ –P5 ₃)	RTP1 ₀ –RTP1 ₃ (ports P5 ₄ –P5 ₇)
Register for setting pulse data	Pulse output data register 0	Pulse output data register 1
Pulse width modulation	Possible (timer A1 is used)	Possible (timer A3 is used)
Negative polarity pulse output function	Possible	Impossible

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

2.9.2 Block description

Figure 2.9.1 shows the block diagram for the pulse output port mode. The pulse output port mode related registers are described below.

Figure 2.9.2 shows the port P5 output control circuit.

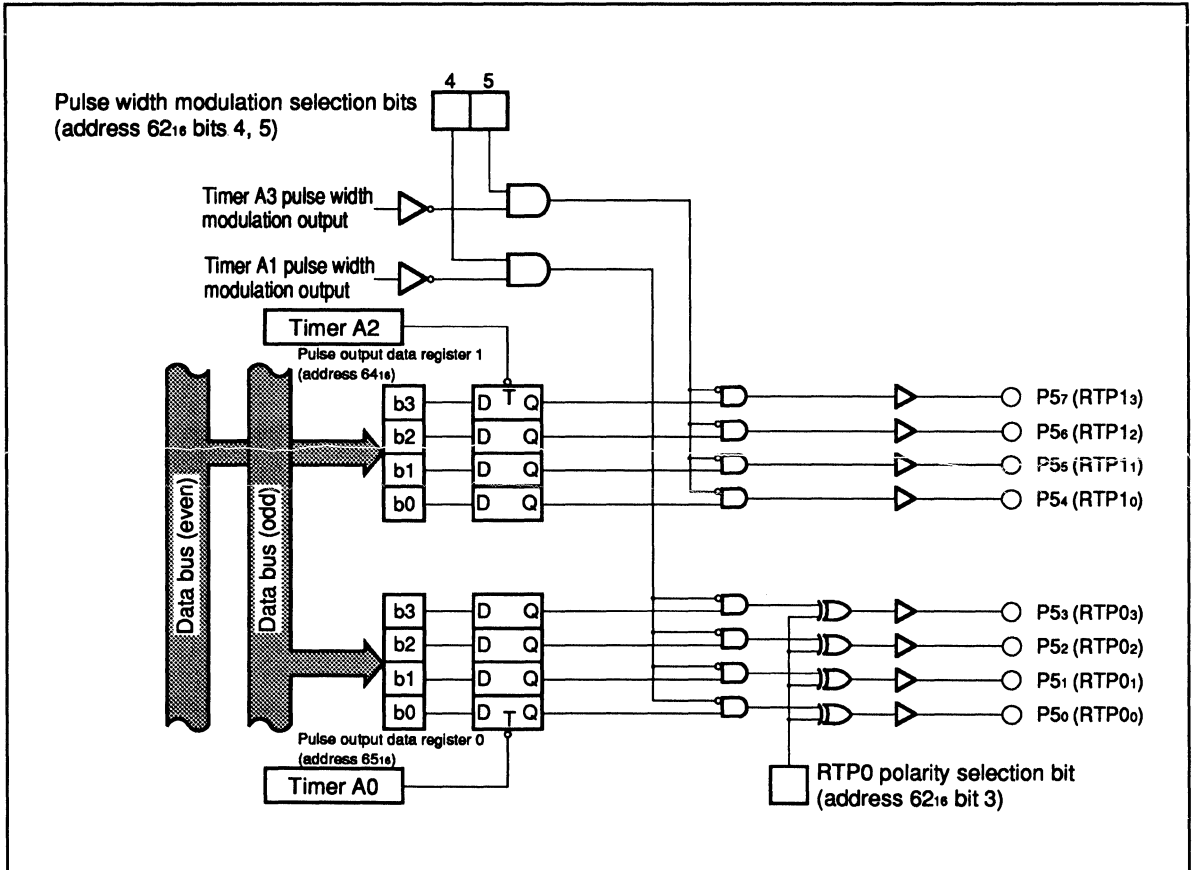
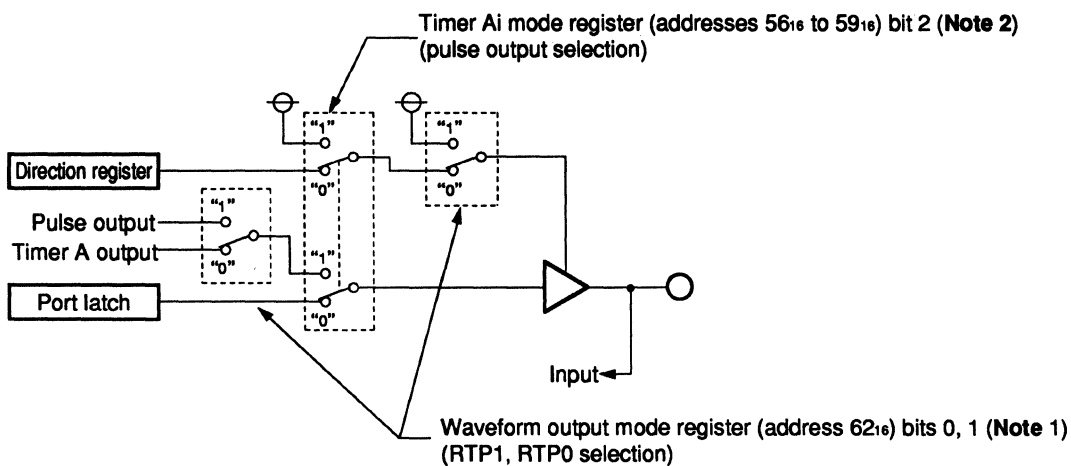


Fig. 2.9.1 Pulse output port mode block diagram

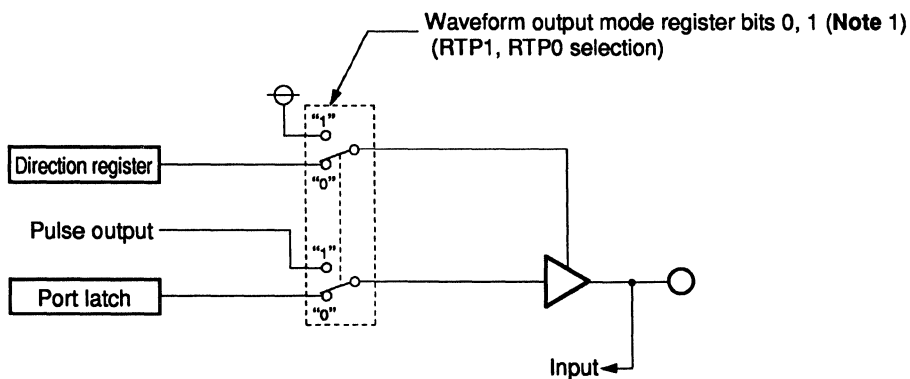
FUNCTIONAL DESCRIPTION

2.9 Pulse output function

- P5₀/TA0_{OUT}/RTP0₀, P5₂/TA1_{OUT}/RTP0₂, P5₄/TA2_{OUT}/RTP1₀, P5₆/TA3_{OUT}/RTP1₂



- P5₁/TA0_{IN}/RTP0₁, P5₃/TA1_{IN}/RTP0₃, P5₅/TA2_{IN}/RTP1₁, P5₇/TA3_{IN}/RTP1₃



Note 1: P5₀–P5₃ are bit 1, P5₄–P5₇ are bit 0.

Note 2: Timer Ai mode register bit 2 corresponding to each port.

Fig. 2.9.2 Port P5 output control circuit

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(1) Counter and reload register (timer A0 to A3 registers)

In pulse output port mode, timers A0 and A2 must be used in timer mode to control the pulse output. When modulating these pulse outputs, timers A1 and A3 must be used in pulse width modulation mode.

Timer Ai (i=0 to 3) counter and reload register consist of 16 bits. The counter counts the selected count source and its contents are decremented by 1 each time a count source is input. The reload register is used to store the initial value of the counter. The contents of the reload register are reloaded into the counter when the counter underflows.

The contents of the counter change each time a count source is input, but the contents of the reload register remain unchanged.

Values are stored in the counter and reload register by writing to the timer Ai register. Table 2.9.2 shows the memory allocation of timer Ai register. The value written in timer Ai register when the timer is not operating is stored in the counter and reload register. The value written in timer Ai register when the timer is operating is stored only in the reload register. In this case, the updated value is transferred to the counter during next reload. If the timer Ai register is read, the result depends on the operating mode. Table 2.9.3 shows results of timer Ai register read and write.

The value of the timer Ai register is undefined at reset. Therefore, at first the counter and the reload register must be initialized when using timer Ai.

Table 2.9.2 Timer Ai register memory allocation

Timer Ai register	High-order byte	Low-order byte
Timer A0 register	Address 47 ₁₆	Address 46 ₁₆
Timer A1 register	Address 49 ₁₆	Address 48 ₁₆
Timer A2 register	Address 4B ₁₆	Address 4A ₁₆
Timer A3 register	Address 4D ₁₆	Address 4C ₁₆

Table 2.9.3 Timer Ai register read and write

Mode	Read	Write
Timer mode (timers A0 and A2)	Read timer counting value	<Timer operating> Write to reload register
Pulse width modulation mode (timers A1 and A3)	Read undefined value	<Timer halted> Write to reload register and counter

(2) Count start flag

The count start flag (address 40₁₆) separately controls the starting/stopping of each timer. Each bit corresponds to one of the timers.

When this flag is set to "1", a count source is input to the counter. When this flag is set to "0", a count source input to the counter is disabled.

Figure 2.9.3 shows the structure of the count start flag register.

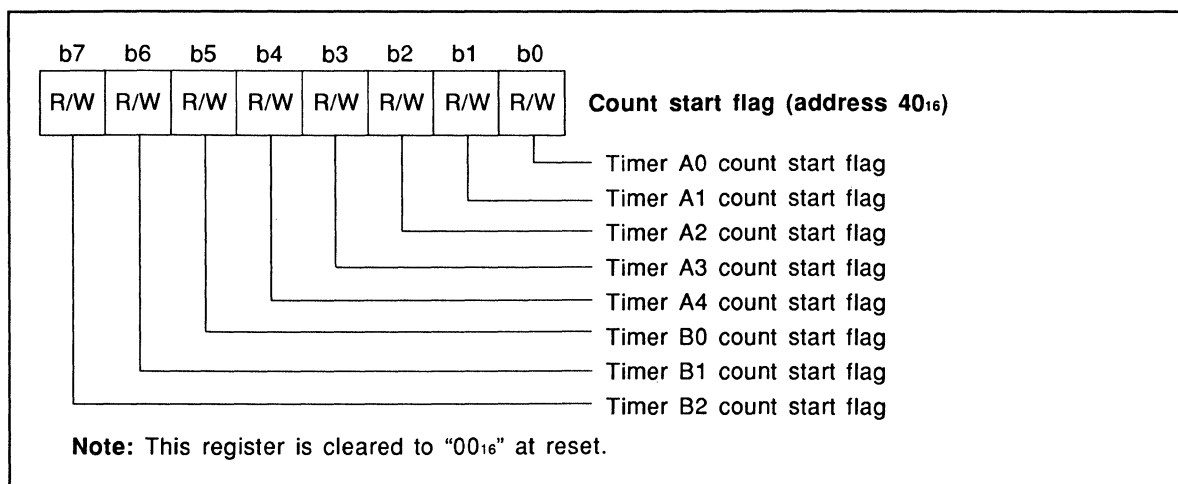


Fig. 2.9.3 Count start flag register structure

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(3) Timer Ai mode register

Timer Ai mode register (addresses 56₁₆ to 59₁₆) consists of operating mode selection bits, count source selection bits, and timer function selection bits. The function of the timer Ai mode register depends on the operating mode of the timer.

In pulse output port mode, timers A0 and A2 must be used in timer mode. Figure 2.9.4 shows the structure of the timer A0 and A2 mode registers (in pulse output port mode).

When modulating the pulse output, timers A1 and A3 must be used in pulse width modulation mode (PWM mode). Figure 2.9.5 shows the structure of the timer A1 and A3 mode registers (when using pulse width modulation in pulse output port mode).

Bit 2 of the timer A1 and A3 mode registers must be fixed to "1" even when the pulse output is not modulated. Figure 2.9.6 shows the structure of the timer A1 and A3 mode registers (when not using pulse width modulation in pulse output port mode). When pulse width modulation is not used in pulse output port mode, timers A1 and A3 can be used in timer mode. However, the gate function cannot be selected because the TA1_{IN} pin and TA3_{IN} pin function as pulse output pins. Therefore, bit 4 must be set to "0".

When using RTP0, bit 2 of the timer A0 and A1 mode registers must be set to "1". When using RTP1, bit 2 of the timer A2 and A3 mode registers must be set to "1".

This is because pins RTP0₀, RTP0₂, RTP1₀, and RTP1₂ are shared with timer A output pins and pulse output is selected by setting timer Ai mode register bit 2 to "1" as shown in Figure 2.9.2.

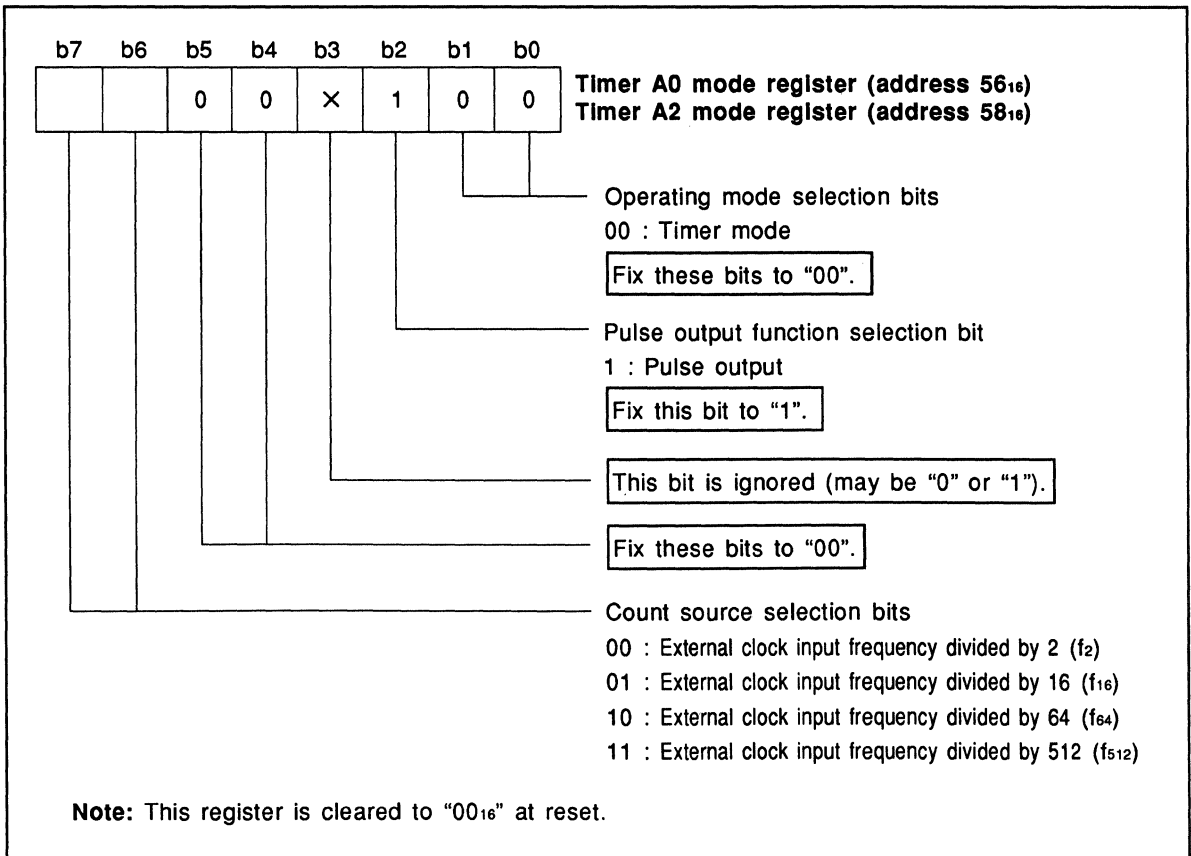


Fig. 2.9.4 Timer A0, A2 mode registers structure (in pulse output port mode)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

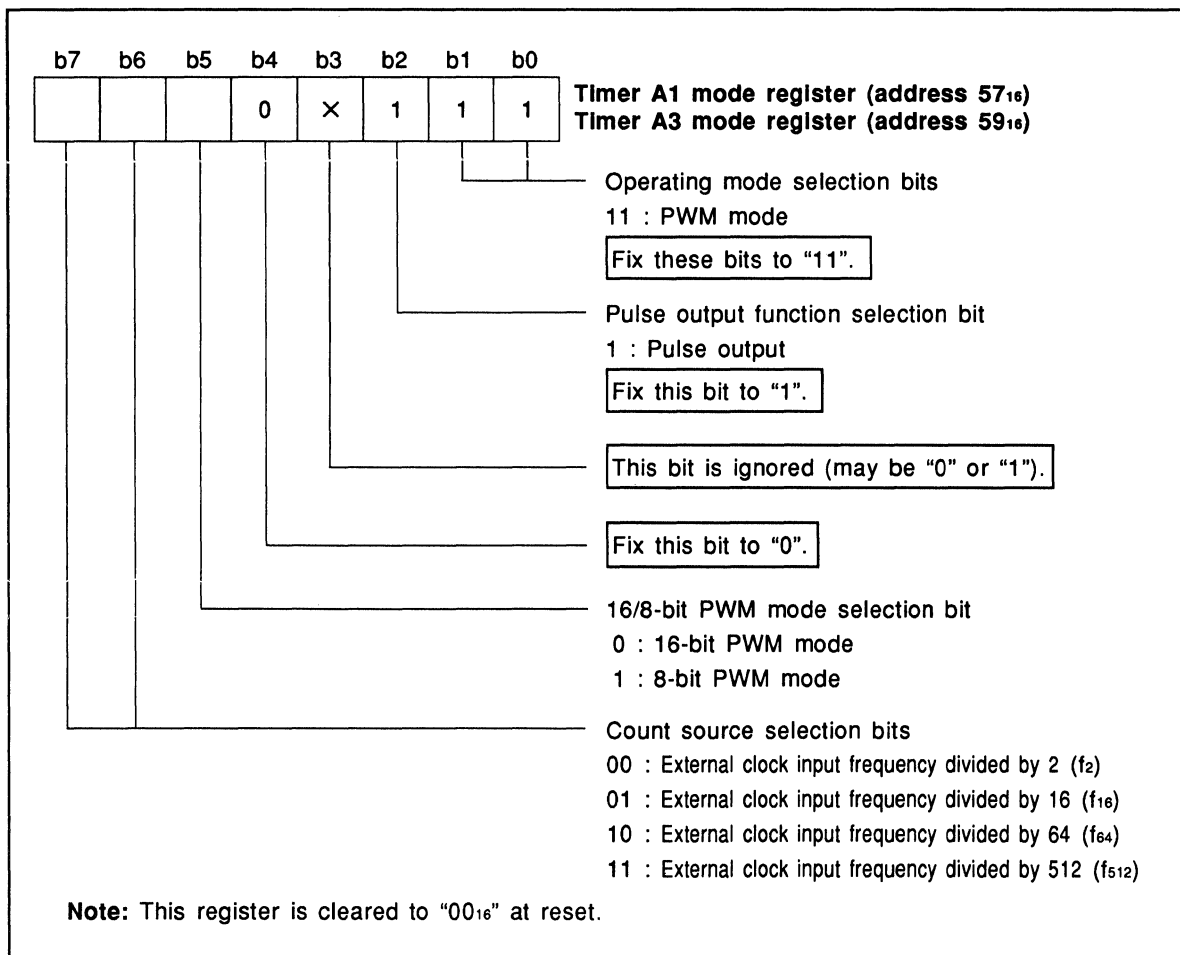


Fig. 2.9.5 Timer A1, A3 mode registers structure (using pulse width modulation in pulse output port mode)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

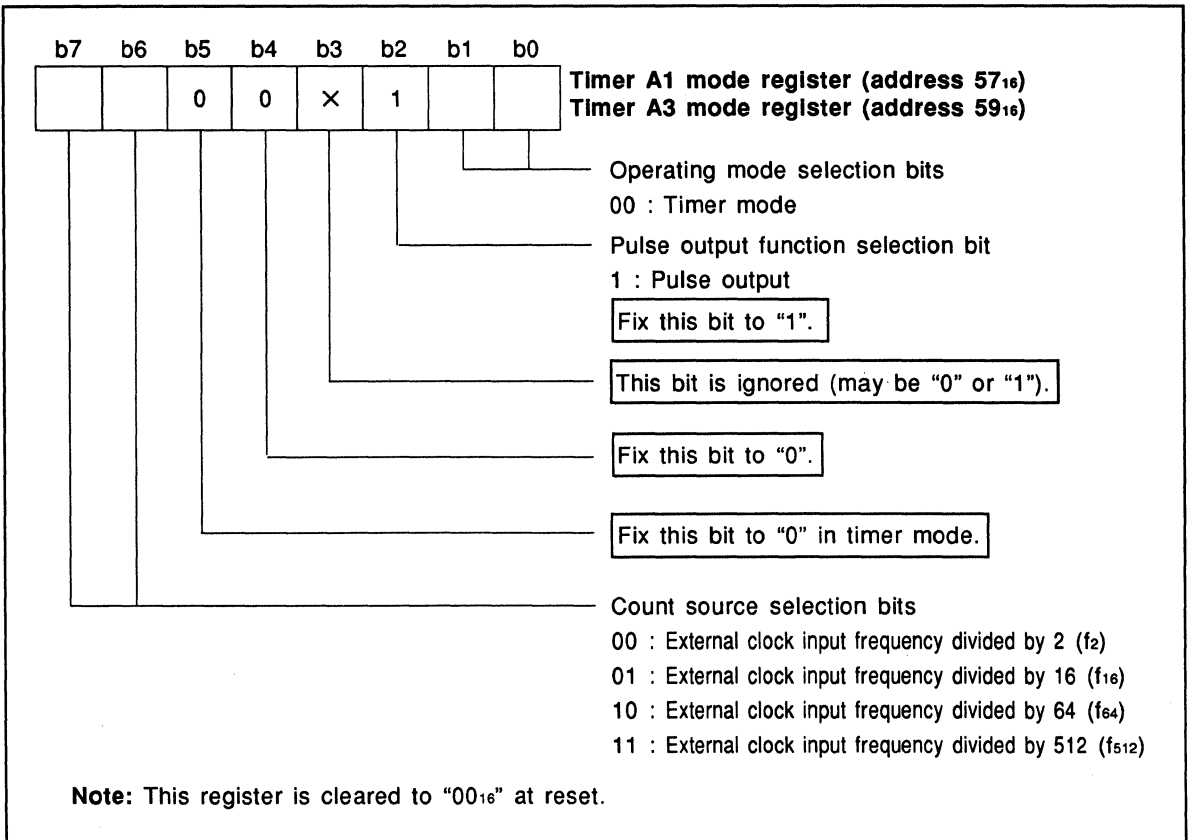


Fig. 2.9.6 Timer A1, A3 mode registers structure (not using pulse width modulation in pulse output port mode)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(4) Waveform output mode register

The waveform output mode register (address 62₁₆) consists of waveform output mode selection bits, RTP0 polarity selection bit, and pulse width modulation selection bit. Figure 2.9.7 shows the structure of the waveform output mode register. Each bit is described below.

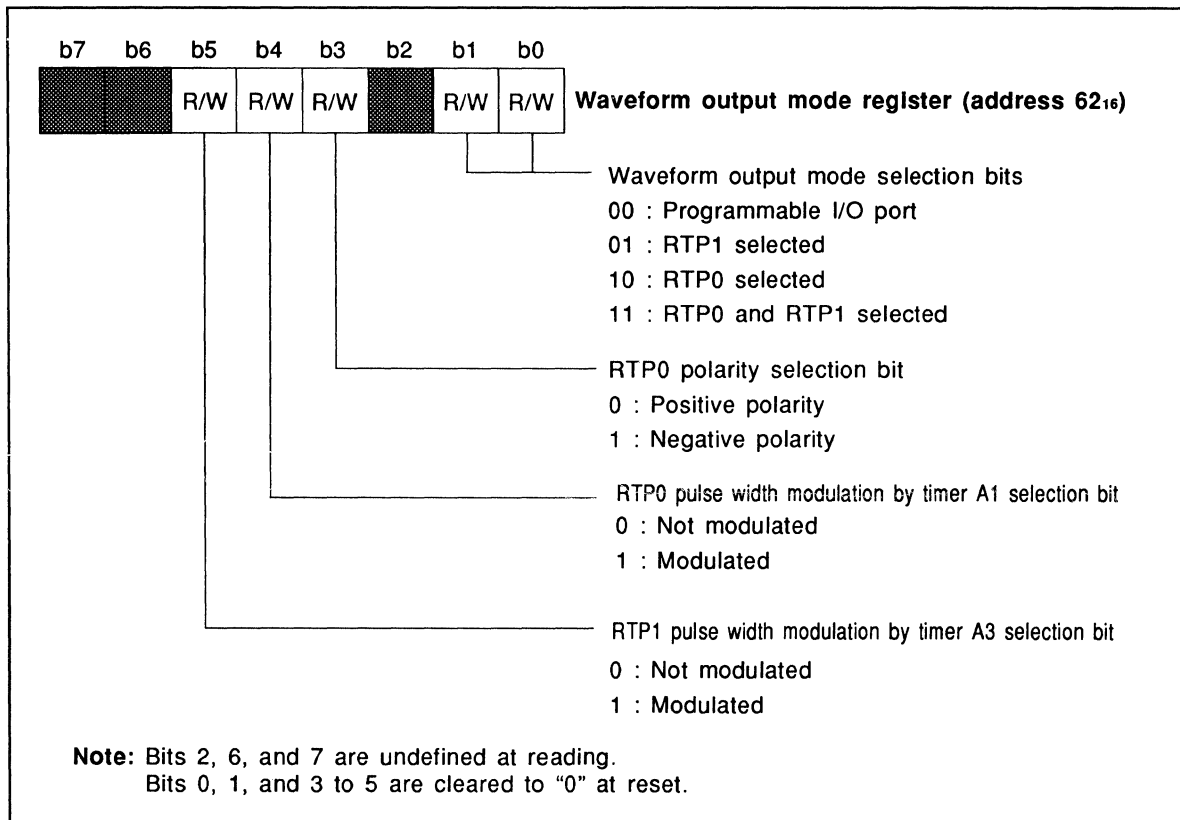


Fig. 2.9.7 Waveform output mode register structure

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

●Waveform output mode selection bits (bits 0, 1)

The waveform output mode selection bits are used to select the waveform output mode in pulse output port mode. Table 2.9.4 shows the relationship between the waveform output mode selection bits and the waveform output mode. When these bits are set to "00", port P5 functions as a programmable I/O port or timer I/O pin. When they are set to "11", RTP0 pulse and RTP1 pulse can be output.

Table 2.9.4 Relationship between waveform output mode selection bits and waveform output mode

b1	b0	Waveform output mode
0	0	Programmable I/O port
0	1	RTP1 selected
1	0	RTP0 selected
1	1	RTP0 and RTP1 selected

●RTP0 polarity selection bit (bit 3)

The RTP0 polarity selection bit is used to select the RTP0 pulse output polarity. When this bit is "0", data is output according to the contents of the pulse output data register 0 (refer to (5) Pulse output data register 1, 0). When this bit is "1", the inverse data (negative polarity pulse) of the contents of the pulse output data register 0 is output.

●RTP0 pulse width modulation by timer A1 selection bit (bit 4)

The RTP0 pulse width modulation by timer A1 selection bit is used to select pulse width modulation for RTP0 pulse output. When this bit is "1", pulse output from RTP0 is modulated. When this bit is "0", pulse output from RTP0 is not modulated.

●RTP1 pulse width modulation by timer A3 selection bit (bit 5)

The RTP1 pulse width modulation by timer A3 selection bit is used to select pulse width modulation for RTP1 pulse output. When this bit is "1", pulse output from RTP1 is modulated. When this bit is "0", pulse output from RTP1 is not modulated.

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(5) Pulse output data registers 1, 0

Pulse output data registers 1 and 0 (addresses 64_{16} , 65_{16}) are used to set output data for RTP1 and RTP0 respectively. The contents of the pulse output data register j ($j=0, 1$) are output from the pulse output pins each time timer A0 and timer A2 underflow respectively.

“L” level is output from the pulse output pin corresponding to the pulse output data bit set to “0” and “H” level is output from the pulse output pin corresponding to the pulse output data bit set to “1”.

However, when the RTP0 polarity selection bit in the waveform output mode register is “1”, the inverse data (negative polarity pulse) of the contents of the pulse output data register 0 is output from pins RTP0₀–RTP0₃. In this case, “H” level is output from the pulse output pin corresponding to the pulse output data bit set to “0” and “L” level is output from the pulse output pin corresponding to the pulse output data bit set to “1”.

Figure 2.9.8 shows the relationship between pulse output data register j and pulse output pins. This is a write-only register. Use the LDM and STA instructions when setting the pulse output data bits. Do not use read-modify-write instruction such as the CLB and SEB instructions.

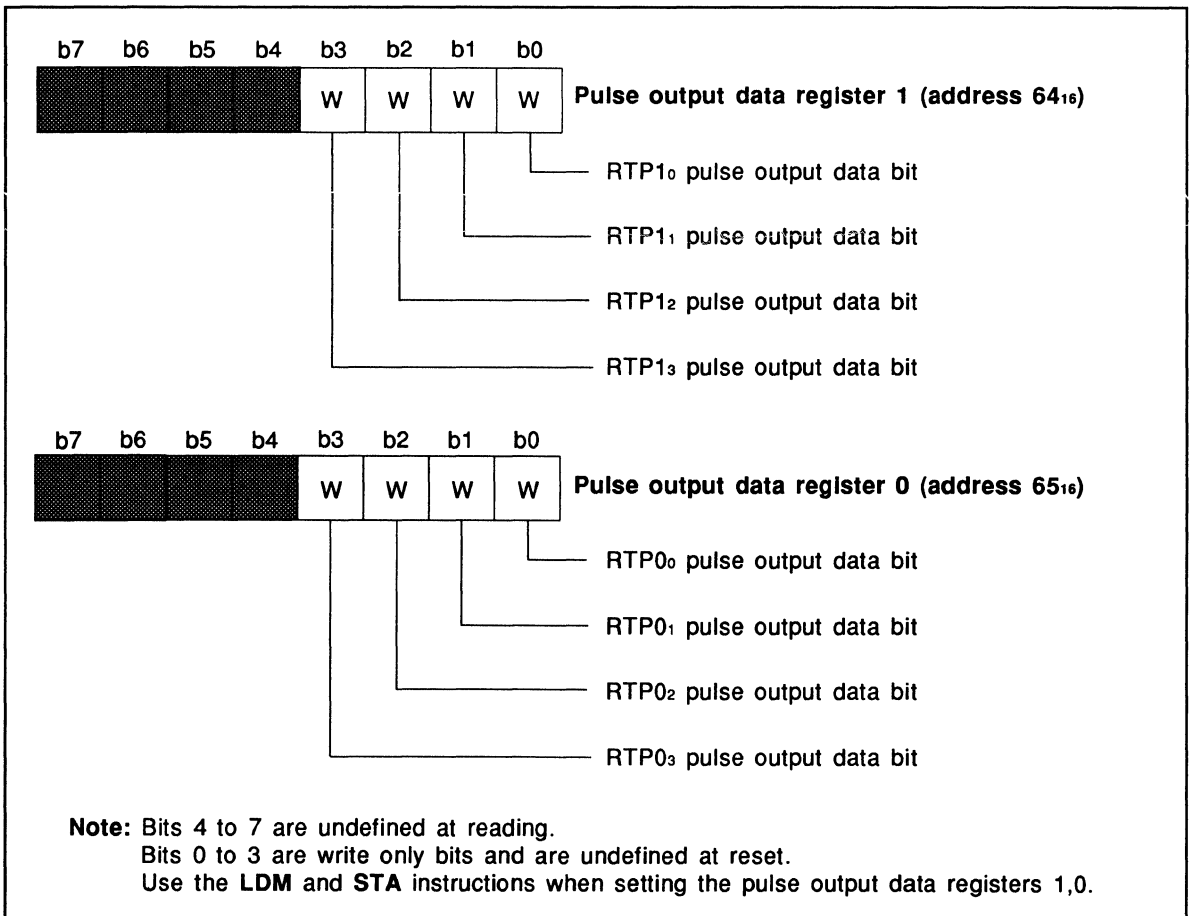


Fig. 2.9.8 Relationship between pulse output data register j and pulse output pins

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(6) Timer Ai Interrupt control register

The timer Ai interrupt control register (addresses 75₁₆ to 78₁₆) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.9.9 shows the structure of the timer Ai interrupt control register. The function of each bit is described below. Use the **SEB** and **CLB** instructions when setting the timer Ai interrupt control register. Refer to section "2.6 Interrupts" for more information.

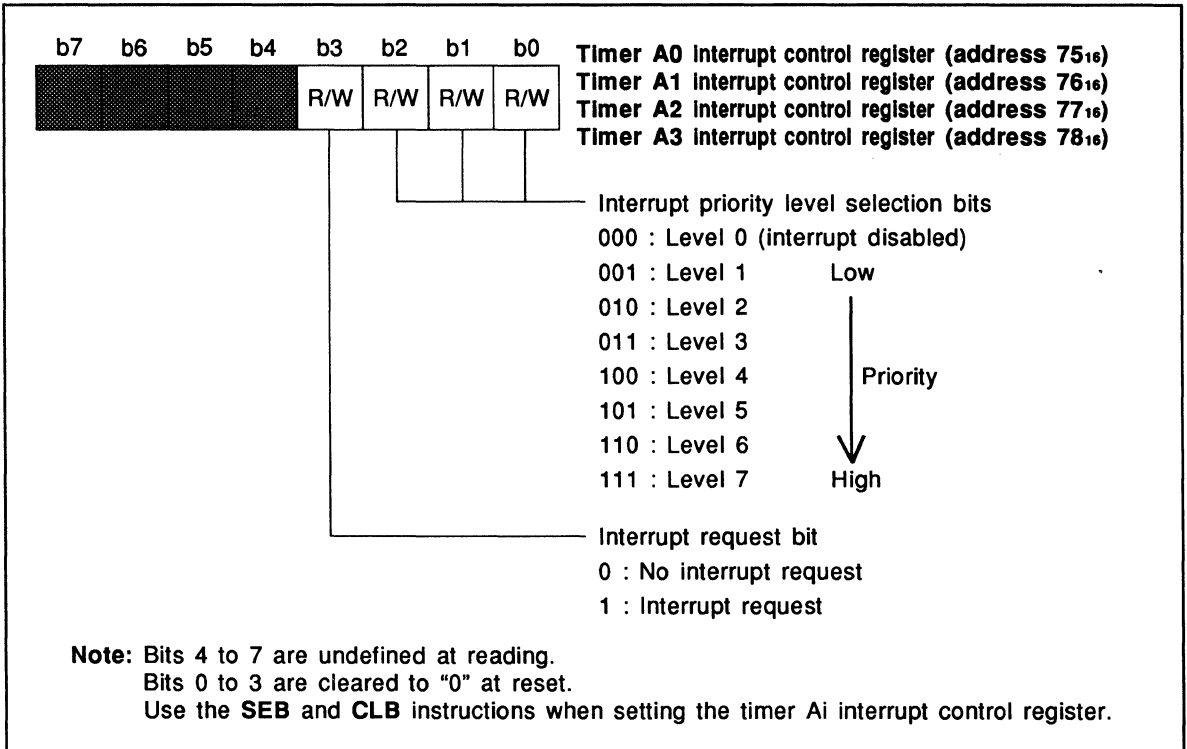


Fig. 2.9.9 Timer Ai interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using a timer Ai interrupt. When an interrupt request occurs, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is higher than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable only timer Ai interrupt.

●Interrupt request bit (bit 3)

This bit is set to "1" when a timer Ai interrupt request occurs. The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted.
This bit can also be set or cleared by program.

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

2.9.3 Operation description

The waveform output modes RTPj (j=0, 1) have identical functions and operate in the same manner except that RTP0 allows selection of pulse output polarity.

Timer Ak (k=0, 2) for pulse output control must be operated in timer mode. With timer mode used in pulse output port mode, the timer Ak mode register bit 2, which functions as pulse output function selection bit in normal timer A timer mode, must be fixed to "1" to select "pulse output". Similarly, bit 4 of the gate function selection bits must be fixed to "0" to select "no gate function".

Fix timer Ah (h=1, 3) mode register bit 2 to "1" regardless of whether pulse width modulation is used or not. By setting timer Ai mode register bit 2 to "1", the ports P5₀ (RTP0₀), P5₂ (RTP0₂), P5₄ (RTP1₀), and P5₆ (RTP1₂) are shared with timer A output pins function as pulse output pins.

Table 2.9.5 shows the waveform output mode.

Figure 2.9.10 shows the structure of pulse output pins and pulse output data register j.

Table 2.9.5 Waveform output mode

Waveform output mode	Pulse output port mode	
	RTP0	RTP1
Control timer	Timer A0	Timer A2
Pulse output pin	RTP0 ₀ –RTP0 ₃ (ports P5 ₀ –P5 ₃)	RTP1 ₀ –RTP1 ₃ (ports P5 ₄ –P5 ₇)
Register for setting pulse data	Pulse output data register 0	Pulse output data register 1
Pulse width modulation	Possible (timer A1 is used)	Possible (timer A3 is used)
Negative polarity pulse output function	Possible	Impossible

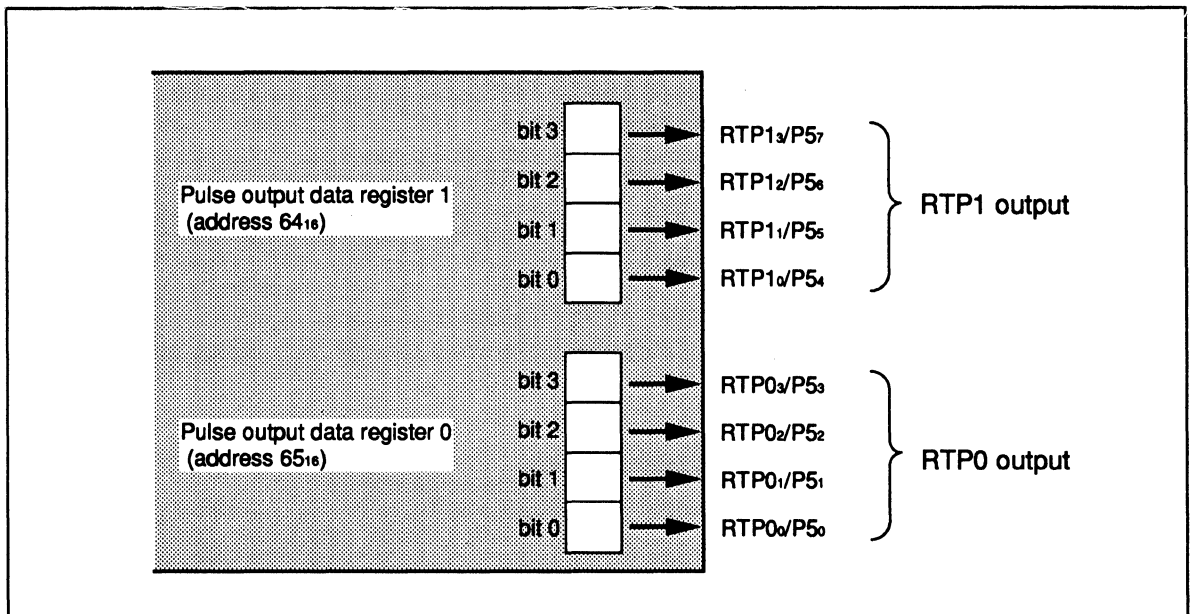


Fig. 2.9.10 Pulse output pins and pulse output data register j structure

(1) RTPj operation description (not modulated, RTP0 positive polarity)

The following is the description of RTPj operation.

First set the pulse output port mode related registers in the following sequence.

① Timer Ai mode register

●Timer A0, A2 mode registers

Select a count source and fix bits 2 to 0 to "100" and bits 5 and 4 to "00".

●Timer A1, A3 mode registers

Fix bit 2 to "1".

② Waveform output mode register

Select waveform output mode, RTP0 polarity, and pulse width modulation.

③ Pulse output data register j

Write data to be output.

④ Timer Ak (k=0,2) register

Write a value n ($n=0000_{16}$ to $FFFF_{16}$) and set the timer dividing ratio.

⑤ Count start flag

Timer corresponding to the bit set to "1" is started.

Set the timer Ai mode register as described above and then set the waveform output mode register. If the waveform output mode register is set (to waveform output mode) before setting the timer Ai mode register (to pulse output), the contents of port latch are output because ports P5₀, P5₂, P5₄, and P5₆ have the circuit structure shown in Figure 2.9.2.

When setting the above registers, the timer Ai mode register and waveform output mode register must be set when the corresponding count start flag is "0" (during timer halted).

For timer Ak (k=0,2) operated in timer mode, when the corresponding count start flag is set to "1", the selected count source is input to the counter and starts the count operation. The contents of the counter are decremented by 1 each time a count source is input. When the counter first underflows (contents of the counter change $0000_{16} \rightarrow$ reloaded value n), the contents of the pulse output data register j at that time are output from RTPj pin. At the same time, a timer Ak interrupt request occurs. The RTPj pin output level is undefined during the interval from the time RTPj is selected until the first time the counter underflows.

Timer Ak dividing ratio is expressed as follows:

$$\text{Timer Ak dividing ratio} = 1/(n+1)$$

n: Value set in counter
(value between 0000_{16} and $FFFF_{16}$)

The counter continues to operate and the pulse output continues at the state of the first underflow. When the counter underflows for the second time, the contents of the pulse output data register j at the time of the second underflow are output as pulse output. The counter continues to operate and pulse output continues in the same manner.

Therefore, when the contents of the pulse output data register j are changed during pulse output, the updated data is output next time the counter underflows after writing to the pulse output data register j. Figure 2.9.11 shows the pulse output example for RTP0 (positive polarity, not modulated).

Refer to section "2.7.3 Timer mode" for the operation description of timer Ak in timer mode.

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

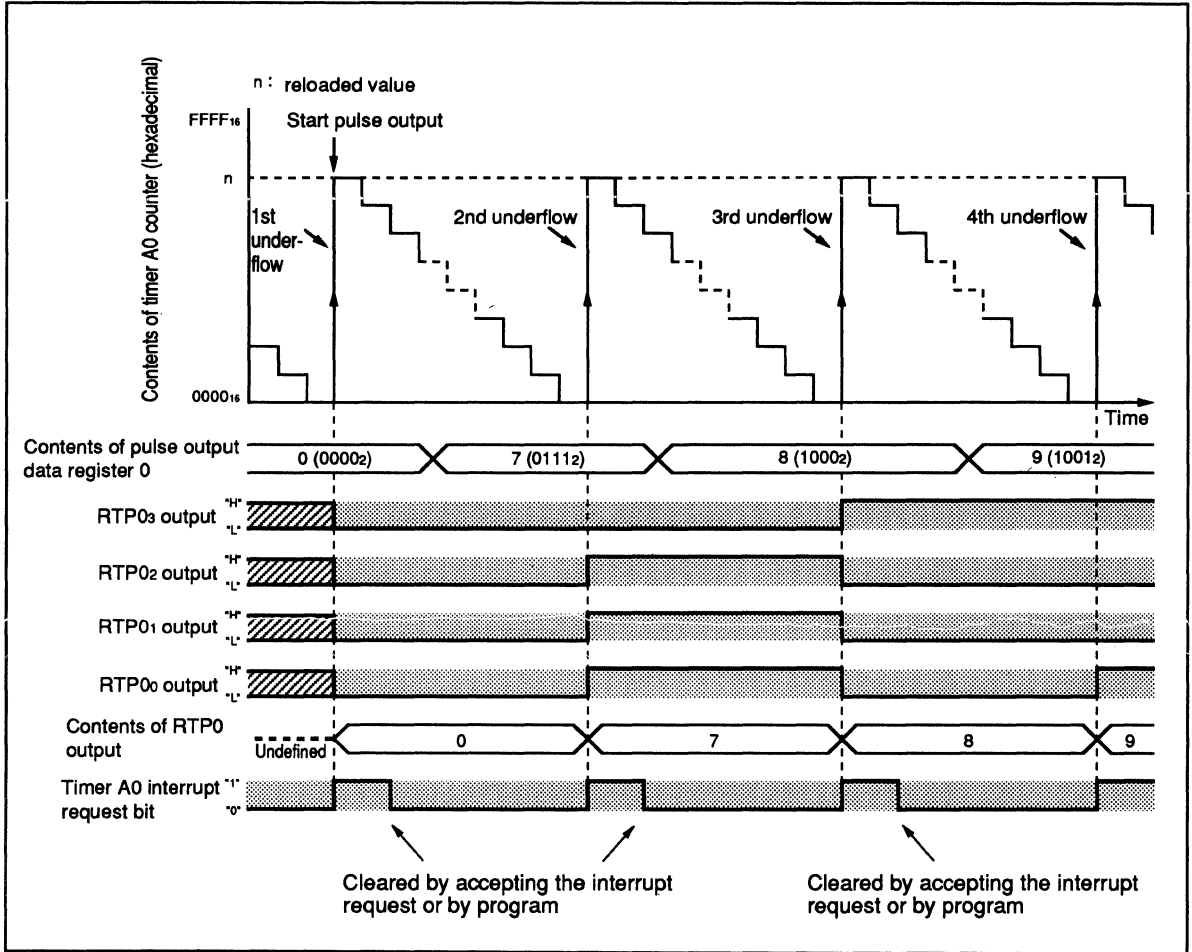


Fig. 2.9.11 RTP0 pulse output example (positive polarity, not modulated)

(2) RTPj operation description (modulated, RTP0 positive polarity)

The following is the description of RTPj operation with pulse width modulation selected.

Timer Ah (h=1,3) used for pulse width modulation must be operated in pulse width modulation (PWM) mode. With the PWM mode used for pulse output port mode, only the internal trigger occurred by setting the count start flag can be used. Except for this, the function is the same as with normal timer A in PWM mode.

Pulse width modulation set with timer Ah is performed while "H" level is output from the respective pins RTPj₀–RTPj₃.

First set the pulse output port mode related registers in the following sequence.

① Timer Ai mode register

●Timer A0, A2 mode registers

Select a count source and fix bits 2 to 0 to "100" and bits 5 and 4 to "00".

●Timer A1, A3 mode registers

Select 16/8-bit PWM mode, select count source, fix bits 2 to 0 to "111" and bit 4 to "0".

② Waveform output mode register

Select waveform output mode, RTP0 polarity, and pulse width modulation.

③ Pulse output data register j

Pulse from the pulse output pin corresponding to the bit set to "1" is modulated.

④ Timer Ai register

●Timer A0, A2 registers

Write a value n (n=0000₁₆ to FFFF₁₆) and set the timer dividing ratio.

●Timer A1, A3 registers

Write a value and set "H" level width of the PWM signal.

⑤ Count start flag

Timer corresponding to the bit set to "1" is started.

Set the timer Ai mode register as described above and then set the waveform output mode register.

If the waveform output mode register is set (to waveform output mode) before setting the timer Ai mode register (to pulse output), the contents of port latch are output because ports P5₀, P5₂, P5₄, and P5₆ have the circuit structure shown in Figure 2.9.2.

When setting the above registers, the timer Ai mode register and waveform output mode register must be set when the corresponding count start flag is "0" (during timer halted).

When Timer Ai operation starts, pulse width modulation is performed while "H" level is output from pins RTPj₀–RTPj₃.

When 16-bit PWM mode is selected, the pulse period and pulse width of the PWM signal are expressed as follows:

[16-bit PWM mode]

Pulse output period = $(1/f_i) \times (2^{16}-1)$ [s]

Pulse output "H" level width = $(1/f_i) \times n$ [s]

f_i: Selected count source frequency

n: Value set in timer A1, A3 counter

(value between 0000₁₆ and FFFE₁₆)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

When 8-bit PWM mode is selected, the pulse period and pulse width of the PWM signal are expressed as follows:

[8-bit PWM mode]

$$\text{Pulse output period} = (1/f_i) \times (m+1) \times (2^8-1) \text{ [s]}$$

$$\text{Pulse output "H" level width} = (1/f_i) \times (m+1) \times n \text{ [s]}$$

f_i : Selected count source frequency

n : Value set in high-order 8 bits of timer A1, A3 counter (value between 00_{16} and FE_{16})

m : Value set in low-order 8 bits of timer A1, A3 counter (value between 00_{16} and FF_{16})

An interrupt request of timer A_k ($k=0,2$) occurs each time the counter underflows. An interrupt request of timer A_h ($h=1,3$) occurs each time the PWM signal level changes from "H" level to "L" level. Figure 2.9.12 shows the pulse output example for RTP0 (positive polarity, modulated). Refer to section "2.7.6 Pulse width modulation (PWM) mode" for the operation description of timer A_h in PWM mode.

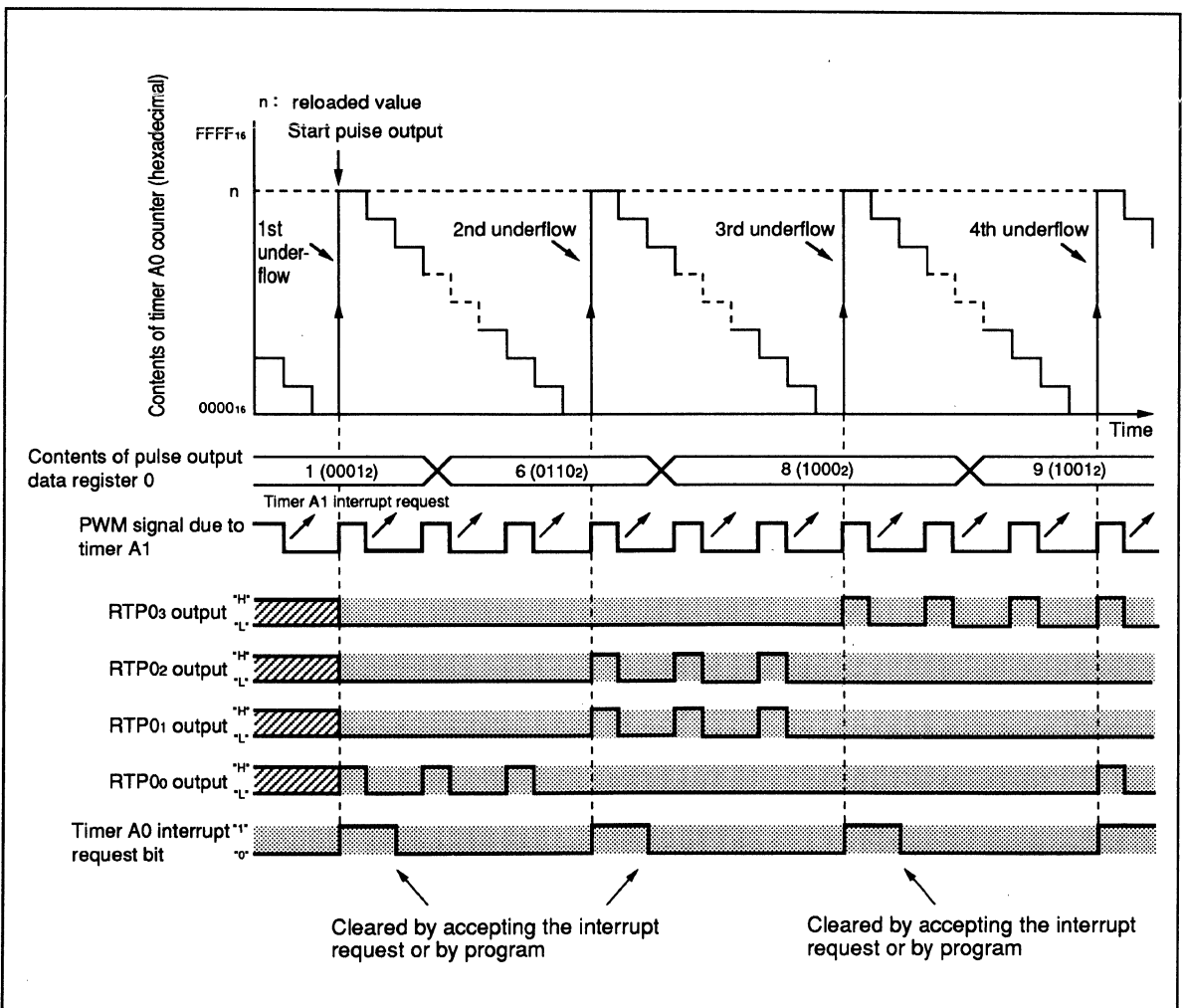


Fig. 2.9.12 RTP0 pulse output example (positive polarity, modulated)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

(3) RTP0 operation description (negative polarity selected)

When the RTP0 polarity selection bit of the waveform output mode register is set to "1", the inverse data (negative polarity pulse) of the contents of the pulse output data register 0 is output from RTP0. In this case, "H" level is output from the pulse output pin corresponding to the pulse output data bit set to "0" and "L" level is output from the pulse output pin corresponding to the pulse output data bit set to "1".

When pulse width modulation is performed with negative polarity selected, the output pulse is inverse of the case when pulse width modulation is performed with positive polarity.

Figure 2.9.13 shows the pulse output example for RTP0 (negative polarity, modulated).

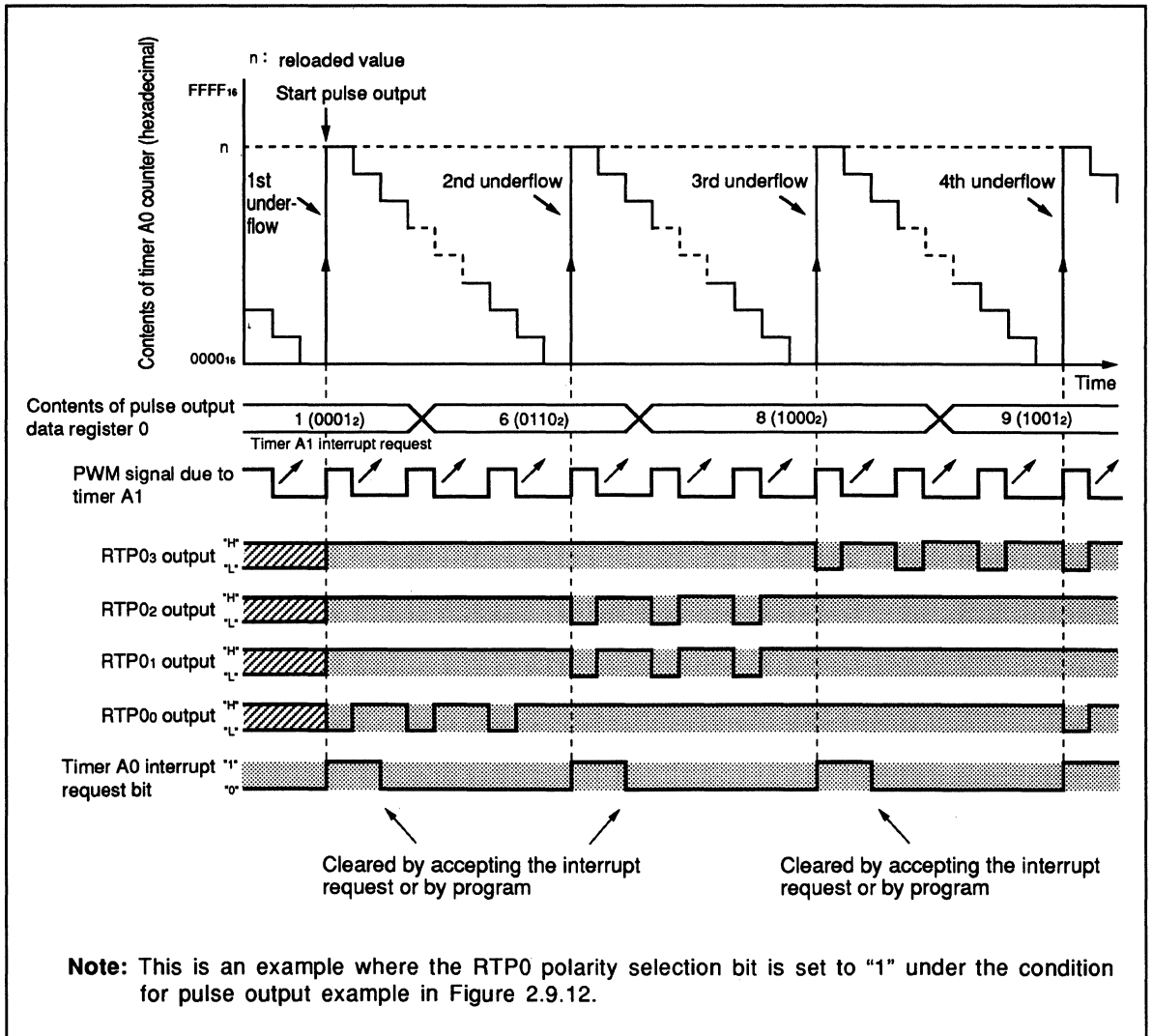


Fig. 2.9.13 RTP0 pulse output example (negative polarity, modulated)

FUNCTIONAL DESCRIPTION

2.9 Pulse output function

[Precautions when using pulse output function (pulse output port mode)]

1. Fix timer A0 to A3 mode registers bit 2 to "1" in order to use ports P5₀ (RTP0₀), P5₂ (RTP0₂), P5₄ (RTP1₀), and P5₆ (RTP1₂) as pulse output pin.

●When using RTP0: Fix timer A0, A1 mode registers bit 2 to "1".

●When using RTP1: Fix timer A2, A3 mode registers bit 2 to "1".

2. When pulse width modulation is not performed, timers A1 and A3 can be used in timer mode. In this case, bit 4 must be fixed to "0" because the gate function cannot be used. Bit 5 must be also fixed to "0". Therefore, timer A1, A3 mode registers can be only used to select the count source.

2.10 Serial I/O

Serial I/O consists of UART0 and UART1 that have same functions. These serial I/O can operate either as clock synchronous serial I/O port or asynchronous serial I/O (UART) port.

2.10.1 Serial I/O description

UART0 and UART1 can operate either as clock synchronous serial I/O port or asynchronous serial I/O (UART) port. These two serial I/O ports are independent, but have identical functions. Each serial I/O port has a transfer clock generation timer (baud rate generator referred to BRG) and can be set a variety of data transfer rate.

Figure 2.10.1 shows the serial I/O operating modes.

Each serial I/O has four operating modes. The following modes are available:

●**Clock synchronous serial I/O**

In this mode, both the transmission side and receiving side use the same clock to transfer data. The data (character) length is 8 bits.

●**7-bit UART**

In this mode, the data is transferred at an arbitrary rate and data format. The data (character) length is 7 bits.

●**8-bit UART**

This mode is identical to 7-bit UART except that the data length is 8 bits.

●**9-bit UART**

This mode is identical to 7-bit UART except that the data length is 9 bits.

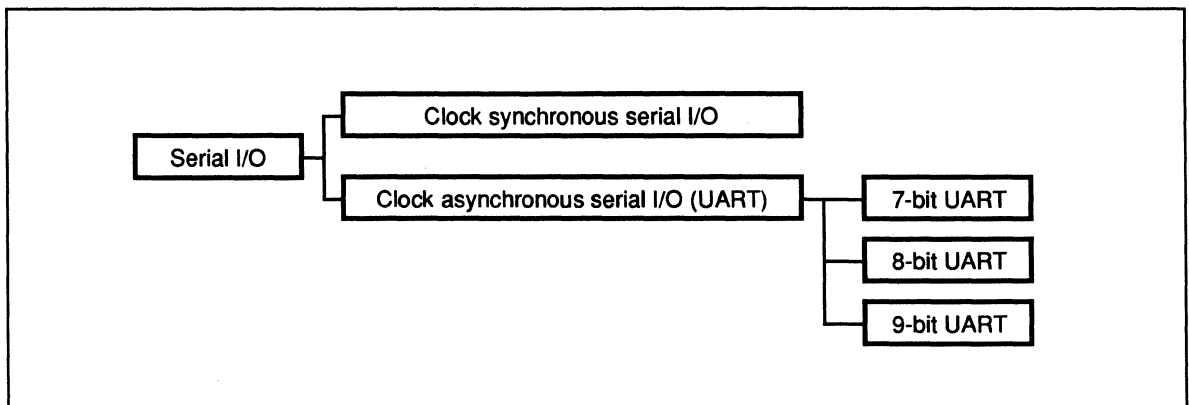


Fig. 2.10.1 Serial I/O operating modes

2.10.2 Block description

Figure 2.10.2 shows the block diagram of serial I/O. The function of each related registers are described below.

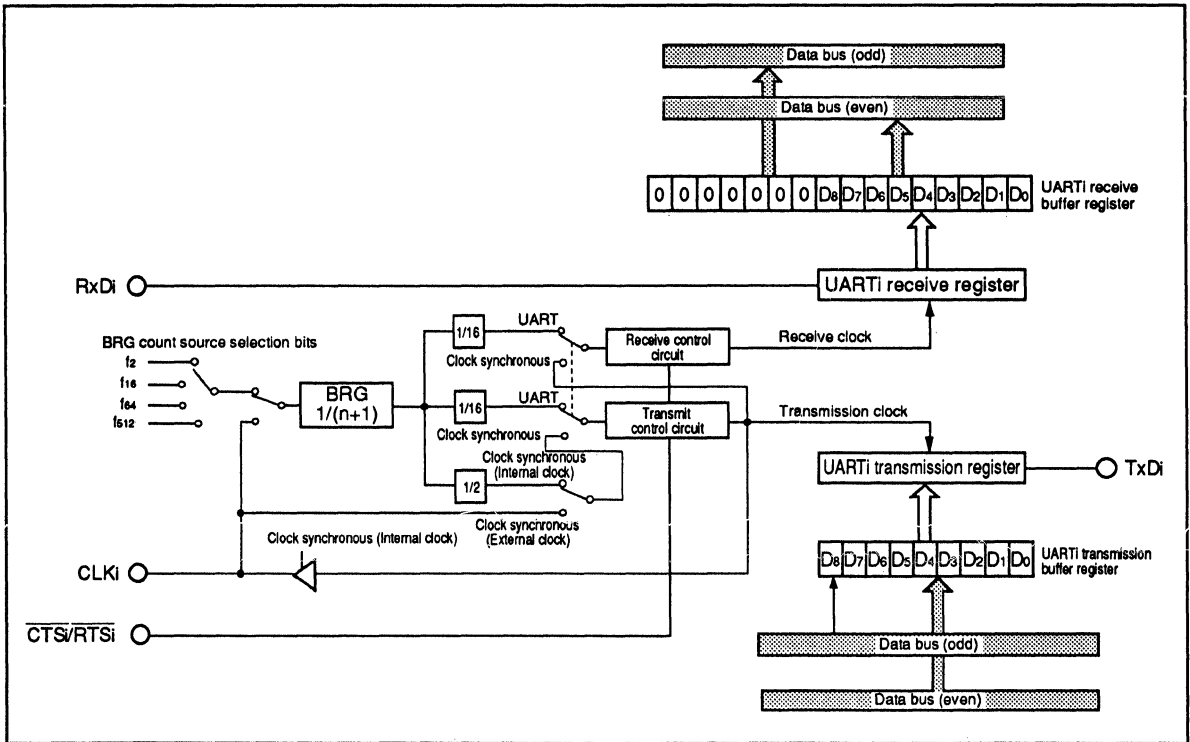


Fig. 2.10.2 Serial I/O block diagram

(1) UARTi transmit/receive mode register (i=0, 1)

The UART0 transmit/receive mode register (address 30₁₆) and the UART1 transmit/receive mode register (address 38₁₆) consist of bits to set serial I/O modes, transferring data format.

Figure 2.10.3 shows the structure of the UARTi transmit/receive mode register.

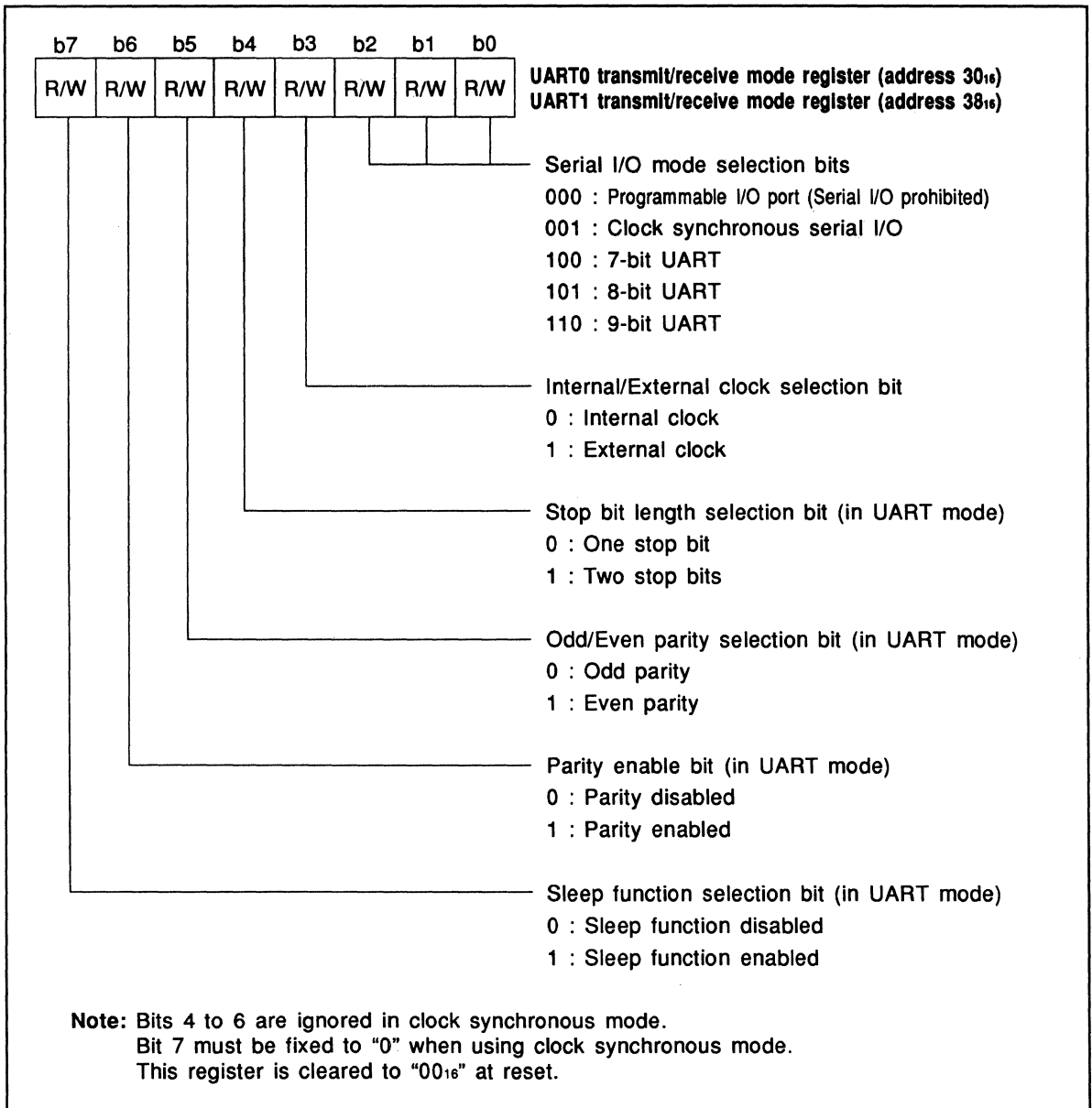


Fig. 2.10.3 UARTi transmit/receive mode register structure

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

●Serial I/O mode selection bits (bits 0 to 2)

These bits are used to select serial I/O modes.

Table 2.10.1 shows the relationship between the serial I/O mode selection bits and serial I/O modes. When bits 2 to 0 are set to "000", serial I/O is disabled and ports P8₀–P8₃, and P8₄–P8₇ function as programmable I/O ports. When one of the serial I/O modes is selected, port P8 has the function shown in Table 2.10.2 and loses its programmable I/O port function (except for some pins in UART mode).

Table 2.10.1 Relationship between serial I/O mode selection bits and serial I/O modes

b2	b1	b0	Serial I/O mode selection bits
0	0	0	Programmable I/O port (Serial I/O prohibited)
0	0	1	Clock synchronous serial I/O
0	1	0	Not available
0	1	1	Not available
1	0	0	7-bit UART
1	0	1	8-bit UART
1	1	0	9-bit UART
1	1	1	Not available

Table 2.10.2 Function of port P8 when serial I/O is selected

Using UART0	Using UART1	Function
P8 ₀	P8 ₄	CTS/RTS (transmission control signal I/O pin)
P8 ₁	P8 ₅	CLK (transfer clock I/O pin) (Note 1)
P8 ₂	P8 ₆	RxD (serial data input pin)
P8 ₃	P8 ₇	TxD (serial data output pin) (Note 2)

Note 1: This depends on the internal/external clock selection bit as follows:

When an external clock is selected : Clock input pin

When an internal clock is selected : •Clock output pin in clock synchronous mode

•Normal I/O port in UART mode

Note 2: TxD pin starts to output "H" level when one of serial I/O modes is selected.

●Internal/External clock selection bit (bit 3)

[Clock synchronous mode]

This bit is used to select either an internal clock or an external clock as the synchronous clock (shift clock) for data transfer.

When this bit is set to "0" to select an internal clock, the divided clock by 2 which the later described baud rate generator (BRG) generates is used as the shift clock. In addition, the CLK_i pin becomes the output pin and the shift clock is output from this pin.

When this bit is set to "1" to select an external clock, the CLK_i pin becomes the input pin and data transfer is synchronized with the clock input to this pin.

[UART mode]

This bit is used to select either an internal clock or an external clock as the input clock to the BRG which is described later.

When this bit is set to "0" to select an internal clock, the clock selected with the later described BRG count source selection bits in the UART_i transmit/receive control register 0 becomes the BRG input clock. In this case, the CLK_i pin (ports P8₁, P8₅) can be used as a programmable I/O port.

When this bit is set to "1" to select an external clock, the CLK_i pin becomes the clock input pin and the clock input to this pin becomes the BRG input clock.

●Stop bit length selection bit (bit 4)

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to select between 1 and 2 bits as the stop bit to indicate the end of data.

1 stop bit is selected when this bit is "0". 2 stop bits are selected when this bit is "1".

●Odd/Even parity selection bit (bit 5)

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to select between even parity and odd parity. Odd parity is selected when this bit is "0", and even parity is selected when this bit is "1".

This bit is valid if the parity enable bit (bit 6) is set to "1" (enabled).

●Parity enable bit (bit 6)

[Clock synchronous mode]

This bit is ignored. It can be either "0" or "1".

[UART mode]

This bit is used to specify whether to add a parity bit at the end of transmitted data or not (whether to perform parity check of received data or not). Whether to use odd parity or even parity is specified with bit 5.

When this bit is "1", a parity is added at transmitting, and parity check is performed at receiving.

●Sleep function selection bit (bit 7)

[Clock synchronous mode]

This bit must be fixed to "0".

[UART mode]

This bit is used to enable or disable the sleep function. If this bit is set to "1" to enable the sleep function, the data is ignored when the most significant bit (MSB) of the received data is "0".

This function is used when multiple microcomputers are connected through the serial I/O port. Refer to section "2.10.5 Sleep mode".

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

(2) UARTi transmit/receive control register 0

The UART0 transmit/receive control register 0 (address 34₁₆) and the UART1 transmit/receive control register 0 (address 3C₁₆) consist of bits to select the BRG count source and $\overline{\text{CTS}}/\overline{\text{RTS}}$ function, and a flag that indicates the UARTi transmission register state.

Figure 2.10.4 shows the structure of the UARTi transmit/receive control register 0.

●BRG count source selection bits (bits 0 and 1)

This bit is used to select the count source of the BRG when an internal clock is selected. Table 2.10.3 shows the relationship between the BRG count source selection bits and the count source.

Table 2.10.3 Relationship between BRG count source selection bits and count source

b1	b0	BRG count source
0	0	f_2 selected which is $f(X_{IN})$ divided by 2
0	1	f_{16} selected which is $f(X_{IN})$ divided by 16
1	0	f_{64} selected which is $f(X_{IN})$ divided by 64
1	1	f_{512} selected which is $f(X_{IN})$ divided by 512

$f(X_{IN})$: External clock input frequency

● $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit (bit 2)

This bit is used to select $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ function. Port P8₀ functions as $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin for UART0, port P8₄ as $\overline{\text{CTS}}/\overline{\text{RTS}}$ pin for UART1.

When this bit is "0", $\overline{\text{CTS}}$ function is selected. Port P8₀ or P8₄ functions as the $\overline{\text{CTS}}$ input pin. This pin must be at "L" level in order for transmission to start.

When this bit is "1", $\overline{\text{RTS}}$ function is selected. Port P8₀ or P8₄ functions as the $\overline{\text{RTS}}$ output pin. "H" level is output when receive is disabled (the receive enable bit in UARTi transmit/receive control register 1 is "0"). "L" level is output when receive is enabled (the receive enable bit is "1"). It returns to "H" level when receive starts and "L" level is output when receive completes.

●Transmission register empty flag (bit 3)

This flag is set to "0" when the contents of UARTi transmission buffer register are transferred to the UARTi transmission register. It is set to "1" when transmission completes and the UARTi transmission register becomes empty. This flag is a read-only flag and set to "1" at reset.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

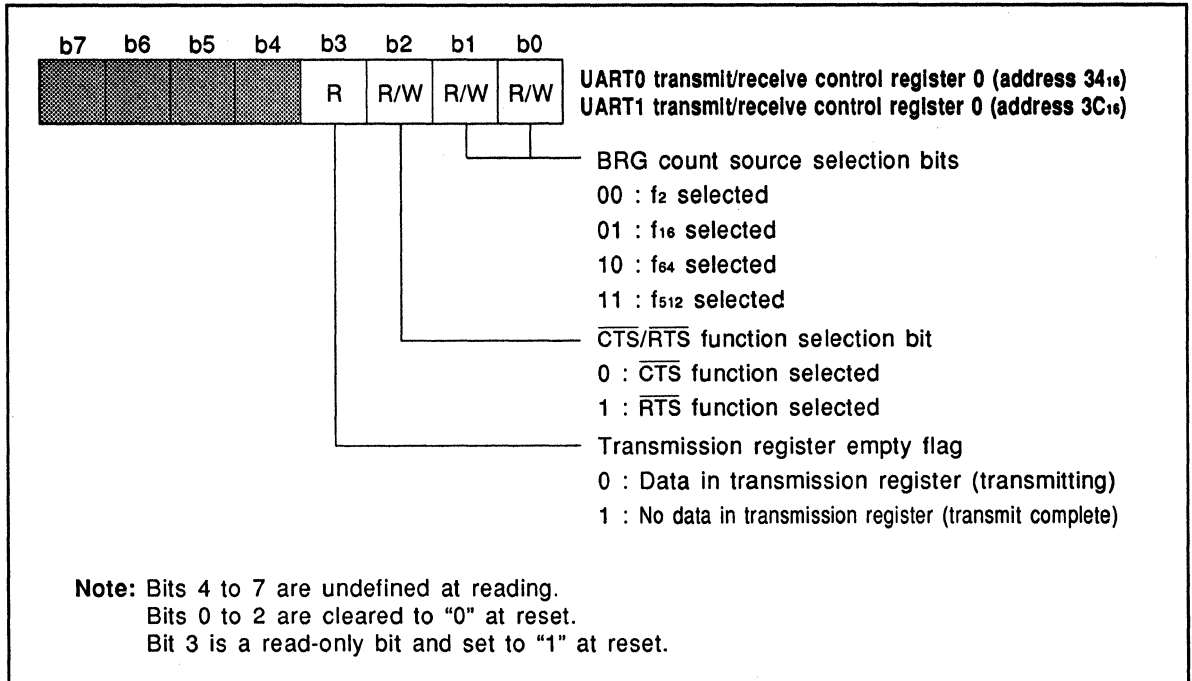


Fig. 2.10.4 UARTI transmit/receive control register 0 structure

FUNCTIONAL DESCRIPTION

(3) UARTi transmit/receive control register 1

The UART0 transmit/receive control register 1 (address 35₁₆) and the UART1 transmit/receive control register 1 (address 3D₁₆) consist of serial I/O enable bits, serial I/O state flags, and serial I/O error flags. Figure 2.10.5 shows the structure of the UARTi transmit/receive control register 1.

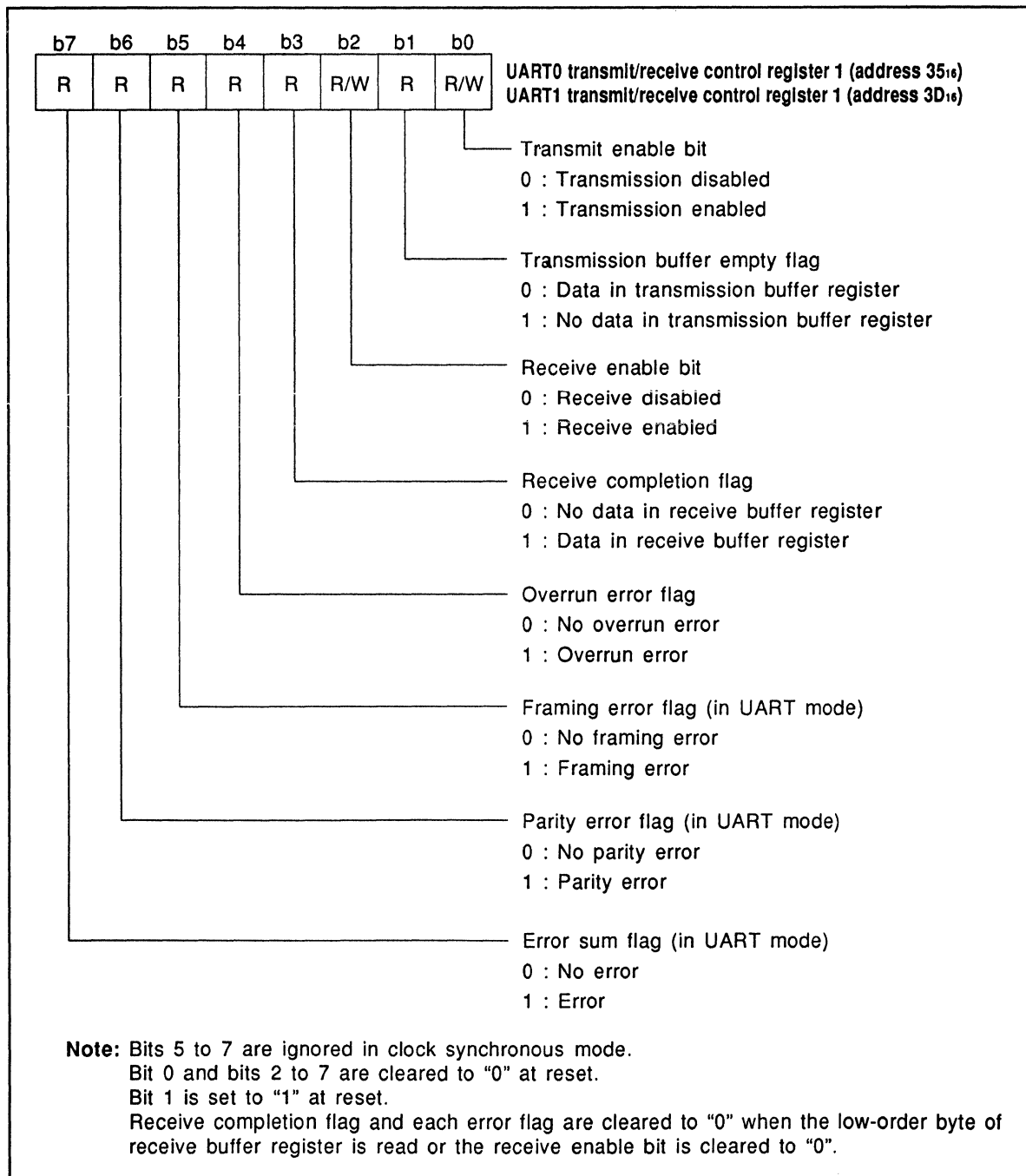


Fig. 2.10.5 UARTi transmit/receive control register 1 structure

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

● Transmit enable bit (bit 0)

Serial I/O transmission is enabled when this bit is set to "1". If this bit is set to "0" during transmitting, serial I/O transmission is disabled after the current transmission completes.

● Transmission buffer empty flag (bit 1)

This flag indicates the state of the transmission buffer register. This bit is set to "1" when the contents of the transmission buffer register are transferred to the transmission register. This flag is automatically cleared to "0" when data is written in the transmission buffer register.

● Receive enable bit (bit 2)

Serial I/O reception is enabled when this bit is set to "1". If the $\overline{\text{RTS}}$ function is selected, the $\overline{\text{RTS}}$ pin becomes "L" level when this bit is set to "1".

● Receive completion flag (bit 3)

This flag is set to "1" when the data in the receive register is transferred to the receive buffer register (receive completion). This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

● Overrun error flag (bit 4)

This flag is set to "1" when receiving of the next data completes and the contents of the receive buffer register are updated while there is data remaining in the receive buffer register (before the contents of the receive buffer register are read).

This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

● Framing error flag (bit 5)

[Clock synchronous mode]

This flag is ignored.

[UART mode]

This flag is set to "1" when the number of stop bits is not the number specified with stop bit length selection bit (bit 4) of the UARTi transmit/receive mode register. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

● Parity error flag (bit 6)

[Clock synchronous mode]

This flag is ignored.

[UART mode]

This flag is set to "1" when the parity odd/even is not the one specified. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

● Error sum flag (bit 7)

[Clock synchronous mode]

This flag is ignored.

[UART mode]

This flag is set to "1" when either an overrun error, a framing error, or a parity error occurs. This flag is cleared to "0" when the low-order byte of the receive buffer register is read or when the receive enable bit is set to "0" (receive disabled).

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

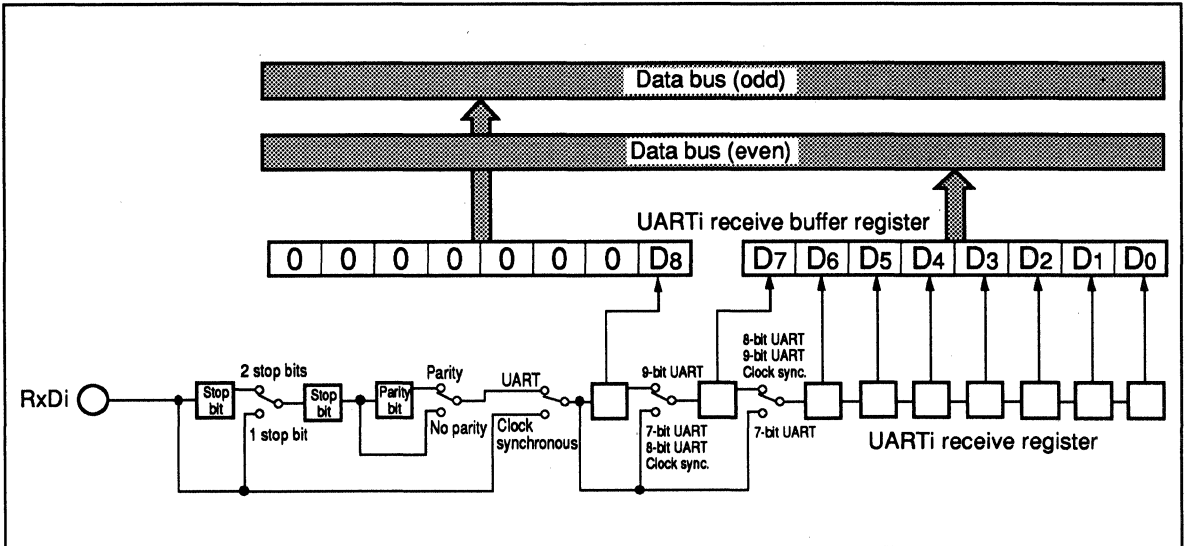


Fig. 2.10.7 Serial I/O receive block diagram

(6) UARTi baud rate register

The baud rate generator (referred to BRG) is the 8-bit timer exclusively for serial I/O equipped with a reload register.

The BRG divides the input clock by $(n+1)$, where "n" is the value set in the UART0 baud rate register (address 31₁₆) and the UART1 baud rate register (address 39₁₆). This register can contain a value between 00₁₆ and FF₁₆.

In clock synchronous serial I/O mode, the BRG becomes effective when an internal clock is selected and the BRG output divided by 2 becomes the synchronous clock for transmit/receive.

In UART mode, the BRG is effective regardless of the clock type and the BRG output divided by 16 becomes the transmit/receive clock.

The contents of the UARTi baud rate register can not be read because it is a write-only register. This register is undefined at reset.

Figure 2.10.8 shows the block diagram of shift clock generation.

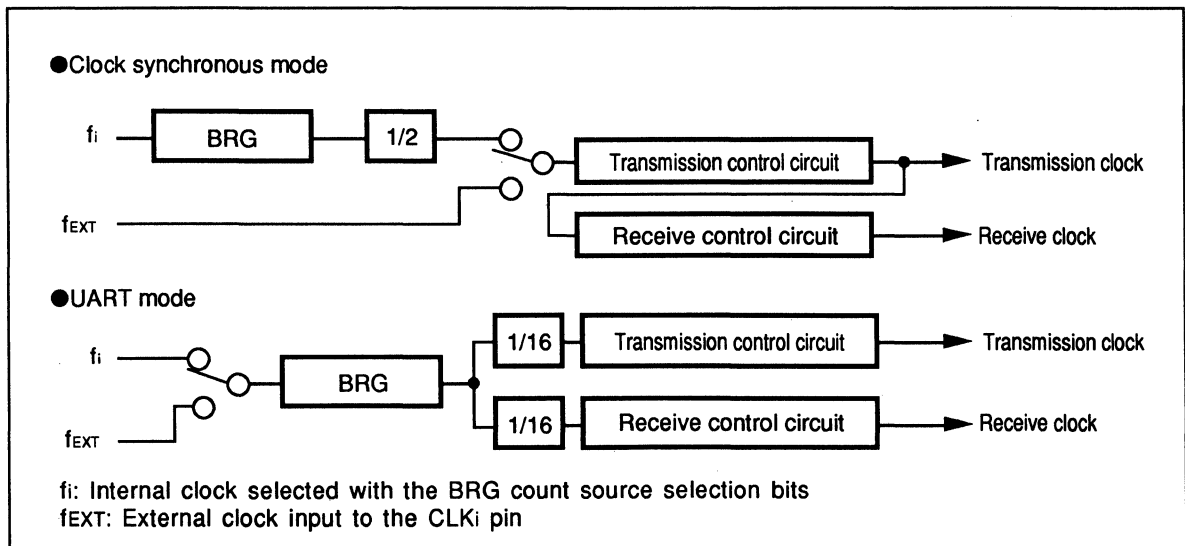


Fig. 2.10.8 Shift clock generation block diagram

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

(7) UARTi transmission interrupt control register and UARTi receive interrupt control register

Transmit interrupt and receive interrupt can be used when the serial I/O mode is selected. Each interrupt has an interrupt control register. Interrupt control register consists of interrupt priority level selection bits and interrupt request bit. Figure 2.10.9 shows the structure of UARTi transmission interrupt control register and UARTi receive interrupt control register and each bit is described below. Use the **SEB** and **CLB** instructions when setting each interrupt control register. Refer to section "2.6 Interrupts" for more information.

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using serial I/O interrupts. When an interrupt request occurs, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS). The interrupt is allowed only when this level is higher than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable an interrupt.

●Interrupt request bit (bit 3)

The transmit interrupt request bit is set to "1" when data is transferred from the UARTi transmission buffer register to the UARTi transmission register for data transmission. The receive interrupt request bit is set to "1" when data receive completes and data is transferred from the UARTi receive register to the UARTi receive buffer register. The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted. This bit can be set or cleared by program.

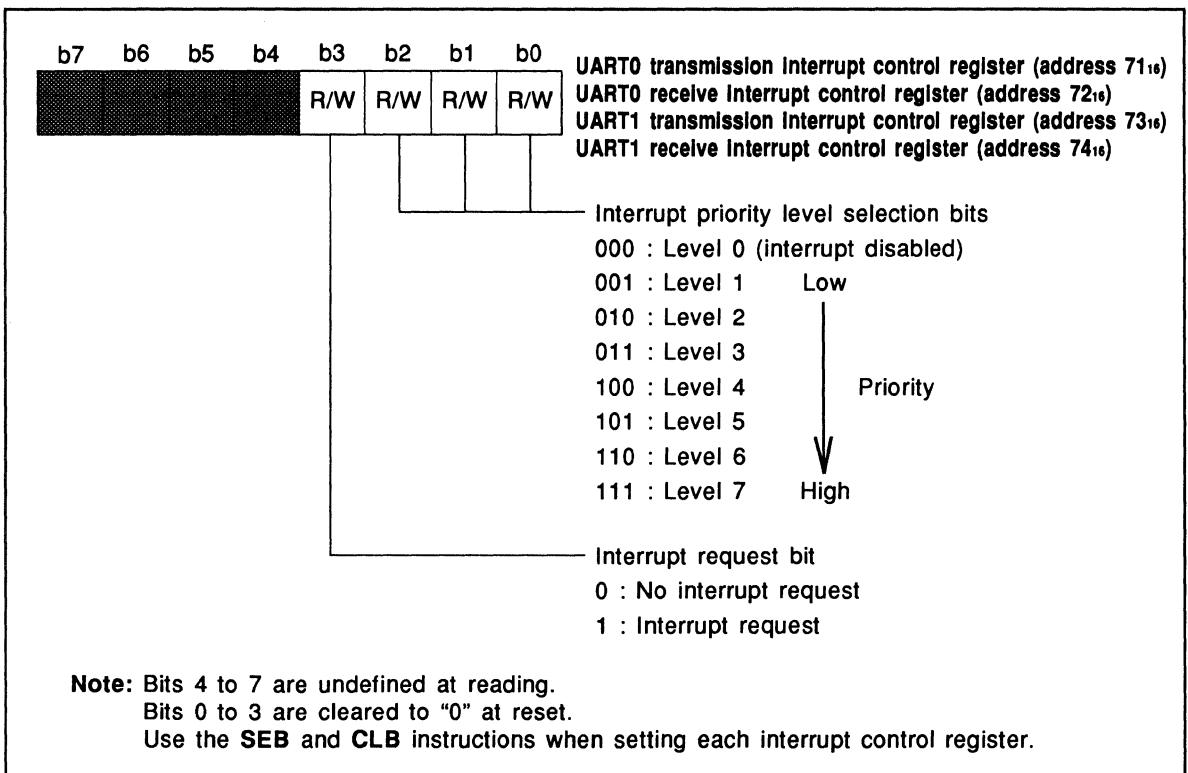


Fig. 2.10.9 UARTi transmission interrupt control register and UARTi receive interrupt control register structure

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

(8) Port P8 direction register

Serial I/O input/output pins are shared with port P8. Port P8 function is selected by the serial I/O mode selection bits of the UART_i transmit/receive mode register. When using port P8 as serial I/O input pins, the corresponding bit in the direction register must be set to "0" (input mode). When using port P8 as serial I/O output pins, it functions as serial I/O output pins regardless of the direction register.

Figure 2.10.10 shows the relationship between the port P8 direction register (address 14₁₆) and the serial I/O pins.

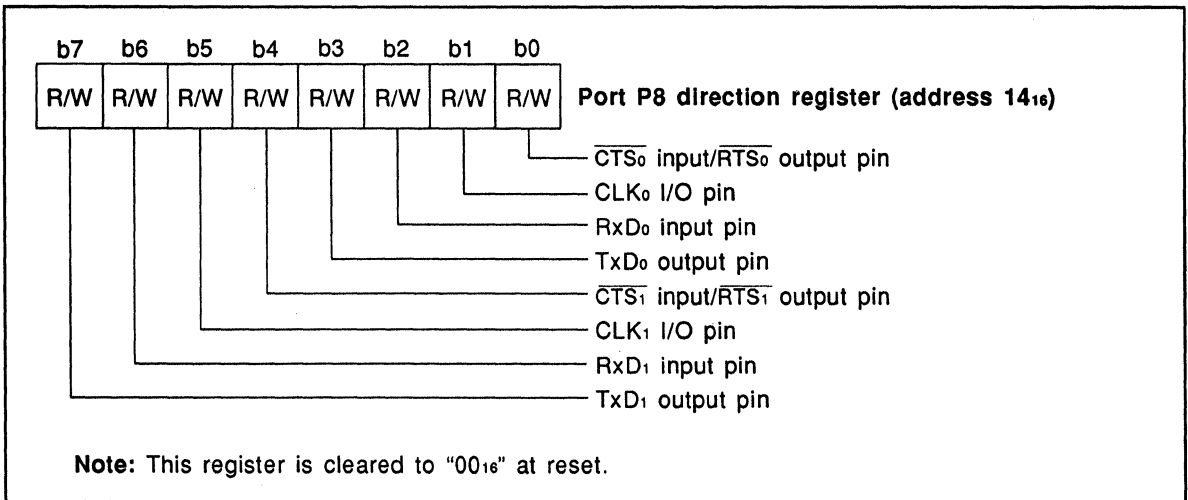


Fig. 2.10.10 Relationship between port P8 direction register and serial I/O pins

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

2.10.3 Clock synchronous serial I/O

Table 2.10.4 shows the performance overview of the clock synchronous serial I/O mode.

Table 2.10.4 Clock synchronous serial I/O performance overview

Parameter		Function
Data format		8 bit fixed, LSB* first
Transfer rate	Internal clock	BRG output divided by 2
	External clock	2Mbps maximum (at f(X _{IN})=8MHz)
		4Mbps maximum (at f(X _{IN})=16MHz)
	5Mbps maximum (at f(X _{IN})=25MHz)	
Transmit/receive control		CTS input or RTS output can be selected by program.

* LSB=Least Significant Bit

(1) Synchronous clock (shift clock)

The serial I/O data transfer rate is determined by the synchronous clock (shift clock). The M37732 group can select whether to generate this clock internally or to use an external input clock as the synchronous clock. The synchronous clock is generated internally when the UART_i transmit/receive mode register bit 3 is set to "0", and externally when it is set to "1".

In clock synchronous mode, the synchronous clock used for data transfer is generated by activating the transmitter. Therefore, the transmitter must be activated even when performing receive only.

●Using Internal generation clock as synchronous clock

When the internal/external clock selection bit is set to "0", the BRG output divided by 2 is used as the synchronous clock. In this case, the CLK_i pin becomes output mode and the transmit/receive synchronous clock is output from the CLK_i pin.

The BRG is a serial I/O timer which has an 8-bit structure and is used as a frequency divider to generate the desired frequency. The BRG divides the clock selected with bits 0 and 1 in the UART_i transmit/receive control register 0 by (n+1). The synchronous clock is the divided clock by 2 which has been divided by (n+1) with the BRG. "n" is the value set in the UART_i baud rate register. It can be set a value between 00₁₆ and FF₁₆.

Synchronous clock frequency $f_i / (2(n+1))$

f_i : BRG input frequency (i=2, 16, 64, 512)

8 synchronous clocks are generated by activating the transmitter.

●Using external input clock as synchronous clock

When the internal/external clock selection bit is set to "1", the external clock is used as the synchronous clock.

When an external clock is selected, the clock input to the CLK_i pin becomes the synchronous clock. Set the port P8 direction register bit 1 (CLK₀) and bit 5 (CLK₁) to "0" to select input mode.

(2) Serial data transmission

The data transmission method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

Set each serial I/O control register for transmission.

●UARTI transmit/receive mode register

- Serial I/O mode selection bits

Set the bits 2 to 0 to "001".

- Internal/External clock selection bit

Set the bit 3 to "0" when an internal clock is selected, or "1" when an external clock is selected.

- Setting the bit 7 to "0" (disable sleep mode)

●UARTI transmit/receive control register 0

- BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for synchronous clock.

- $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit

Set the bit 2 to "0" when using $\overline{\text{CTS}}$ function, and to "1" when not use.

●UARTI baud rate register

- Dividing ratio

Set the BRG dividing ratio between 00_{16} and FF_{16} when an internal clock is selected for synchronous clock.

●Port P8 direction register

- Port direction selection bits

Set the corresponding bit to "0" when the $\overline{\text{CTS}}$ function is selected and an external clock is selected.

●UARTI transmission interrupt control register

- Interrupt priority level selection bits

When using UARTi transmission interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTI transmission buffer register

- Transfer data

Set a transfer data to the low-order byte of UARTi transmission buffer register. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTI transmit/receive control register 1

- Transmit enable bit

Set the bit 0 to "1" to enable transmitting.

Figure 2.10.11 shows the setting example of clock synchronous serial I/O related registers at transmitting.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

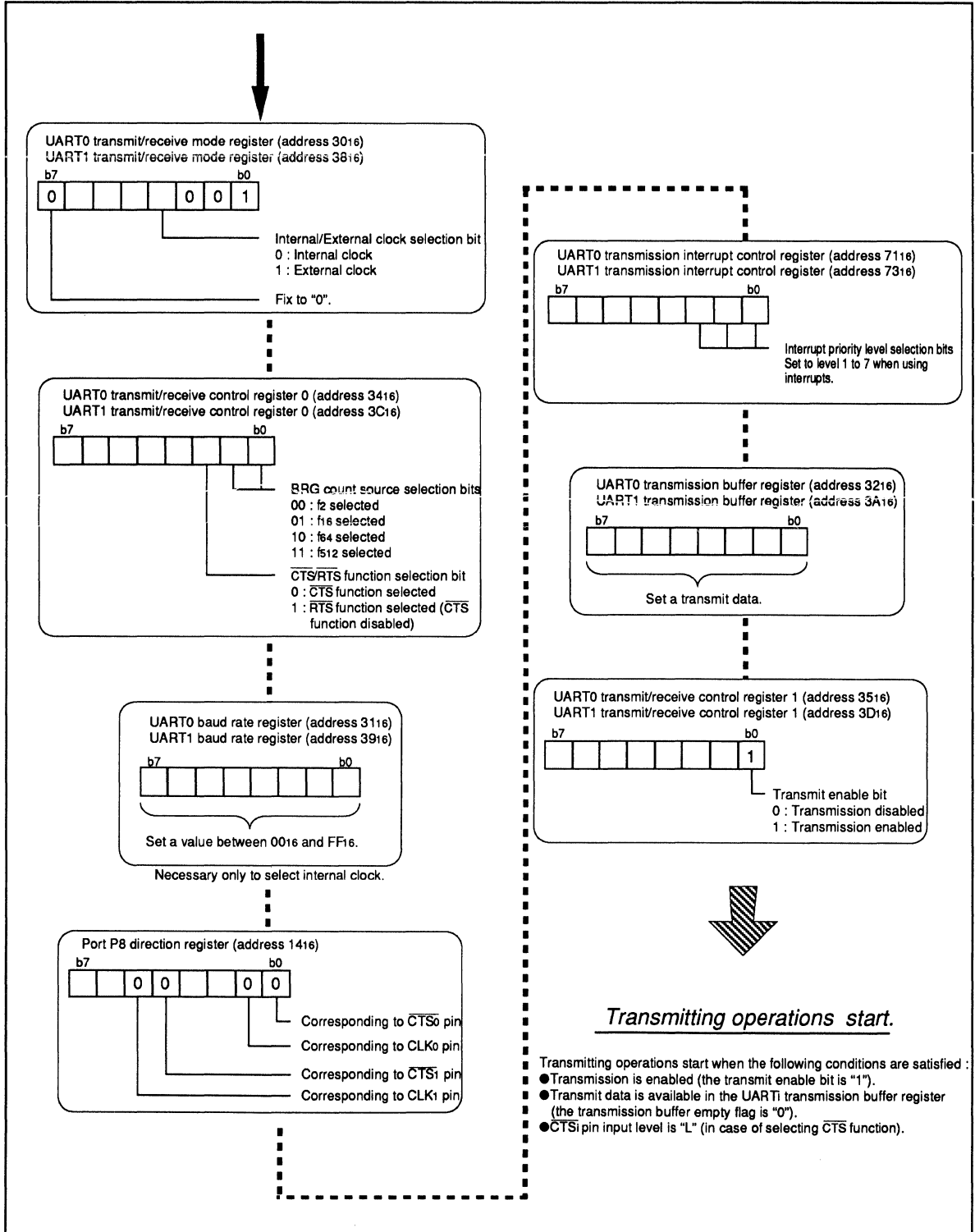


Fig. 2.10.11 Setting example of clock synchronous serial I/O related registers at transmitting

[Transmit operation]

The transmission of serial data starts when the following conditions are satisfied :

- ① Transmission is enabled (the transmit enable bit is "1").
- ② Transmit data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").
- ③ $\overline{\text{CTS}}_i$ pin input level is "L".
(Note: This condition is ignored if the $\overline{\text{CTS}}$ function is not selected.)

When the above three (① to ③) conditions are satisfied (two (① and ②) if $\overline{\text{CTS}}$ function is not selected), the following operations are performed automatically at the same time :

- Transfer the contents of UARTi transmission buffer register to the UARTi transmission register.
- Generate 8 shift clocks.
- Set the transmission buffer empty flag to "1".
- Clear the transmission register empty flag to "0".
- UARTi transmission interrupt request occurs and set the interrupt request bit to "1" .

Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information.

The shift clock is input to the transmit control circuit through the CLKi pin. The data in the UARTi transmission register is transmitted bit by bit from the TxDi pin (starting at the low-order bit) at each falling edge of shift clock (LSB first). When the 1-byte data transmission is completed by the 8 shift clocks, the transmission register empty flag is set to "1". Figure 2.10.12 shows the clock synchronous serial I/O transmit operation.

The synchronous clock is generated continuously if the conditions for the next data are satisfied when the transmission completes. Therefore, to transmit data continuously, the next data must be written in the UARTi transmission buffer register while data is being transmitted (when the transmission register empty flag is "0"). If the conditions to transmit the next data are not satisfied, the synchronous clock halts at "H" level.

Figure 2.10.13 shows the timing diagram of clock synchronous serial I/O at transmitting (internal clock is selected as synchronous clock, $\overline{\text{CTS}}$ function is selected).

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

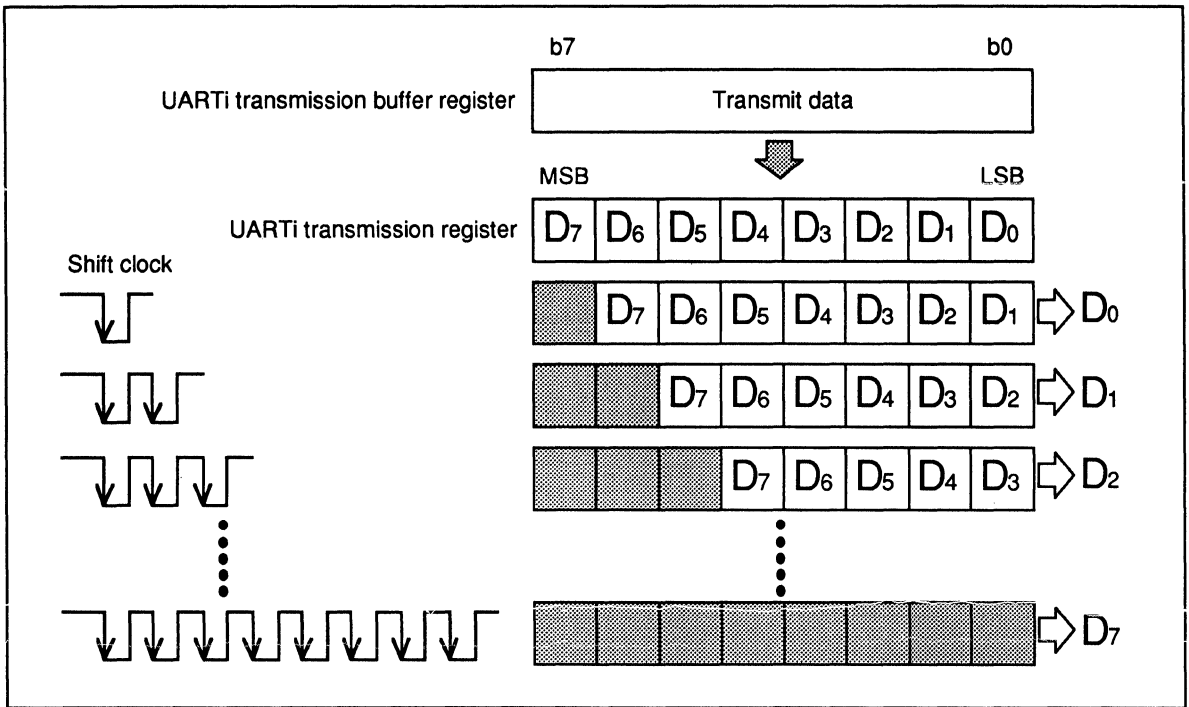


Fig. 2.10.12 Clock synchronous serial I/O transmit operation

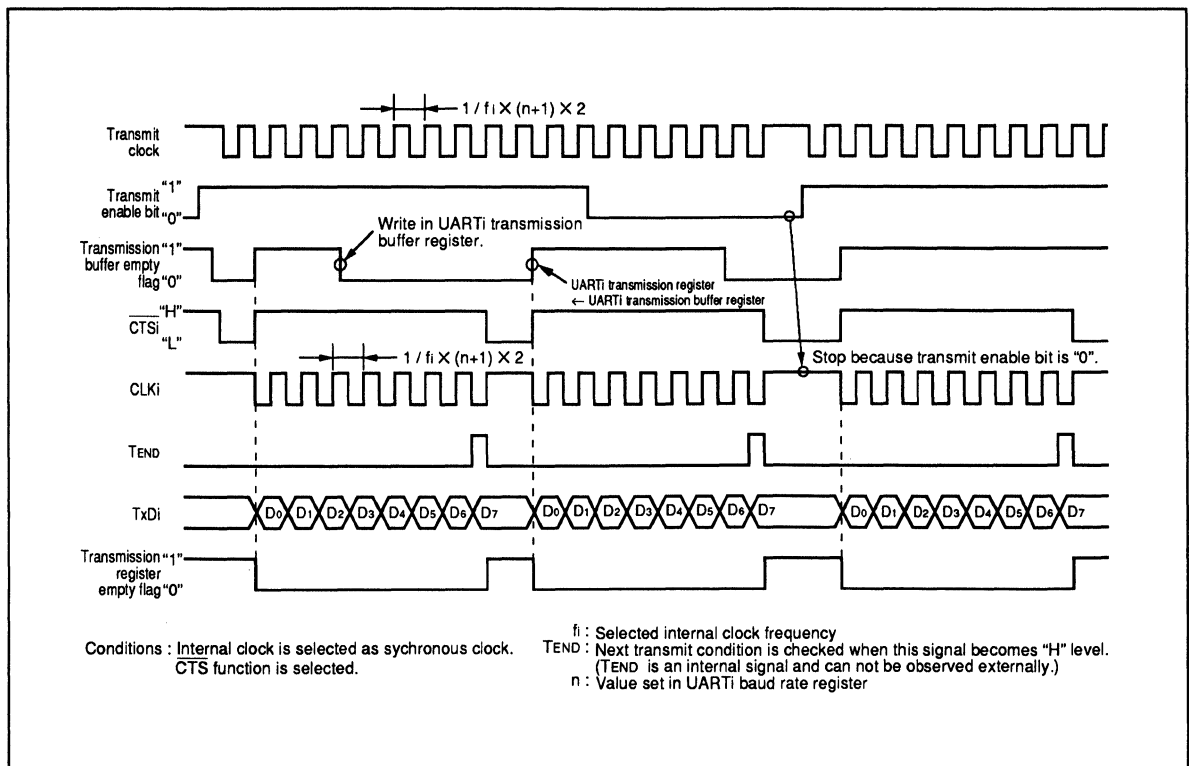


Fig. 2.10.13 Clock synchronous serial I/O timing diagram at transmitting (selecting internal clock)

(3) Serial data receive

The data receive method in clock synchronous serial I/O mode is described below.

[Setting the control registers]

Set each serial I/O control register for receive.

●UARTI transmit/receive mode register

•Serial I/O mode selection bits

Set the bits 2 to 0 to "001".

•Internal/External clock selection bit

Set the bit 3 to "0" when an internal clock is selected, or "1" when an external clock is selected.

•Setting the bit 7 to "0" (disable sleep mode)

●UARTI transmit/receive control register 0

•BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for synchronous clock.

• $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit

Set the bit 2 to "1" when using $\overline{\text{RTS}}$ function, and to "0" when not use.

●UARTI baud rate register

•Dividing ratio

Set the BRG dividing ratio between 00_{16} and FF_{16} when an internal clock is selected for synchronous clock.

●Port P8 direction register

•Port direction selection bits

Set the corresponding bit to "0" to the CLK_i pin in case that an external clock is selected and RxDi pin is selected.

●UARTI receive Interrupt control register

•Interrupt priority level selection bits

When using UARTi receive interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTI transmission buffer register

•Dummy data

Set a dummy data to the low-order byte of UARTi transmission buffer register in order to activate a transmitter. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTI transmit/receive control register 1

•Transmit enable bit

Set the bit 0 to "1" in order to enable transmitting.

•Receive enable bit

Set the bit 2 to "1" in order to enable receiving.

Figure 2.10.14 shows the setting example of clock synchronous serial I/O related registers at receiving.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

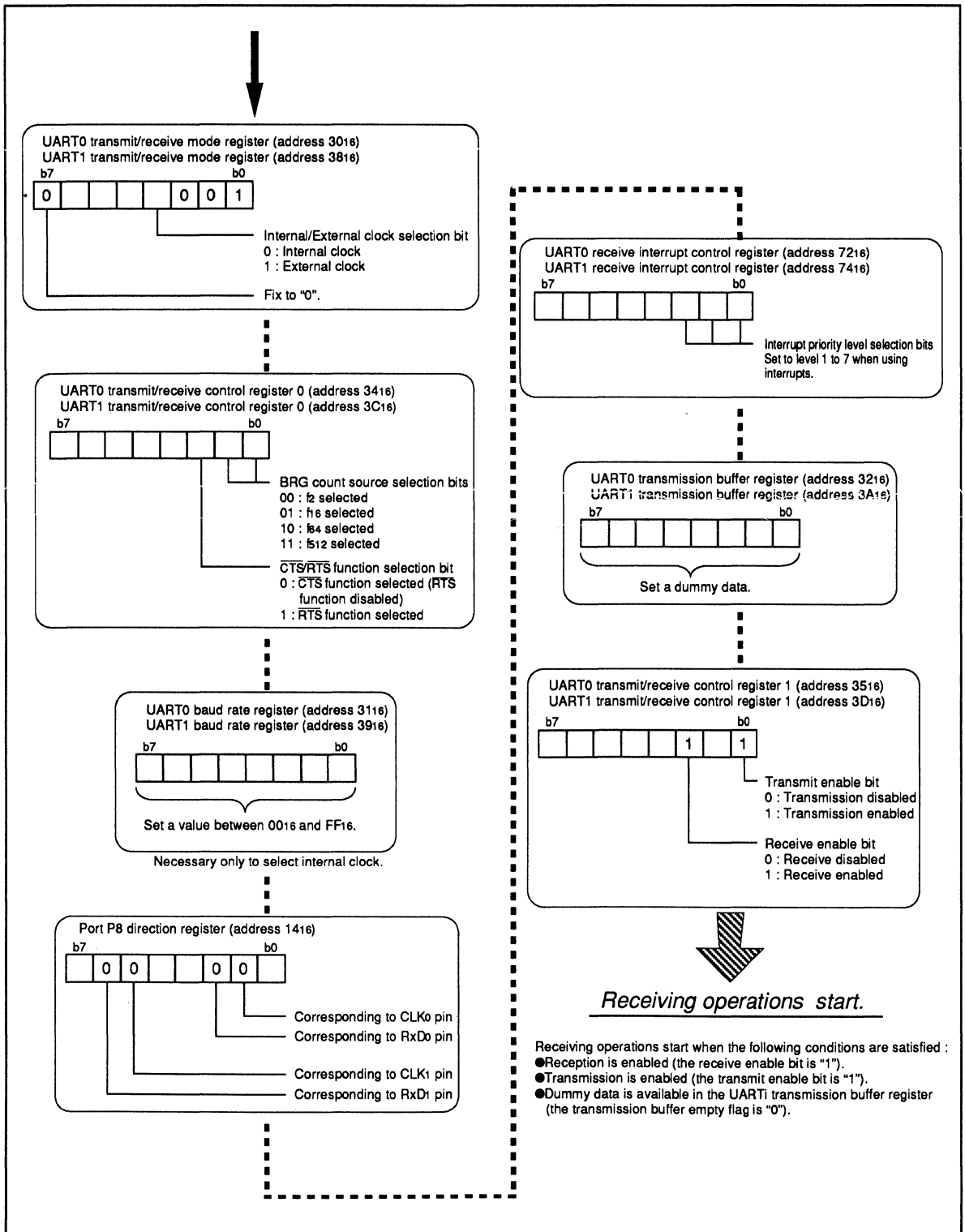


Fig. 2.10.14 Setting example of clock synchronous serial I/O related registers at receiving

[Receive operation]

The reception of serial data starts when the following conditions are satisfied :

- ① Reception is enabled (the receive enable bit is "1").
- ② Transmission is enabled (the transmit enable bit is "1").
- ③ Dummy data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").

Serial data receive is enabled by enabling transmission and setting the receive enable bit to "1". When the receive enable bit is set to "1", the $\overline{\text{RTSi}}$ pin becomes "L" level to indicate externally that the microcomputer is ready to receive serial data (in case that $\overline{\text{RTS}}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{\text{RTSi}}$ output pin to the $\overline{\text{CTS}}$ input pin on the transmit side. Figure 2.10.15 shows the connecting example of clock synchronous serial I/O. When $\overline{\text{RTS}}$ function is selected, if receiving operations start, $\overline{\text{RTSi}}$ pin level becomes "H".

When receiving operations start, the RxDi pin level is used to establish the most significant bit of the UARTi receive register at the rising edge of the shift clock (the clock input to the CLKi pin when an external clock is selected), and the contents of the UARTi receive register are shifted by 1 bit to the right. This operation is repeated each time a rising edge is input. When 1-byte data is accumulated in the UARTi receive register after 8 shift clocks, the contents of UARTi receive register are transferred to the UARTi receive buffer register. At the same time, the receive completion flag is set to "1". In case $\overline{\text{RTS}}$ function is selected, $\overline{\text{RTSi}}$ pin level becomes "L" at the same time. When the receive completion flag is set to "1", UARTi receive interrupt request occurs and the interrupt request bit is set to "1". Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information.

The receive completion flag is cleared to "0" when the UARTi receive buffer register is read.

Figure 2.10.16 shows the clock synchronous serial I/O receive operation and Figure 2.10.17 shows the timing diagram at receiving (external clock is selected, $\overline{\text{RTS}}$ function is selected).

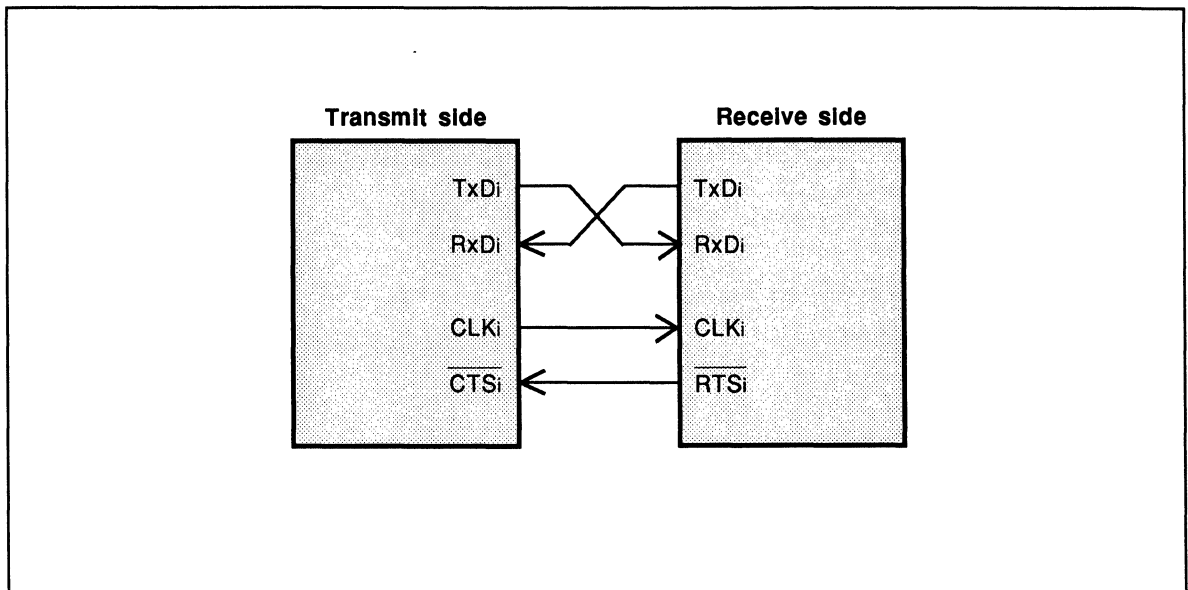


Fig. 2.10.15 Connecting example of clock synchronous serial I/O

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

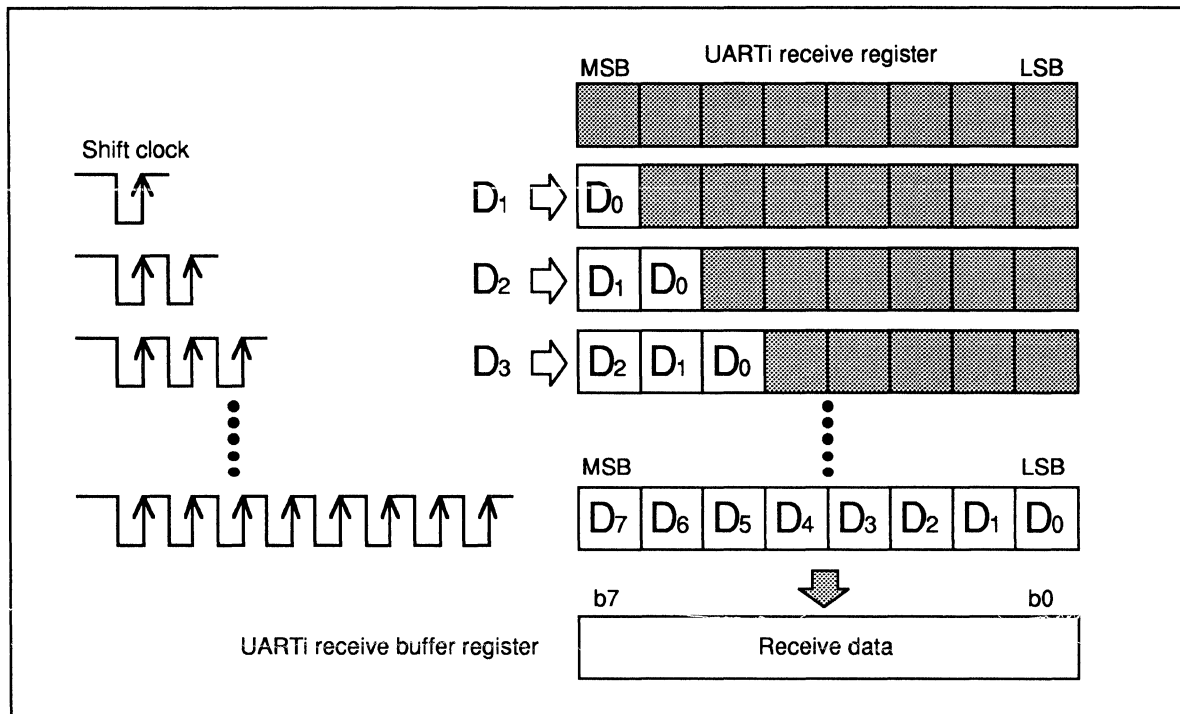


Fig. 2.10.16 Clock synchronous serial I/O receive operation

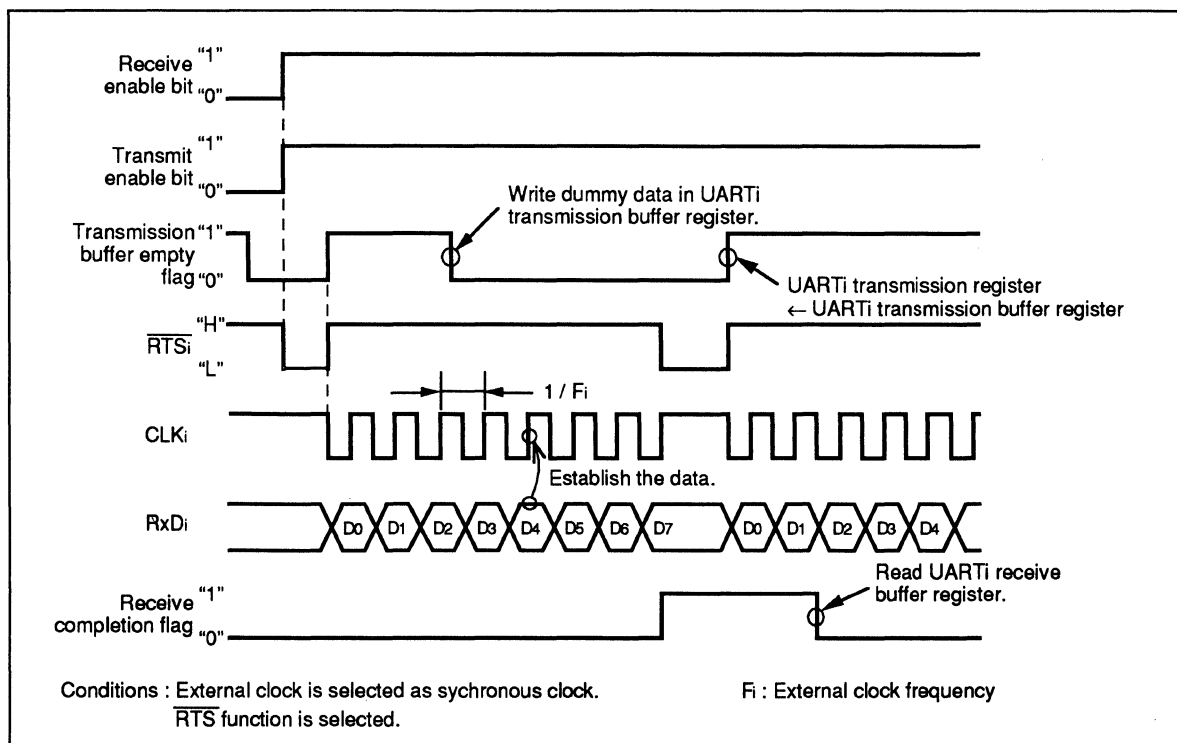


Fig. 2.10.17 Clock synchronous serial I/O timing diagram at receiving (selecting external clock)

[Precautions during clock synchronous serial I/O receive]

1. With clock synchronous serial I/O, shift clocks are generated by operating a transmitter. Therefore, transmission operations (setting for transmission) must be performed even if only receive is necessary. Also note that dummy data is externally output from the TxDi pin (transmit pin) during receive.
2. In case that an internal clock is selected, a shift clock is generated when the transmit enable bit is set to "1" (transmit enabled) and a dummy data is set in the UARTi transmission buffer register.
In case that an external clock is selected, a shift clock is generated when the transmit enable bit is set to "1", a dummy data is set in the UARTi transmission buffer register, and external clock is input to the CLKi pin.
3. When receiving data continuously, an overrun error occurs and bit 4 (overrun error flag) in the UARTi transmit/receive control register 1 is set to "1" if the next receive data becomes available in the UARTi receive register while the receive completion flag is "1" (before reading the contents of the UARTi receive buffer register). In this case, the UARTi receive buffer register contains the next data. Therefore, the transmit and receive programs must make arrangements to re-transmit the previous data when an overrun error occurs.
The interrupt request bit is not set to "1" when an overrun error occurs.
4. When continuously receiving data, a dummy data must be set in the low-order byte of the UARTi transmission buffer register as each receive.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

2.10.4 Clock asynchronous serial I/O (UART)

Table 2.10.5 shows the serial I/O performance overview in UART mode.

Table 2.10.5 UART performance overview

	Parameter	Function
Data format	Start bit	1 bit
	Data bit (character length)	7 bits, 8 bits, or 9 bits
	Parity bit	0 bit or 1 bit (Odd or even can be selected.)
	Stop bit	1 bit or 2 bits
Baud rate	Internal clock	BRG output divided by 16
	External clock	125Kbps maximum ($f(X_{IN})=8\text{MHz}$)
		250Kbps maximum ($f(X_{IN})=16\text{MHz}$)
	312.5Kbps maximum ($f(X_{IN})=25\text{MHz}$)	
Error detection		4 types (overrun, parity, framing, error sum) (Error sum can be used to check existence of error.)

In UART mode, the baud rate*¹ and the transfer data format must be set beforehand. The setting of the baud rate and the transfer format are described below.

Baud rate*¹ : Frequency of the clock used for transmission and receive.

(1) Baud rate

The serial data transfer rate is determined by the baud rate. The baud rate is set by the UARTi baud rate register. The BRG is a frequency divider that has an 8-bit structure. The BRG input clock can be either an internal clock or an external clock input to the CLKi pin with the internal/external clock selection bit.

When an internal clock is selected, 1/2, 1/16, 1/64, or 1/512 of the $f(X_{IN})$ is selected with the BRG count source selection bits. When an external clock is selected, the clock input to the CLKi pin is input to the BRG.

The clock input to BRG is divided by (n+1) and then by 16 to obtain the baud rate.

Table 2.10.6 shows the baud rate selection table.

If the required baud rate is B (bps), use the following equation to determine the value "n" set in the UARTi baud rate register.

$$"n" = F_i / (16 \times B) - 1$$

"B"=the required baud rate,
"F_i"=the clock frequency input to the BRG

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

Table 2.10.6 Baud rate selection table (1)

Baud rate (bps)		f(X _{IN})=8MHz		f(X _{IN})=16MHz	
Rated	Actual	f _i	n*	f _i	n*
75	75.12	f ₅₁₂	12 (0C ₁₆)	f ₅₁₂	25 (19 ₁₆)
110	110.04	f ₆₄	70 (46 ₁₆)	f ₆₄	141 (8D ₁₆)
134.5	134.70	f ₆₄	57 (39 ₁₆)	f ₆₄	115 (73 ₁₆)
150	150.24	f ₆₄	51 (33 ₁₆)	f ₆₄	103 (67 ₁₆)
300	300.48	f ₆₄	25 (19 ₁₆)	f ₆₄	51 (33 ₁₆)
600	600.96	f ₆₄	12 (0C ₁₆)	f ₆₄	25 (19 ₁₆)
1200	1201.92	f ₁₆	25 (19 ₁₆)	f ₁₆	51 (33 ₁₆)
2400	2403.85	f ₁₆	12 (0C ₁₆)	f ₁₆	25 (19 ₁₆)
4800	4807.69	f ₂	51 (33 ₁₆)	f ₂	103 (67 ₁₆)
9600	9615.39	f ₂	25 (19 ₁₆)	f ₂	51 (33 ₁₆)
19200	19230.77	f ₂	12 (0C ₁₆)	f ₂	25 (19 ₁₆)
31250	31250.00	f ₂	7 (07 ₁₆)	f ₂	15 (0F ₁₆)
62500	62500.00	f ₂	3 (03 ₁₆)	f ₂	7 (07 ₁₆)
125000	125000.00	f ₂	1 (01 ₁₆)	f ₂	3 (03 ₁₆)
250000	250000.00	f ₂	0 (00 ₁₆)	f ₂	1 (01 ₁₆)
500000	500000.00	f ₂	—	f ₂	0 (00 ₁₆)

n* : UARTi baud rate register value

Table 2.10.6 Baud rate selection table (2)

Baud rate (bps)	f(X _{IN})=20MHz			f(X _{IN})=25MHz		
	f _i	n*	Actual (bps)	f _i	n*	Actual (bps)
150	f ₆₄	129 (81 ₁₆)	150.24	f ₆₄	162 (A2 ₁₆)	149.78
300	f ₆₄	64 (40 ₁₆)	300.48	f ₆₄	80 (55 ₁₆)	301.41
600	f ₁₆	129 (81 ₁₆)	600.96	f ₁₆	162 (A2 ₁₆)	599.12
1200	f ₁₆	64 (40 ₁₆)	1201.92	f ₁₆	80 (55 ₁₆)	1205.63
2400	f ₁₆	32 (20 ₁₆)	2367.42	f ₁₆	40 (28 ₁₆)	2381.86
4800	f ₂	129 (81 ₁₆)	4807.69	f ₂	162 (A2 ₁₆)	4792.94
9600	f ₂	64 (40 ₁₆)	9615.38	f ₂	80 (55 ₁₆)	9645.06
19200	f ₂	32 (20 ₁₆)	18939.39	f ₂	40 (28 ₁₆)	19054.88
31250	f ₂	19 (13 ₁₆)	31250.00	f ₂	24 (18 ₁₆)	31250.00

n* : UARTi baud rate register value

FUNCTIONAL DESCRIPTION

(2) Transfer format

The format of the transfer data is set with the UARTi transmit/receive mode register. Data can be transferred in the following format shown by Figure 2.10.18.

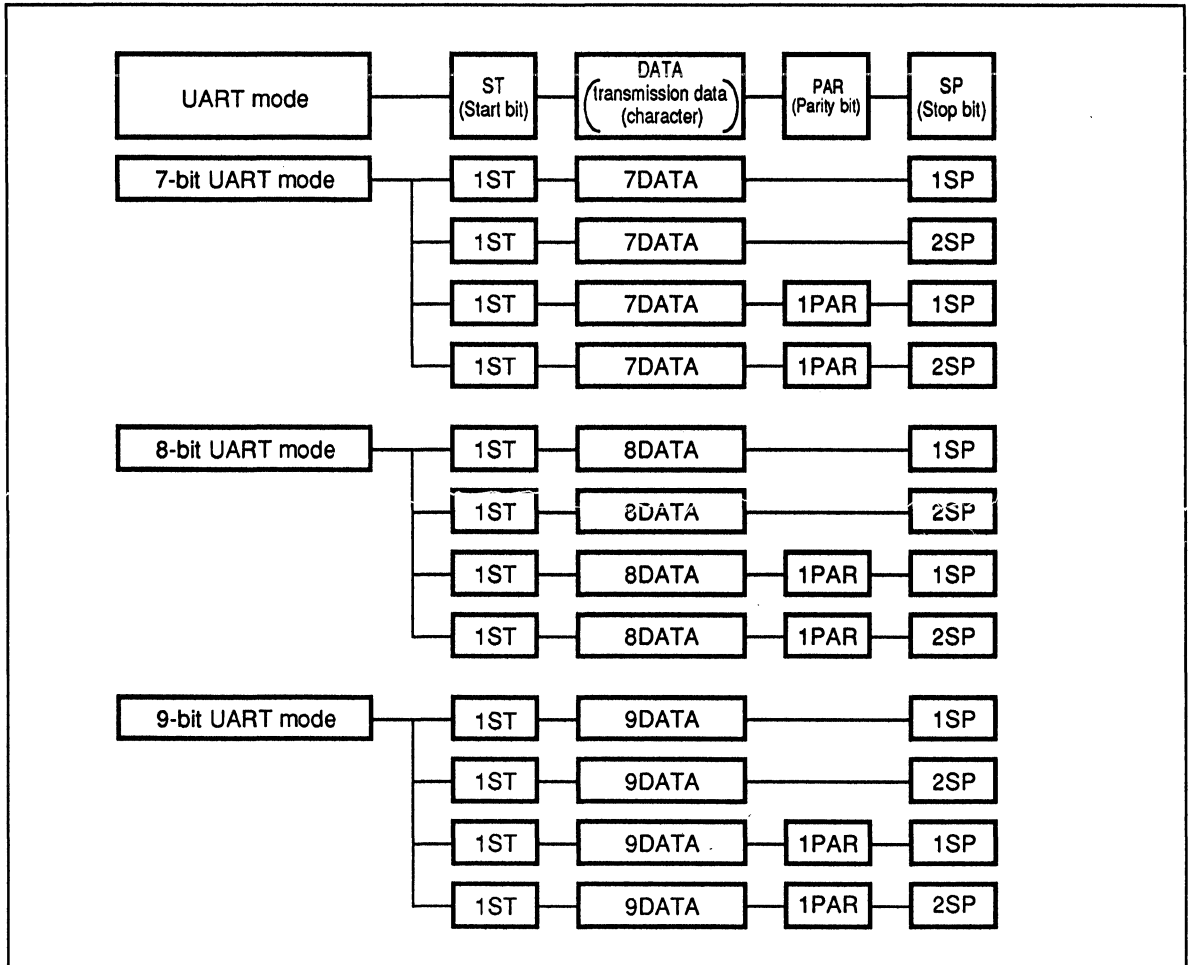


Fig. 2.10.18 Data format for transfer

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

Figure 2.10.19 shows the data format for example and Table 2.10.7 shows the transmission data in UART mode.

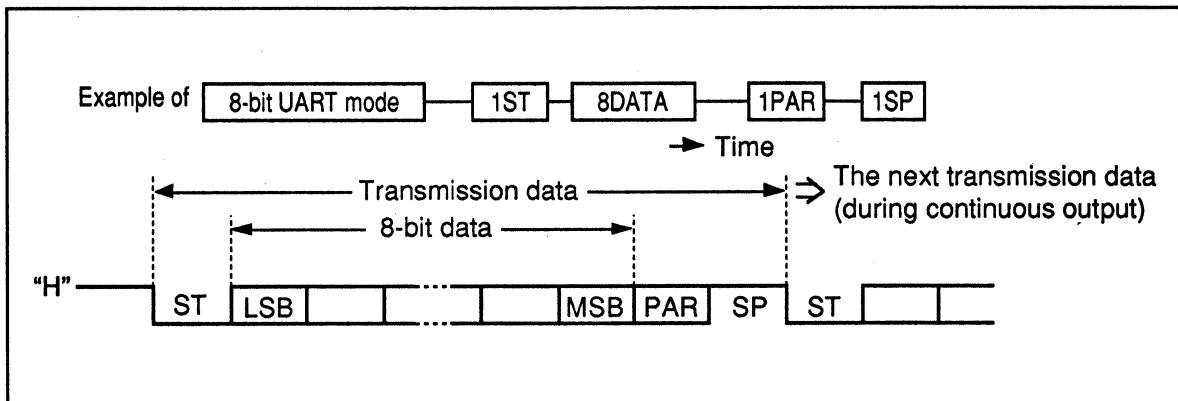


Fig. 2.10.19 Data format example

Table 2.10.7 Transmission data in UART mode

Item	Function
ST (Start bit)	This bit indicates the start of data transmission. A 1-bit "L" signal is appended in front of the transmission data.
DATA (Character)	This is the transmission data written in the UARTi transmission buffer register.
SP (Stop bit)	This bit appends after the data (or after the parity bit if it is included) to indicate the end of transmission. A 1 or 2-bit "H" signal is output as a stop bit.
PAR (Parity bit)	This bit appends to the end of data to improve the reliability of data. This bit is appended by the contents of the odd/even parity selection bit so that the number of 1s in the data including the parity bit is always even or odd.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

(3) Serial data transmission

The data transmission method in UART mode is described below.

[Setting the control registers]

Set each serial I/O control register for transmission.

●UARTI transmit/receive mode register

- Serial I/O mode selection bits
Select the data length with the bits 0 to 2.

Table 2.10.8 Setting of UART mode

b2	b1	b0	Serial I/O mode selection bits
1	0	0	7-bit UART mode
1	0	1	8-bit UART mode
1	1	0	9-bit UART mode

•Transfer format

Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.

•Internal/External clock selection bit

Set the bit 3 to "0" when an internal clock is selected, or "1" when an external clock is selected as the BRG count source.

•Sleep function selection bit

Set the bit 7 to "1" when enable the sleep function, and to "0" when disable it.
(Refer to "2.10.5 Sleep mode" for details of sleep mode.)

●UARTI transmit/receive control register 0

- BRG count source selection bits
Select the BRG count source with bits 0 and 1 when an internal clock is selected for BRG input clock.

• $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit

Set the bit 2 to "0" when using $\overline{\text{CTS}}$ function, and to "1" when not use.

●UARTI baud rate register

- Dividing ratio
Set the baud rate register value between 00_{16} and FF_{16} to determine the baud rate.

●Port P8 direction register

- Port direction selection bits
Set the corresponding bit to "0" when the $\overline{\text{CTS}}$ function is selected and an external clock is selected.

●UARTI transmission interrupt control register

- Interrupt priority level selection bits
When using UARTi transmission interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTI transmission buffer register

- Transmission data
Set a transmission data to the UARTi transmission buffer register. The transmission buffer empty flag of UARTi transmit/receive control register 1 is cleared to "0" at the same time.

●UARTI transmit/receive control register 1

- Transmit enable bit
Set the bit 0 to "1" to enable transmitting.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

Figure 2.10.20 shows the setting example of UART related registers at transmitting.

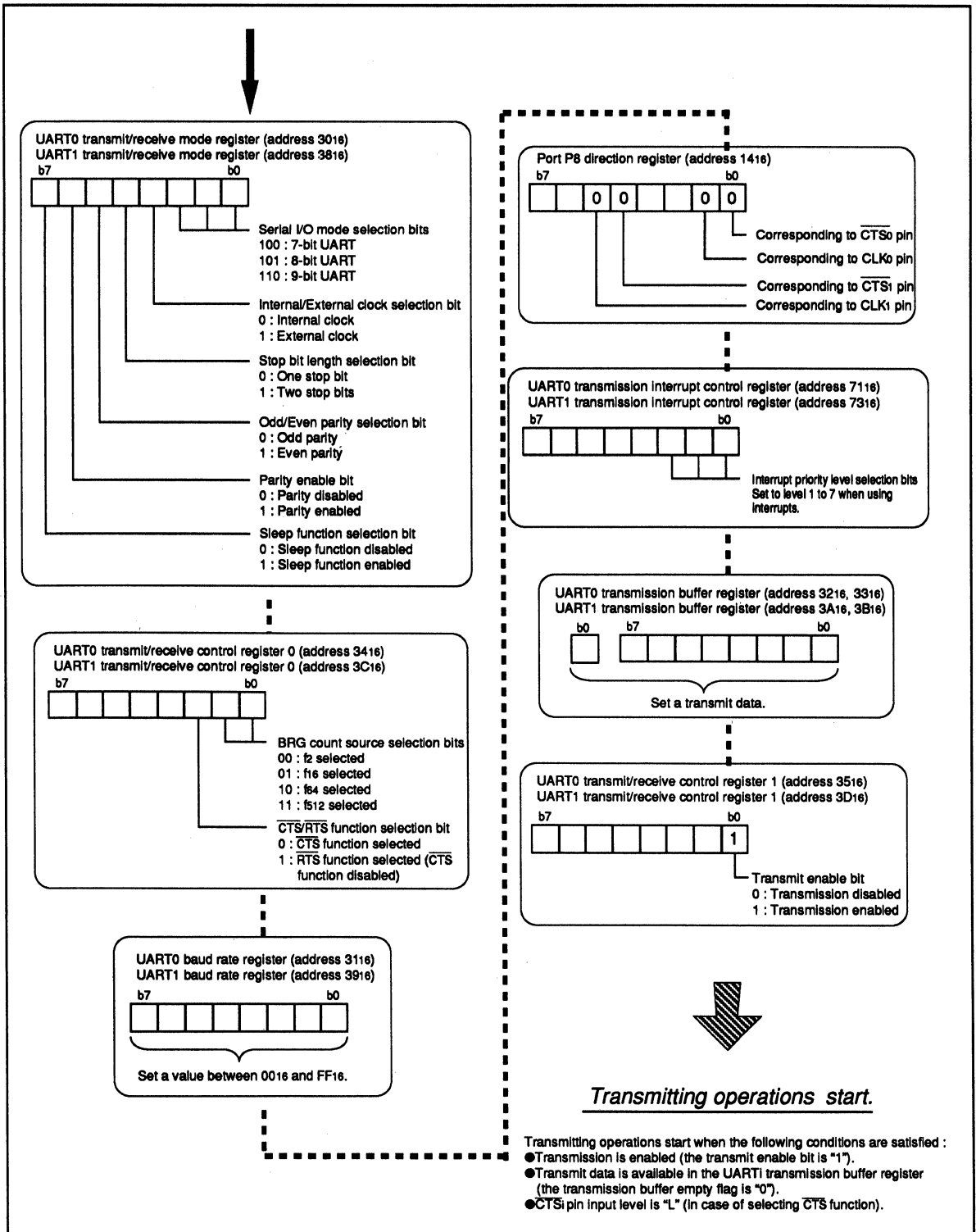


Fig. 2.10.20 Setting example of UART related registers at transmitting

[Transmit operation]

The only difference between 7-bit UART, 8-bit UART, and 9-bit UART is the length of the transferred data. The low-order byte of the UARTi transmission buffer register is used for 7-bit and 8-bit UART. The low-order byte and bit 0 of the high-order byte is used for 9-bit UART.

The transmission of serial data starts when the following conditions are satisfied :

- ① Transmission is enabled (the transmit enable bit is "1").
- ② Transmit data is available in the UARTi transmission buffer register (the transmission buffer empty flag is "0").
- ③ $\overline{\text{CTS}}_i$ pin input level is "L".
(Note: This condition is ignored if the $\overline{\text{CTS}}$ function is not selected.)

When the above three (① to ③) conditions are satisfied (two (① and ②) if $\overline{\text{CTS}}$ function is not selected), the following operations are performed automatically at the same time :

- Transfer the contents of UARTi transmission buffer register to the UARTi transmission register.
- Set the transmission buffer empty flag to "1".
- Clear the transmission register empty flag to "0".
- UARTi transmission interrupt request occurs and set the interrupt request bit to "1".

Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information.

Data transmission starts from the TxDi pin when the data is transferred to the UARTi transmission register. When transmission starts, data is output from the TxDi pin in the format specified by the UARTi transmit/receive mode register. The data is output bit by bit in the order;

ST→DATA(LSB)→...→DATA(MSB)→PAR→SP.

After the stop bit has been output, the transmission register empty flag is set to "1" to indicate that the transmission has completed. If the next data is available when transmission completes, a start bit is generated following the stop bit and the next data is transferred. In order to continuously transfer data, the next transmission data must be set in the UARTi transmission buffer register during transmitting operations (when the transmission register empty flag is "0"). If the transmit conditions for the next data is not satisfied, "H" level is output from the TxDi pin.

Figure 2.10.21 shows the timing diagram of 8-bit UART at transmitting (with parity bit, 1 stop bit and $\overline{\text{CTS}}$ function). Figure 2.10.22 shows the timing diagram of 9-bit UART at transmitting (with 2 stop bits, no parity bit and no $\overline{\text{CTS}}$ function).

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

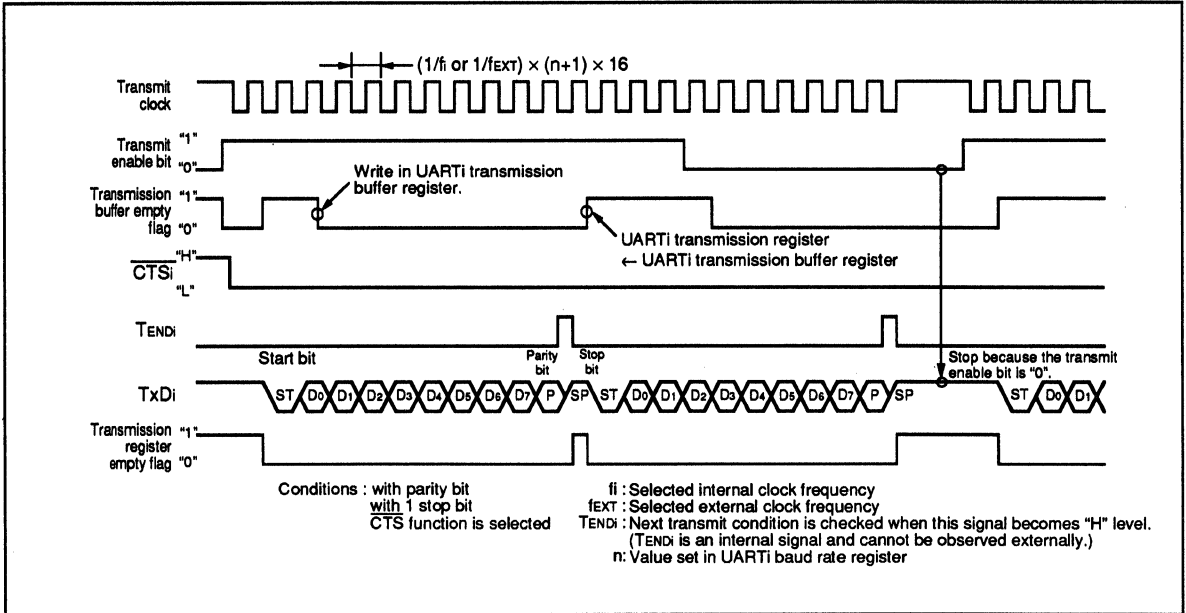


Fig. 2.10.21 8-bit UART timing diagram at transmitting (with parity and 1 stop bit)

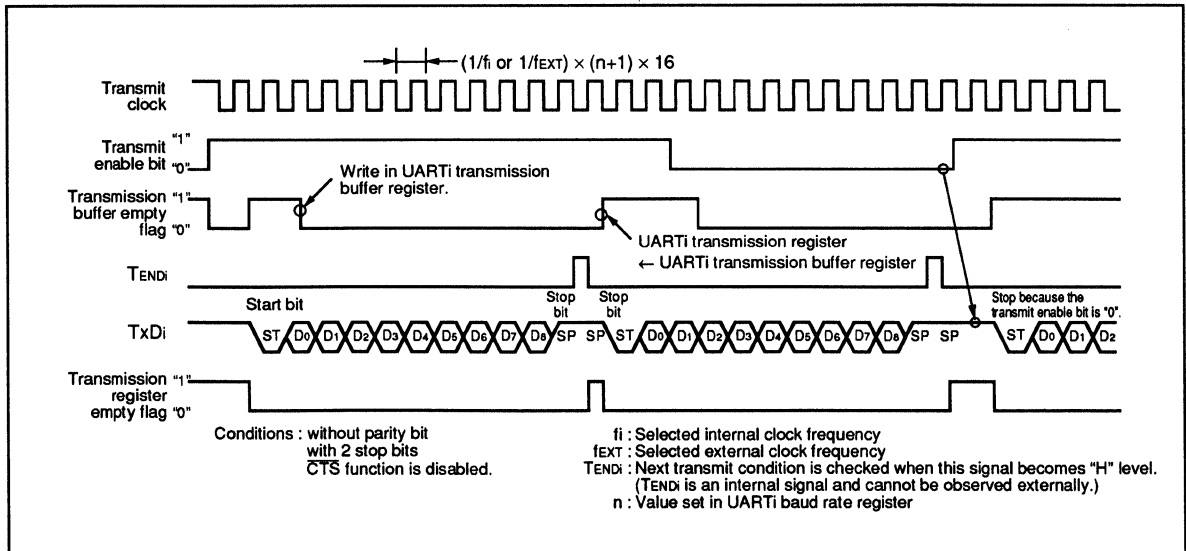


Fig. 2.10.22 9-bit UART timing diagram at transmitting (no parity and 2 stop bits)

FUNCTIONAL DESCRIPTION

(4) Serial data receive

The data receive method in UART mode is described below.

[Setting the control registers]

Set each serial I/O control register for receive.

●UART transmit/receive mode register

Match the format with the transmitting side.

•Serial I/O mode selection bits

Select the data length with the bits 0 to 2.

Table 2.10.9 Setting of UART mode

b2	b1	b0	Serial I/O selection bits
1	0	0	7-bit UART mode
1	0	1	8-bit UART mode
1	1	0	9-bit UART mode

•Transfer format

Select stop bit length, parity enable/disable, and odd/even parity if parity is enabled.

•Internal/External clock selection bit

Set the bit 3 to "0" when an internal clock is selected, or "1" when an external clock is selected as the BRG count source.

•Sleep function selection bit

Set the bit 7 to "1" when enable the sleep function, and to "0" when disable it.
(Refer to "2.10.5 Sleep mode" for details of sleep mode.)

●UARTI transmit/receive control register 0

•BRG count source selection bits

Select the BRG count source with bits 0 and 1 when an internal clock is selected for BRG input clock.

• $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function selection bit

Set the bit 2 to "1" when using $\overline{\text{RTS}}$ function, and to "0" when not use.

●UARTI baud rate register

•Dividing ratio

Set the baud rate register value between 00₁₆ and FF₁₆ to determine the baud rate.

●Port P8 direction register

•Port direction selection bits

Set the corresponding bit to "0" to the CLK_i pin when an external clock and the RxD_i pin is selected.

●UARTI receive interrupt control register

•Interrupt priority level selection bits

When using UART_i receive interrupt, set the priority level to the level 1 to 7. When not use, set to the level 0.

●UARTI transmit/receive control register 1

•Receive enable bit

Set the bit 2 to "1" to enable receiving.

Figure 2.10.23 shows the setting example of UART related registers at receiving.

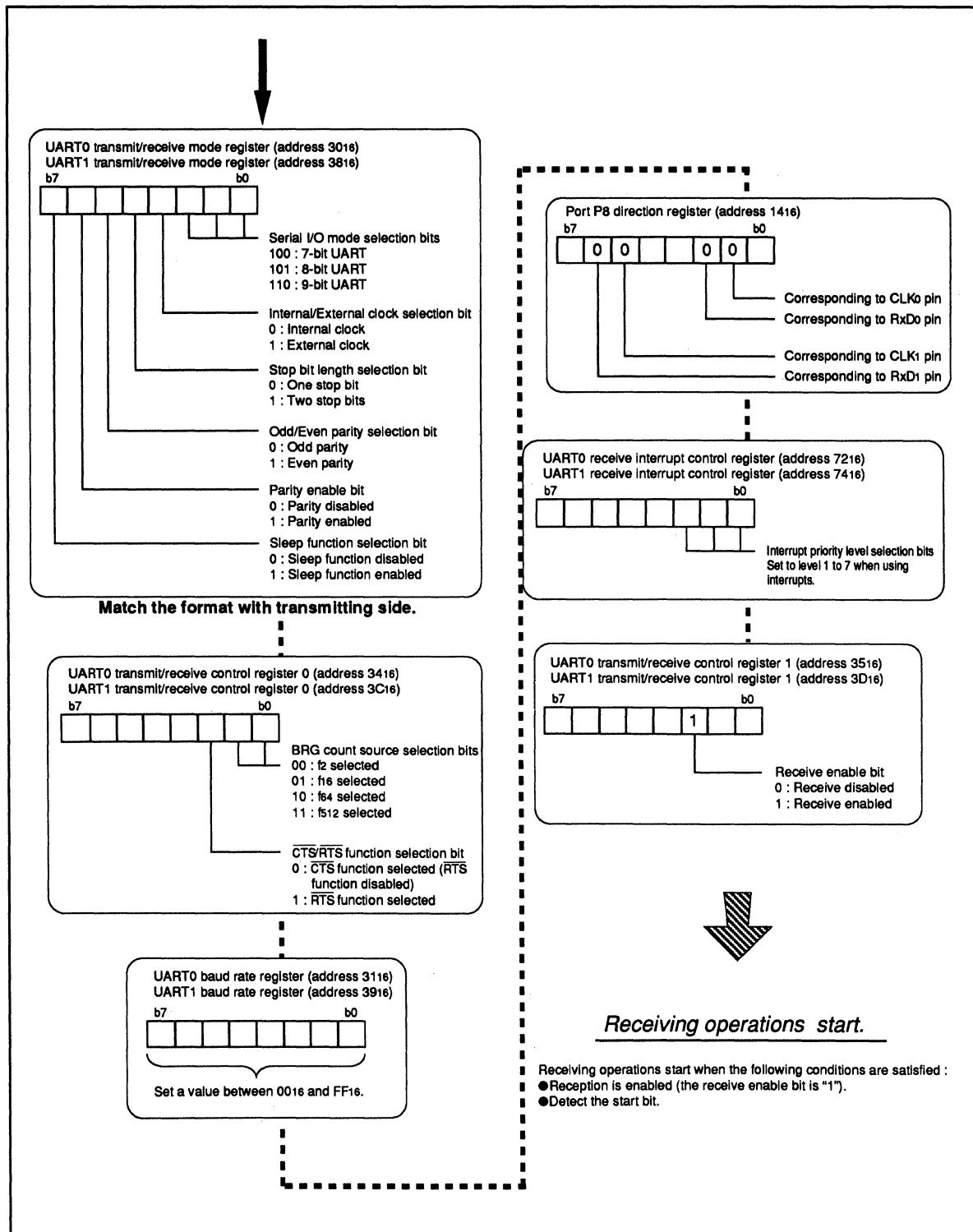


Fig. 2.10.23 Setting example of UART related registers at receiving

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

[Receive operation]

The reception of serial data starts when the following conditions are satisfied :

- ① Reception is enabled (the receive enable bit is "1").
- ② Detect the start bit.

Serial data receive is enabled by setting the receive enable bit to "1". When the receive enable bit is set to "1", the $\overline{\text{RTSi}}$ pin becomes "L" level to indicate externally that the microcomputer is ready to receive serial data (in case that $\overline{\text{RTS}}$ function is selected). The transmit and receive timing can be synchronized by connecting the $\overline{\text{RTSi}}$ output pin to the $\overline{\text{CTS}}$ input pin on the transmit side. Figure 2.10.24 shows the connecting example of UART.

When the RxDi pin detects a start bit, a receive clock is generated and data receive starts. At the same time, the $\overline{\text{RTSi}}$ pin returns to "H" level if $\overline{\text{RTS}}$ function is selected. The RxDi pin level is used to establish the most significant bit of the UARTi receive register at the rising edge of the receive clock and the contents of UARTi receive register are shifted by 1 bit to the right. This operation is repeated to receive the entire data from ST to SP. Then the contents of UARTi receive register are transferred to the UARTi receive buffer register. At the same time, the receive completion flag is set to "1". When the receive completion flag is set to "1" by detecting SP, the UARTi receive interrupt request occurs and the interrupt request bit is set to "1". In case $\overline{\text{RTS}}$ function is selected, $\overline{\text{RTSi}}$ pin level becomes "L" at the same time. Interrupts must be enabled before they can be used. Refer to section "2.6 Interrupts" for more information. The receive completion flag is cleared to "0" when the UARTi receive buffer register is read.

Figure 2.10.25 shows the timing diagram of 8-bit UART at receive (no parity bit, with 1 stop bit and $\overline{\text{RTS}}$ function).

When receiving data continuously, an overrun error occurs and the bit 4 (overrun error flag) in the UARTi transmit/receive control register 1 is set to "1" if the next receive data becomes available in the UARTi receive register while the receive completion flag is "1" (before reading the contents of the UARTi receive buffer register). In this case, the next data is written in the UARTi receive buffer register. Therefore, if an overrun error occurs, the transmit and receive programs must make arrangements to re-transmit the previous data. The interrupt request bit is not set to "1" when an overrun error occurs.

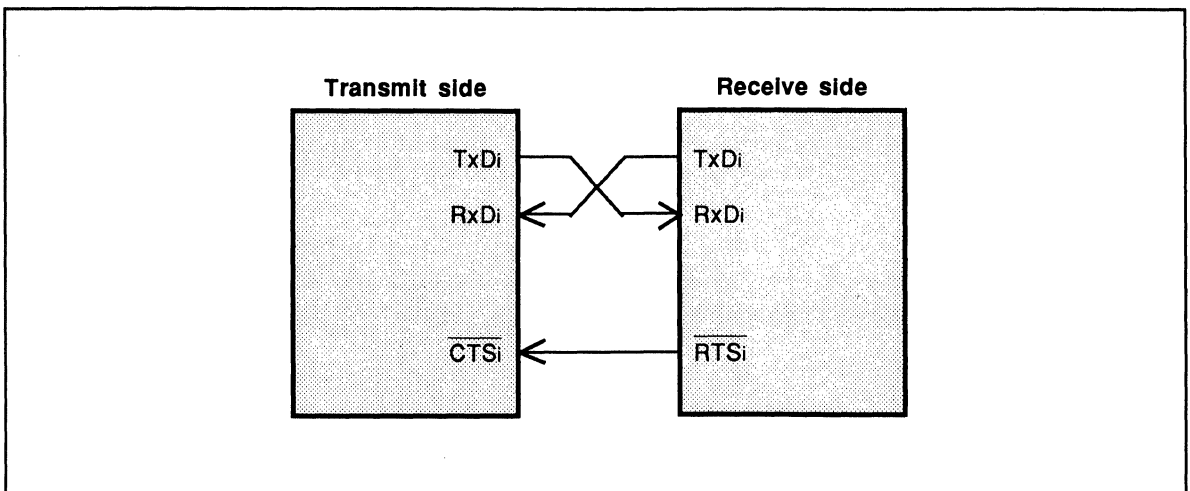


Fig. 2.10.24 Connecting example of UART

FUNCTIONAL DESCRIPTION

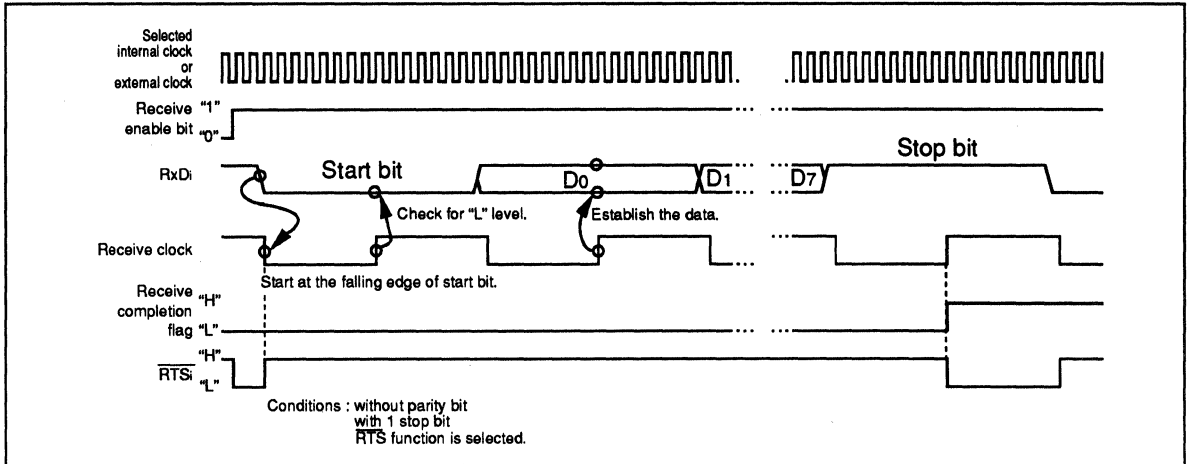


Fig. 2.10.25 8-bit UART timing diagram at receiving (no parity and 1 stop bit)

[Error flag]

During UART mode operation, transfer data errors can be detected using four error flags. These errors are detected when transferring data from the UARTi receive register to the UARTi receive buffer register. The error flags set to "1" are cleared to "0" when the low-order byte of the UARTi receive buffer register is read or when the receive enable bit is set to "0".

● Overrun error

An overrun error occurs and the overrun error flag is set to "1" when the next receive data becomes available in the UARTi receive register and transferred to the UARTi receive buffer register while the receive completion flag is "1" (data exists in UARTi receive buffer register), that is to say, the next receive data becomes available before the contents of the UARTi receive buffer register are read.

● Framing error

A framing error occurs and the framing error flag is set to "1" when there is insufficient number of stop bits.

● Parity error

A parity error occurs and the parity error flag is set to "1" when parity checking is enabled and the number of 1s in the data including the parity bit conflicts with the parity specified by the odd/even parity selection bit.

● Sum error

The error sum flag is set to "1" when either an overrun error, a framing error, or a parity error occurs. The existence of errors can be determined by checking only the error sum flag.

FUNCTIONAL DESCRIPTION

2.10 Serial I/O

2.10.5 Sleep mode

Sleep mode is used for communication between certain microcomputers when multiple microcomputers are connected through serial I/O.

Sleep mode is entered by setting the UARTi transmit/receive mode register bit 7 to "1". In sleep mode, the contents of UARTi receive register are not transferred to the UARTi receive buffer register when the most significant bit (MSB: bit 8 if 9-bit UART mode, bit 7 if 8-bit UART mode, and bit 6 if 7-bit UART mode) of the received data is "0". In this case, the receive completion flag and the error flags remain unchanged and no receive interrupt request occurs. Normal receive operation is performed only when the most significant bit of the received data is "1".

The following is a description of sleep mode usage in 8-bit UART mode. The main microcomputer first transmits a data with bit 7 set to "1" and the remaining bits 0 to 6 forming the address of the destination microcomputer. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data and sets the sleep function selection bit to "0" if the address matches its own address and to "1" if otherwise. Next the main microcomputer starts transmitting data with bit 7 set to "0". Then only the microcomputer with the sleep function selection bit set to "0" will receive this data. This enables communication between the main microcomputer and a specific subordinate microcomputer. Figure 2.10.26 shows the sleep mode.

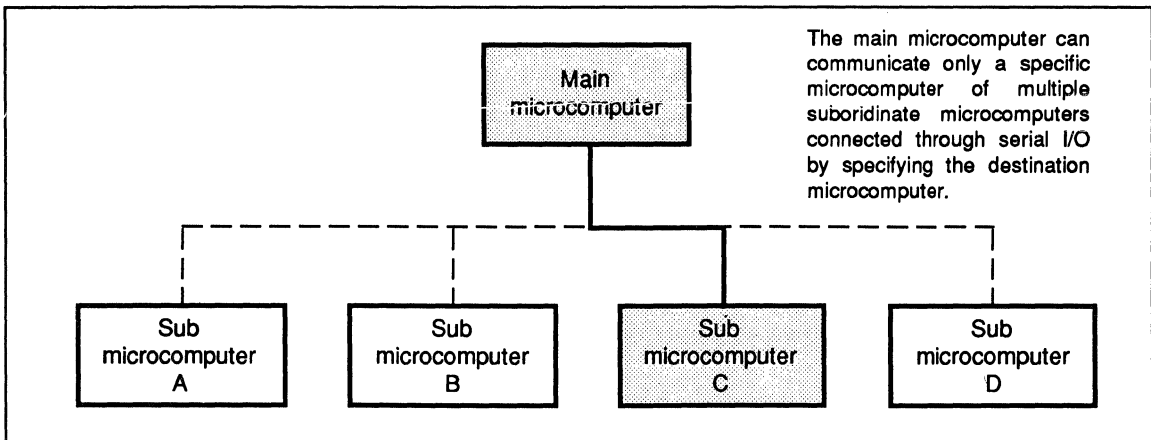


Fig. 2.10.26 Sleep mode

FUNCTIONAL DESCRIPTION

2.11 A-D converter

2.11 A-D converter

The M37732 group has a built-in 8-bit A-D converter that performs successive approximation to convert analog values input from pins AN₀–AN₇ to digital values.

The A-D converter provides four selectable conversion modes.

2.11.1 A-D converter overview

Table 2.11.1 shows the performance overview of the A-D converter.

Table 2.11.1 A-D converter performance overview

Parameter	Description
Analog input pin	8 pins (AN ₀ –AN ₇)
A-D conversion mode	One-shot mode
	Repeat mode
	Single sweep mode
	Repeat sweep mode
A-D conversion method	Successive approximation
Resolution	8 bits
Absolute accuracy	±3LSB
Conversion speed	57 ϕ_{AD} cycles ϕ_{AD} : A-D converter operating clock (for 1 analog input pin)

The A-D converter provides the following four conversion modes.

- **One-shot mode**..... The input voltage to one selected analog input pin is converted. After conversion, the result is stored in the corresponding A-D register and an A-D conversion interrupt request occurs.
- **Repeat mode**..... The input voltage to one selected analog input pin is repeatedly converted. The result is stored in the corresponding A-D register each time the conversion completes, but no A-D conversion interrupt request occurs.
- **Single sweep mode**..... The analog input pins to be converted can be selected with the A-D sweep pin selection register. The selected pins are converted in the order AN₀, AN₁,... and an A-D conversion interrupt request occurs when the last pin is converted. The result is stored in the corresponding A-D register when each pin is converted.
- **Repeat sweep mode**..... This is similar to single sweep mode except that conversion is repeated in order from the AN₀ pin without an A-D conversion interrupt request after converting the last pin.

FUNCTIONAL DESCRIPTION

2.11 A-D converter

2.11.2 Block description

Figure 2.11.1 shows the block diagram of the A-D converter. The A-D converter related registers are described below.

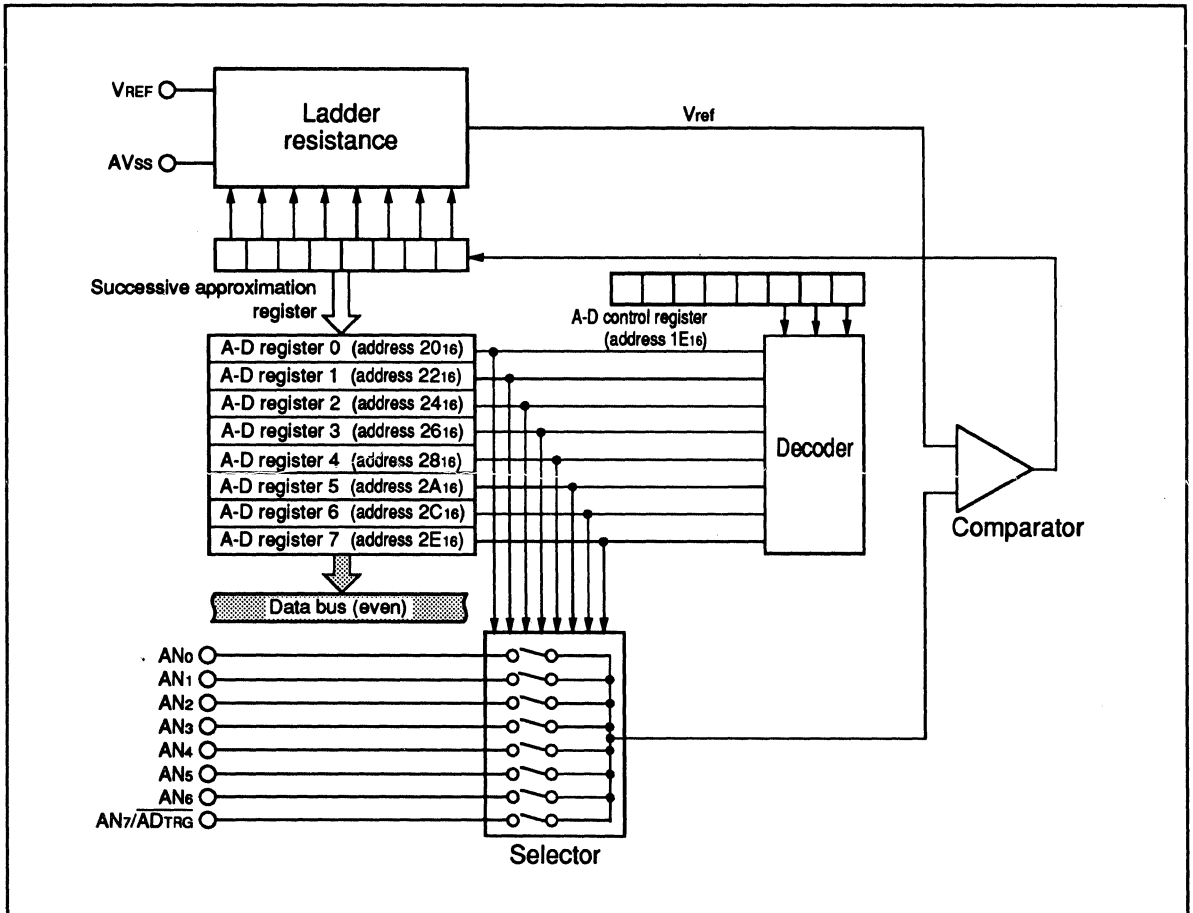


Fig. 2.11.1 A-D converter block diagram

FUNCTIONAL DESCRIPTION

2.11 A-D converter

(1) A-D control register

The A-D control register (address 1E₁₈) consists of bits that control the A-D converter. Figure 2.11.2 shows the structure of the A-D control register followed by description of each bit.

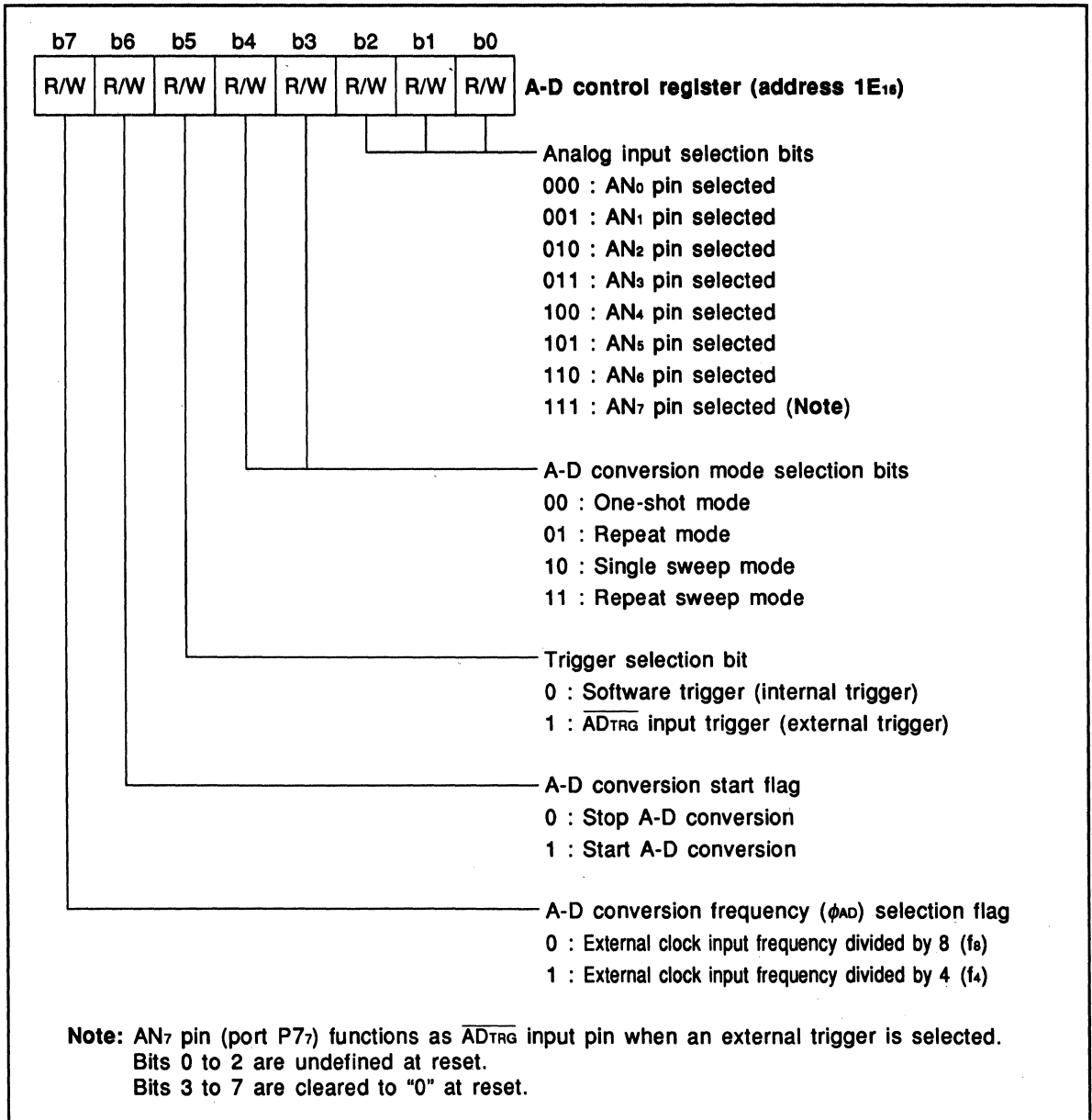


Fig. 2.11.2 A-D control register structure

FUNCTIONAL DESCRIPTION

2.11 A-D converter

●Analog Input selection bits (bits 0 to 2)

The analog input selection bits are used to select one analog input pin in one-shot mode and repeat mode. Table 2.11.2 shows the relationship between the analog input selection bits and analog input pins. The selected analog input pin remains unchanged when the mode is switched between one-shot mode and repeat mode. These bits are ignored in single sweep mode and repeat sweep mode. If an external trigger is selected with the trigger selection bit (described later), port P7 functions as AD_{TRG} pin and cannot be used as AN₇ pin. Pins not selected as analog input pin can be used as normal I/O ports.

Table 2.11.2 Relationship between analog input selection bits and analog input pins

b2	b1	b0	Analog input pin
0	0	0	AN ₀ pin selected
0	0	1	AN ₁ pin selected
0	1	0	AN ₂ pin selected
0	1	1	AN ₃ pin selected
1	0	0	AN ₄ pin selected
1	0	1	AN ₅ pin selected
1	1	0	AN ₆ pin selected
1	1	1	AN ₇ pin selected

●A-D conversion mode selection bits (bits 3, 4)

The A-D conversion mode selection bits are used to select one of the four available conversion modes. Table 2.11.3 shows the relationship between the A-D conversion mode selection bits and the conversion modes.

Table 2.11.3 Relationship between A-D conversion mode selection bits and conversion modes

b4	b3	A-D conversion mode
0	0	One-shot mode
0	1	Repeat mode
1	0	Single sweep mode
1	1	Repeat sweep mode

●Trigger selection bit (bit 5)

The trigger selection bit is used to select the trigger occurrence factor which start an A-D conversion operation. An internal trigger or an external trigger is available for the trigger. An internal trigger (software trigger) is selected when this bit is "0" and an external trigger (input signal to AD_{TRG} pin) is selected when this bit is "1".

<Internal trigger>

A trigger is occurred and A-D conversion starts when the A-D conversion start flag (described later) is set to "1".

<External trigger>

A trigger is occurred when the level of the input signal to the AD_{TRG} pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is "1". When an external trigger is selected, a retrigger is available during A-D conversion. In this cases, the conversion is restarted from the beginning.

The AD_{TRG} pin is shared with the AN₇ pin (port P7₇). Therefore, the AN₇ pin cannot be used as the analog input pin when an external trigger is selected.

When an external trigger is selected in each A-D conversion mode, the port P7 direction register bit 7 must be set to "0" (input mode).

FUNCTIONAL DESCRIPTION

2.11 A-D converter

●A-D conversion start flag (bit 6)

The A-D conversion start flag is used to start or stop A-D conversion.

<Internal trigger>

An internal trigger is occurred and A-D conversion starts when the A-D conversion start flag is set to "1". A-D conversion stops when it is cleared to "0". This bit is automatically cleared to "0" after A-D conversion in one-shot mode and single sweep mode. It is not cleared in other modes and conversion continues until it is cleared to "0".

<External trigger>

The A-D conversion start flag must be set to "1" before the falling edge is input to the $\overline{\text{ADTRG}}$ pin. If an external trigger is selected, this flag is retained "1" after conversion.

●A-D conversion frequency selection flag (bit 7)

This flag is used to select the A-D converter operating clock (ϕ_{AD}). When this flag is "0", the external clock input frequency $f(X_{\text{IN}})$ divided by 8 is selected. When this flag is "1", the external clock input frequency $f(X_{\text{IN}})$ divided by 4 is selected.

In one-shot mode and repeat mode, A-D conversion completes after $57 \times \phi_{\text{AD}}$ cycles from the beginning of A-D conversion. In single sweep and repeat sweep mode, one A-D conversion completes after $57 \times$ number of selected pins $\times \phi_{\text{AD}}$ cycles from the beginning of A-D conversion.

The A-D converter operating clock ϕ_{AD} during A-D conversion must be no less than 250kHz because the comparator in the A-D conversion circuit consists of capacity coupling amplifiers.

Table 2.11.4 shows the relationship between the A-D conversion frequency selection flag and the A-D converter operating clock and conversion time.

Table 2.11.4 Relationship between A-D conversion frequency selection flag and A-D converter operating clock and conversion time

A-D conversion frequency selection flag	"0"	"1"
A-D converter operating clock	$\phi_{\text{AD}} = \frac{f(X_{\text{IN}})}{8}$	$\phi_{\text{AD}} = \frac{f(X_{\text{IN}})}{4}$
Conversion time (Note)	$f(X_{\text{IN}}) = 8\text{MHz}$	57.0 μs
	$f(X_{\text{IN}}) = 16\text{MHz}$	28.5 μs
	$f(X_{\text{IN}}) = 25\text{MHz}$	18.24 μs

Note: Conversion time per one analog input pin.

$f(X_{\text{IN}})$: External clock input frequency

FUNCTIONAL DESCRIPTION

2.11 A-D converter

(2) A-D sweep pin selection register

The A-D sweep pin selection register (address 1F₁₆) is used to select the analog input pins in single sweep mode and repeat sweep mode. Figure 2.11.3 shows the structure of the A-D sweep pin selection register.

The number of analog input pins can be selected among either 2, 4, 6, or 8 pins with bits 0 and 1 (A-D sweep pin selection bits). Table 2.11.5 shows the relationship between the A-D sweep pin selection bits and the number of the analog input pins. Pins AN₀–AN₇ (8 pins) are selected for analog input pins because the A-D sweep pin selection bits are set to “11” at reset.

The analog input pins must be selected before occurring the trigger and A-D conversion starts.
Pins not selected as analog input pin can be used as normal I/O ports.

Table 2.11.5 Relationship between A-D sweep pin selection bits and number of analog input pins

b1	b0	A-D sweep pin selection bits
0	0	AN ₀ , AN ₁ (2 pins)
0	1	AN ₀ –AN ₃ (4 pins)
1	0	AN ₀ –AN ₅ (6 pins)
1	1	AN ₀ –AN ₇ (8 pins)

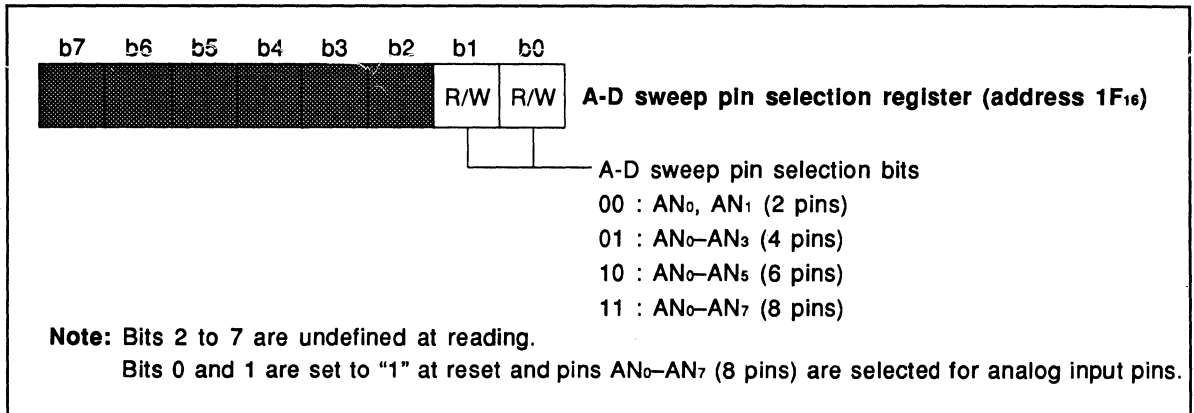


Fig. 2.11.3 A-D sweep pin selection register structure

FUNCTIONAL DESCRIPTION

2.11 A-D converter

(3) A-D register I (I=0 to 7)

The A-D register i (addresses 20₁₆ to 2E₁₆) are 8-bit read-only registers. The conversion results (contents of successive approximation register) are stored in these registers. A-D register i corresponds to each analog input pin. The contents of the A-D register i can be read during A-D conversion. However, if the A-D register i corresponding to the analog input being converted is read, the conversion result completed before reading is obtained.

Table 2.11.6 shows the relationship between the analog input pin and A-D register i and Figure 2.11.4 shows the structure of A-D register i.

Table 2.11.6 Relationship between analog input pin and A-D register I

Analog input pin	Register containing the result	Address
AN ₀ pin	A-D register 0	20 ₁₆
AN ₁ pin	A-D register 1	22 ₁₆
AN ₂ pin	A-D register 2	24 ₁₆
AN ₃ pin	A-D register 3	26 ₁₆
AN ₄ pin	A-D register 4	28 ₁₆
AN ₅ pin	A-D register 5	2A ₁₆
AN ₆ pin	A-D register 6	2C ₁₆
AN ₇ pin	A-D register 7	2E ₁₆

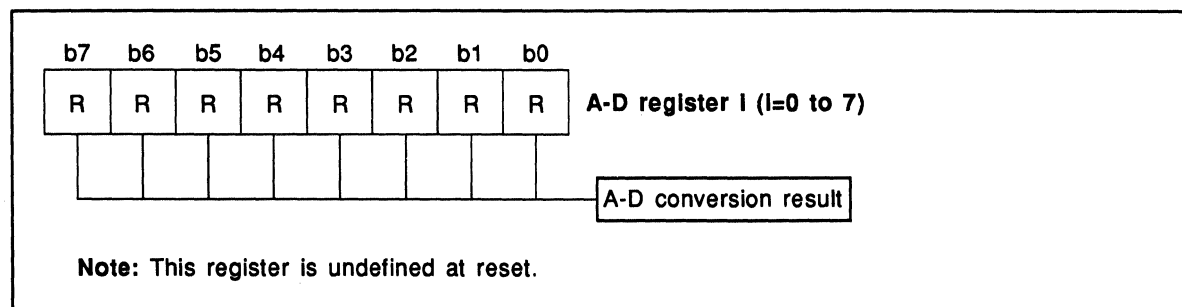


Fig. 2.11.4 A-D register I structure

(4) Comparator and successive approximation register

The compare reference voltage V_{ref} and analog input voltage V_{IN} are compared for bits 7 to 0 of successive approximation register and the result is set in each bit. Comparison starts from bit 7 and the contents of this register (conversion result) are transferred to the A-D register i after comparing bit 0. The contents of the successive approximation register change according to the comparison result of each bit. Therefore, the compare reference voltage V_{ref} also changes according to the contents of the successive approximation register. The analog input voltage V_{IN} is selected by the decoder (refer to section "2.11.3 Successive approximation conversion").

FUNCTIONAL DESCRIPTION

2.11 A-D converter

(5) A-D conversion Interrupt control register

The A-D conversion interrupt control register (address 70₁₆) consists of interrupt priority level selection bits and interrupt request bit. Figure 2.11.5 shows the structure of the A-D conversion interrupt control register followed by description of each bit. Use the **SEB** and **CLB** instructions when setting the A-D conversion interrupt control register. Refer to section "2.6 Interrupts" for more information concerning interrupts.

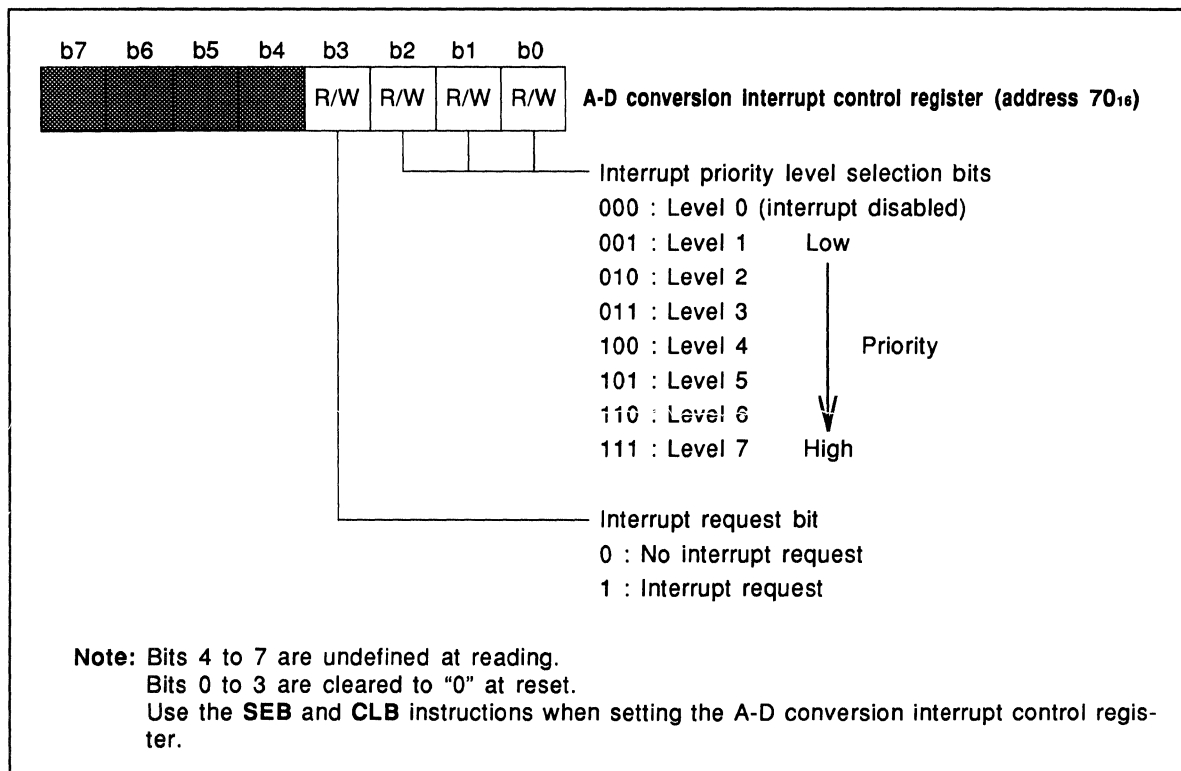


Fig. 2.11.5 A-D conversion interrupt control register structure

●Interrupt priority level selection bits (bits 0 to 2)

These bits are used to select the interrupt priority level. They should be set to a level between 1 and 7 when using an A-D conversion interrupt. When an interrupt request occurs, this level is compared with the processor interrupt priority level (IPL) in the processor status register (PS) and an interrupt is allowed only when this level is higher than IPL (interrupt disable flag I must be "0"). Set these bits to "000" (level 0) to disable only A-D conversion interrupt.

●Interrupt request bit (bit 3)

This bit is set to "1" when an A-D conversion interrupt request occurs. The interrupt request bit set to "1" is cleared to "0" when the interrupt request is accepted.

This bit can be set or cleared by program.

FUNCTIONAL DESCRIPTION

2.11 A-D converter

(6) Port P7 direction register

The analog input pin is shared with port P7. When using these ports as analog input pins or using port P7₇ as external trigger input pin, the corresponding bit in the port P7 direction register must be set to "0" (input mode).

Figure 2.11.6 shows the relationship between the port P7 direction register (address 11₁₆) and analog input pins.

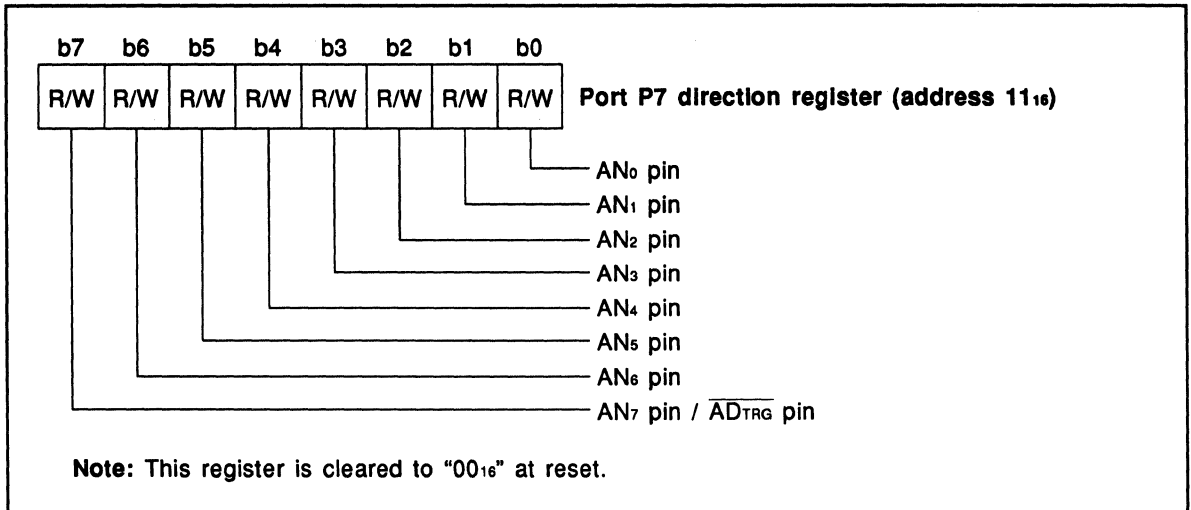


Fig. 2.11.6 Relationship between port P7 direction register and analog input pins

FUNCTIONAL DESCRIPTION

2.11 A-D converter

2.11.3 Successive approximation conversion

A-D conversion starts when an internal or an external trigger is occurred.

A-D conversion is performed by successive approximation. When A-D conversion starts, the following operations are performed automatically to convert analog values to digital values.

① Initialization of successive approximation register

The successive approximation register is cleared to "00₁₆".

② Setting the most significant bit (bit 7)

The successive approximation register bit 7 is set to "1". Then the compare reference voltage V_{ref} is compared with the analog input voltage V_{IN} and bit 7 changes as follows:

Unchanged if $V_{ref} < V_{IN}$

Cleared to "0" if $V_{ref} > V_{IN}$

The reference voltage input to V_{REF} pin must be set between AV_{SS} and AV_{CC} .

The compare reference voltage V_{ref} depends on the value in the successive approximation register. Table 2.11.7 shows the relationship between V_{ref} and the value in the successive approximation register.

Table 2.11.7 Relationship between successive approximation register and V_{ref}

Contents of successive approximation register	0	1 to 255
Compare reference voltage V_{ref} (V)	0	$V_{REF}/256 \times (n-0.5)$ n : The contents of the successive approximation register

Step ② above is repeated for all bits from bit 7 to bit 0 and the value in the successive approximation register (digital equivalent of the analog input voltage) is transferred to the A-D register i when comparison of bit 0 completes.

Table 2.11.8 shows the change in the successive approximation register and the compare reference voltage during A-D conversion.

FUNCTIONAL DESCRIPTION

2.11 A-D converter

A-D conversion successively convert the analog input voltage V_{IN} in the process of comparison shown in Table 2.11.8. Therefore, conversion results become meaningless values if the input voltage of the analog input pin changes during the A-D conversion. In order to avoid the cases, the analog input voltage must be controlled externally without changing during A-D conversion. A-D conversion interval is $\phi_{AD} \times 57$ cycles (ϕ_{AD} is $f(X_{IN})/8$ or $f(X_{IN})/4$) per pin.

Table 2.11.8 Change in successive approximation register and compare reference voltage during A-D conversion

	Successive approximation register	Compare voltage V_{ref}
Conversion start	<div style="text-align: center;"> b7 b0 0 0 0 0 0 0 0 0 </div>	0 [V]
↓		
First comparison	<div style="text-align: center;"> 1 0 0 0 0 0 0 0 </div>	$\frac{V_{REF}}{2} - \frac{V_{REF}}{512}$ [V]
↓		
Second comparison	<div style="text-align: center;"> n_7 1 0 0 0 0 0 0 ↙ 1st comparison result </div>	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{512}$ [V]
↓		
Third comparison	<div style="text-align: center;"> n_7 n_6 1 0 0 0 0 0 ↙ 2nd comparison result = = = </div>	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{512}$ [V]
⋮		
↓		
Eighth comparison	<div style="text-align: center;"> n_7 n_6 n_5 n_4 n_3 n_2 n_1 1 </div>	
↓		
Conversion completed	<div style="text-align: center;"> n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0 </div>	

2.11.4 A-D conversion mode

Four different A-D conversion modes can be selected with the A-D conversion mode selection bits. In each mode, the trigger selection bit is used to determine whether to use a software trigger (internal trigger) or an external input signal (external trigger).

Conversion operating of each A-D conversion mode is described below for cases selecting an internal trigger and an external trigger.

(1) One-shot mode [A-D control register bits 4, 3="00"]

In one-shot mode, the input voltage to the one analog input pin selected with the analog input selection bits is converted and an A-D conversion interrupt request occurs when conversion completes. The analog input pin must be selected before occurring the trigger. Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is occurred and A-D conversion starts. After 57 cycles of ϕ_{AD} , A-D conversion completes, the contents of the successive approximation register (converted result) are transferred to the A-D register i . At the same time, the A-D conversion interrupt request occurs and the interrupt request bit is set to "1". Then the A-D conversion start flag is cleared to "0" and A-D converter operation stops.

●When an external trigger is selected to start A-D conversion

A-D conversion starts when the input level of the $\overline{AD_{TRG}}$ pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1". When A-D conversion completes after 57 cycles of ϕ_{AD} , the contents of the successive approximation register (converted result) are transferred to the A-D register i . At the same time, the A-D conversion interrupt request occurs and the interrupt request bit is set to "1". At this point, the A-D conversion start flag retains "1". Therefore, A-D conversion can be repeated by occurring another trigger.

A retrigger can be also occurred during A-D conversion. In this cases, comparison stops when occurring retrigger, and then comparison restarts from bit 7 of the successive approximation register.

(2) Repeat mode [A-D control register bits 4, 3="01"]

In repeat mode, the input voltage to the one analog input pin selected with the analog input selection bits is repeatedly converted. No interrupt request occurs and the A-D conversion start flag is not cleared to "0" automatically. The A-D conversion of the selected analog input pin is repeated while the A-D conversion start flag is "1".

The analog input pin must be selected before occurring the trigger. Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is occurred and A-D conversion starts. Each time an A-D conversion completes, the contents of the successive approximation register (converted result) are transferred to the A-D register i . The A-D converter does not stop at this point and conversion is repeated.

●When an external trigger is selected to start A-D conversion

A-D conversion starts when the input level of the AD_{TRG} pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1". Each time an A-D conversion completes, the contents of the successive approximation register (converted result) are transferred to the A-D register i . The A-D converter does not stop at this point and conversion is repeated.

(3) Single sweep mode [A-D control register bits 4, 3="10"]

In single sweep mode, multiple analog input pins can be converted. The number of analog input pins are selected with the A-D sweep pin selection register. The number of analog input pins can be selected among either 2, 4, 6, or 8 pins with the A-D sweep pin selection bits of the A-D sweep pin selection register. The analog input pins must be selected before occurring the trigger.

A-D conversion is performed only for the input voltage of the selected input pins. An A-D conversion interrupt request occurs and the interrupt request bit is set to "1" when all selected pins are converted. The analog input selection bits in the A-D control register are ignored in this mode.

Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is occurred and A-D conversion of the AN₀ pin starts. After the AN₀ pin is converted, the selected analog input pins are converted in sequence. The converted result is transferred from the successive approximation register to the corresponding A-D register *i* each time a pin is converted. When all selected pins are converted, an A-D conversion interrupt request occurs and the interrupt request bit is set to "1". At this point, the A-D conversion start flag is cleared to "0" and the A-D converter stops.

●When an external trigger is selected to start A-D conversion

The selected pins are converted in order starting from the AN₀ pin similar to selecting an internal trigger when the input level of the AD_{TRG} pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1". The converted result is transferred from the successive approximation register to the corresponding A-D register *i* each time a pin is converted. When all selected pins are converted, an A-D conversion interrupt request occurs and the interrupt request bit is set to "1". At this point, the A-D conversion start flag retains "1". Therefore, A-D conversion can be repeated from the AN₀ pin by occurring another trigger.

A retrigger can be also occurred during A-D conversion. In this cases, executing pin conversion stops when occurring retrigger, and then conversion restarts from AN₀ pin.

(4) Repeat sweep mode [A-D control register bits 4, 3="11"]

In repeat sweep mode, the A-D sweep pin selection register can be used to select multiple analog input pins to be converted as with single sweep mode. Conversion is performed in order from the AN₀ pin. After converting all selected pins, A-D converter does not stop, but repeats conversion from the AN₀ pin. No interrupt request occur when all selected pins are converted.

The analog input selection bits in the A-D control register are ignored in this mode.

Pins not selected as analog input pin can be used as normal I/O ports.

●When an internal trigger is selected to start A-D conversion

When the A-D conversion start flag is set to "1", an internal trigger is occurred and A-D conversion starts from the AN₀ pin. After the AN₀ pin is converted, the selected analog input pins are converted in sequence. The converted result is transferred from the successive approximation register to the corresponding A-D register *i* each time a pin is converted.

When all selected pins are converted, conversion is repeated from the AN₀ pin. Conversion is repeated until the A-D conversion start flag is cleared to "0".

●When an external trigger is selected to start A-D conversion

The selected analog input pins are converted in order starting from the AN₀ pin similar to selecting an internal trigger when the input level of the AD_{TRG} pin changes from "H" to "L" (falling edge) while the A-D conversion start flag is set to "1". The converted result is transferred from the successive approximation register to the corresponding A-D register *i* each time a pin is converted. Conversion is repeated until the A-D conversion start flag is cleared to "0".

FUNCTIONAL DESCRIPTION

2.11 A-D converter

[Precautions when using A-D converter]

1. Analog input pins must be selected (with analog input selection bits of A-D control register and A-D sweep pin selection register) before an internal or an external trigger is occurred.
 2. The port P7 direction register bit corresponding to the pin selected as analog input pin and external trigger input pin (port P7₇) must be set to "0" (input mode).
 3. When an external trigger is selected, port P7₇ functions as $\overline{AD_{TRG}}$ pin. If at the same time, AN₇ pin is selected as the analog input pin, the external trigger input signal is converted and the converted result is transferred to A-D register 7.
 4. Analog input selection bits of A-D control register must not be written during A-D conversion in single sweep mode and repeat sweep mode.
The contents of analog input selection bits change to pins during A-D conversion.
Pins to convert are forced to change while converting if analog input selection bits are written during A-D conversion.
 5. When not using A-D converter, set the AV_{CC} pin, AV_{SS} pin and V_{REF} pin as follows.
 - AV_{CC} pinConnect to V_{CC} pin.
 - AV_{SS} pin and V_{REF} pinConnect to V_{SS} pin.
- * Refer to "Appendix 5. Setting example of unused pins".

FUNCTIONAL DESCRIPTION

2.12 Watchdog timer

2.12 Watchdog timer

The watchdog timer is a 12-bit timer that is used to detect unexpected execution sequence caused by software run-away. It is also used to stabilize the oscillator when returning from the STP instruction. Figure 2.12.1 shows the block diagram of the watchdog timer.

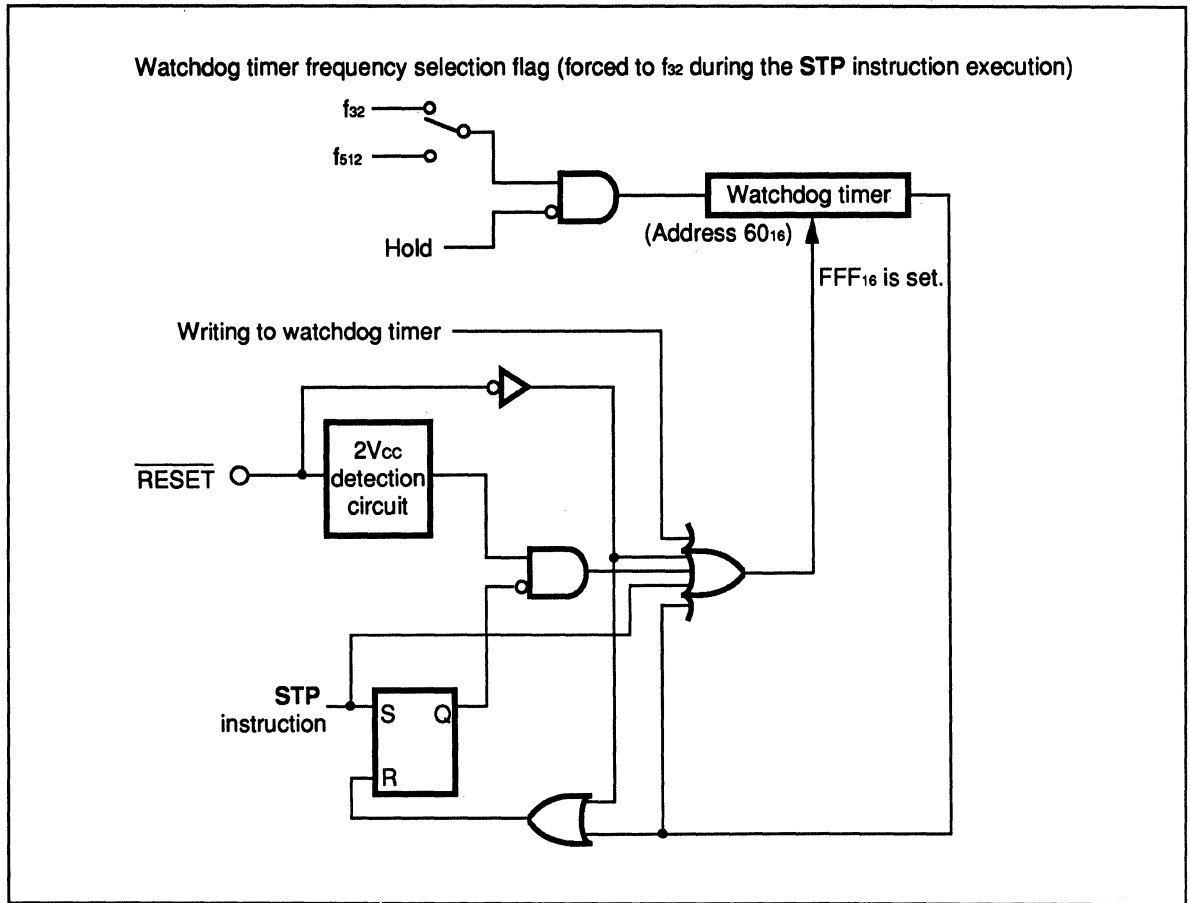


Fig. 2.12.1 Watchdog timer block diagram

FUNCTIONAL DESCRIPTION

2.12 Watchdog timer

2.12.1 Operation description

The watchdog timer (address 60₁₆) consists of 12 bits and its contents are decremented by 1 each time the clock selected with the watchdog timer frequency selection flag (bit 0 at address 61₁₆) is input to the watchdog timer.

The watchdog timer frequency selection flag is cleared to "0" at reset, and f_{512} (external clock input frequency $f(X_{IN})$ divided by 512) is selected as the watchdog timer count source after reset. It can be also set to f_{32} (external clock input frequency $f(X_{IN})$ divided by 32) by setting the watchdog timer frequency selection flag by software. Figure 2.12.2 shows the structure of the watchdog timer frequency selection flag.

When there is a reset, "FFF₁₆" is set in the watchdog timer. Then the count source f_{512} is counted after removing reset. The contents of the watchdog timer are decremented by 1 each time a clock is input. An interrupt request occurs when the most significant bit of the watchdog timer becomes "0" after 2048 counts. The watchdog timer interrupt is a non-maskable interrupt with the highest priority. Processor interrupt priority level (IPL) is set to level 7 when accepting the interrupt request.

An arbitrary value cannot be set in the watchdog timer. A value "FFF₁₆" is automatically set in the watchdog timer when there is a reset, when the STP instruction is executed, or when a writing operation is performed in the watchdog timer. The watchdog timer is a write-only register and its contents are undefined at reading.

The watchdog timer is in Hold state and the clock input of the watchdog timer is disabled while "L" level is applied to the HOLD pin (in Hold state).

When using the watchdog timer to detect a software run-away, a dummy data must be written to the watchdog timer by software before the most significant bit of the watchdog timer becomes "0". Then if this code is not executed due to a software run-away, the most significant bit of the watchdog timer becomes "0" and an interrupt occurs. Thereafter, the control should be passed to the interrupt service routine.

To restart from reset after detecting a software run-away, bit 3 of the processor mode register (software reset bit) must be set to "1" in the watchdog timer interrupt service routine. In this way, a run-away software can be automatically reset and returned to normal routine.

In addition to detecting a software run-away, the watchdog timer is also used as a return timer from a stop mode (halting of oscillating circuit with the STP instruction). When the STP instruction is executed, the watchdog timer count source is forced to f_{32} and "FFF₁₆" is set in the watchdog timer. Then when the watchdog timer is started with an external interrupt and when the most significant bit of the watchdog timer becomes "0" after 2048 counts, a supply of internal clock ϕ starts. This is because some time is required for the oscillator to stabilize. In case that watchdog timer is used as a return timer from a stop mode, watchdog timer interrupt does not occur. Refer to section "4.2 Clock generating circuit" for more detail concerning the stop mode.

In order to stop the watchdog timer (disable its function), twice of the V_{CC} voltage must be applied to the RESET pin. During this time, the watchdog timer stops with "FFF₁₆" set.

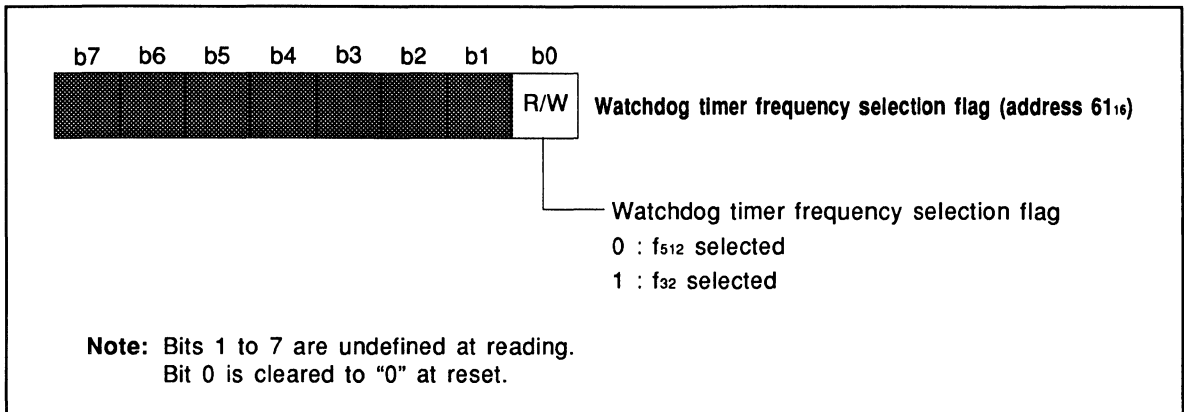


Fig. 2.12.2 Watchdog timer frequency selection flag structure

FUNCTIONAL DESCRIPTION

2.13 Hold function

2.13 Hold function

The microcomputer is in Hold state while "L" level is input to the $\overline{\text{HOLD}}$ pin. Table 2.13.1 shows the state of the microcomputer during Hold.

The $\overline{\text{HOLD}}$ input ("L" level input) to the $\overline{\text{HOLD}}$ pin is accepted at the falling edge of the internal clock ϕ from "H" to "L" level while the bus is unused.

When the $\overline{\text{HOLD}}$ input is accepted, ϕ_{CPU} (CPU operating clock: $f(X_{\text{IN}})/2$) stops at "L" level, and the output level of $\overline{\text{E}}$ pin becomes "H" after completing the executing bus cycle. In addition, "L" level is output from the HLDA pin to indicate externally that the microcomputer is in Hold state. Address/data bus (pins $A_0\text{--}A_7$, $A_8/D_8\text{--}A_{15}/D_{15}$, $A_{16}/D_0\text{--}A_{23}/D_7$), R/W pin and $\overline{\text{BHE}}$ pin are floated during Hold. These pins are floated after one cycle of the internal clock ϕ later than the HLDA pin changes from "H" to "L" level.

Only ϕ_{CPU} is stopped during Hold. The oscillator remains operating so that internal peripherals can be operated. However, the watchdog timer is stopped.

Hold state can be removed by returning the $\overline{\text{HOLD}}$ pin to "H" level. In this case, $\overline{\text{HOLD}}$ input ("H" level input) is also accepted at the falling edge of the internal clock ϕ from "H" to "L" level while the bus is unused. At the removing of Hold state, these ports are removed from Hold state after one cycle of the internal clock ϕ later than the HLDA pin changes from "L" to "H" level.

Figure 2.13.1 shows the timing example when the Hold.

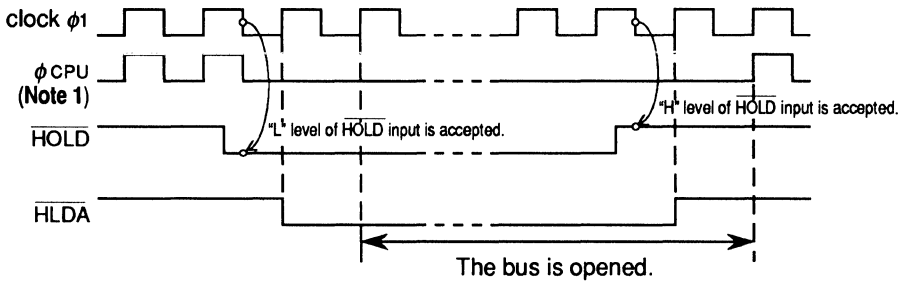
Table 2.13.1 Microcomputer state during Hold

Parameter	State during Hold
Oscillator	Operating
Internal clock ϕ	Operating
Clock ϕ_1	Output
ϕ_{CPU}	Stopped at "L" level
$\overline{\text{E}}$ pin	Stopped at "H" level
Pins $A_0\text{--}A_7$, $A_8/D_8\text{--}A_{15}/D_{15}$, $A_{16}/D_0\text{--}A_{23}/D_7$, R/W pin, $\overline{\text{BHE}}$ pin	Floating
ALE pin, HLDA pin	Stopped at "L" level
Ports P4 ₃ –P4 ₇ , P5, P6, P7, P8	Retain the state at inputting "L" level to the $\overline{\text{HOLD}}$ pin
Watchdog timer	Stopped

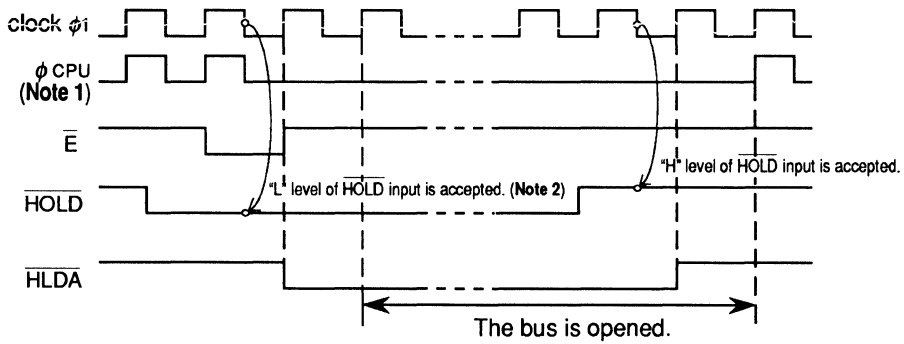
FUNCTIONAL DESCRIPTION

2.13 Hold function

(a) $\overline{\text{HOLD}}$ input is performed when the bus is unused



(b) $\overline{\text{HOLD}}$ input is performed when the bus is used



Note 1: ϕ CPU is an internal signal and cannot be observed externally.

Note 2: $\overline{\text{E}}$ output is "L" level at this point, however, $\overline{\text{HOLD}}$ input is accepted because the bus is unused internally.

Fig. 2.13.1 Timing example when the Hold

FUNCTIONAL DESCRIPTION

2.14 Ready function

2.14 Ready function

The microcomputer is in Ready state while "L" level is input to the $\overline{\text{RDY}}$ pin. Table 2.14.1 shows the state of the microcomputer during Ready.

The Ready function is used when externally connecting slow memory.

The $\overline{\text{RDY}}$ input ("L" level input) to the $\overline{\text{RDY}}$ pin is accepted at the falling edge of the internal clock ϕ from "H" to "L" level. When the $\overline{\text{RDY}}$ input is accepted, internal clock ϕ and ϕ_{CPU} (CPU operating clock: $f(X_{\text{IN}})/2$) stop at "L" level. Address/data bus (pins A_0-A_7 , $A_8/D_8-A_{15}/D_{15}$, $A_{16}/D_0-A_{23}/D_7$), R/W pin, $\overline{\text{BHE}}$ pin, ALE pin and ports P4-P8 retain the state at inputting "L" level to the $\overline{\text{RDY}}$ pin.

During Ready state, the internal clock ϕ and ϕ_{CPU} are stopped, but the oscillator remains operating so that internal peripherals can be operated.

Ready state can be removed by returning the $\overline{\text{RDY}}$ pin to "H" level. In this case, $\overline{\text{RDY}}$ input ("H" level input) is also accepted at the falling edge of the internal clock ϕ from "H" to "L" level.

Figure 2.14.1 shows the timing example when the Ready.

Table 2.14.1 Microcomputer state during Ready

Parameter	State during Ready
Oscillator	Operating
Internal clock ϕ	Stopped at "L" level
Clock ϕ_1	Output
ϕ_{CPU}	Stopped at "L" level
$\overline{\text{E}}$ pin	Stopped at either "H" level or "L" level
Pins A_0-A_7 , $A_8/D_8-A_{15}/D_{15}$, $A_{16}/D_0-A_{23}/D_7$, R/W pin, $\overline{\text{BHE}}$ pin, ALE pin, ports P4 ₃ -P4 ₇ , P5-P8	Retain the state at inputting "L" level to the $\overline{\text{RDY}}$ pin
Watchdog timer	Operating

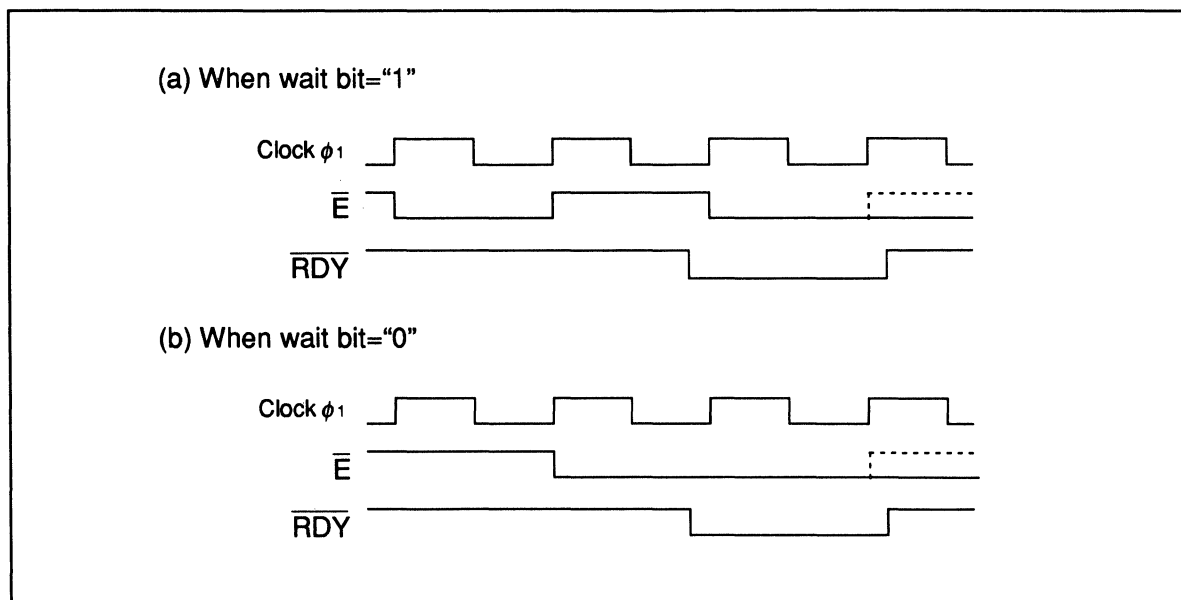


Fig. 2.14.1 Timing example when the Ready



CHAPTER 3

RESET

- 3.1 Reset
- 3.2 Reset circuit
- 3.3 Software reset

RESET

3.1 Reset

3.1 Reset

The CPU becomes reset state when "L" level is applied to the $\overline{\text{RESET}}$ pin. Reset state is removed and program execution starts from address set in the reset vector table when "H" level is then applied to the $\overline{\text{RESET}}$ pin.

3.1.1 Reset operation

The CPU becomes reset state when "L" level is applied to the $\overline{\text{RESET}}$ pin in case the supply voltage is $5V \pm 10\%$. When oscillation is stable, the "L" level must be applied for at least $2\mu\text{s}$.

Apply "L" level to the $\overline{\text{RESET}}$ pin for sufficient interval (approximately 10ms) before returning to "H" level if sufficient time is required for the oscillator to stabilize such as reset during stop mode entered with the STP instruction. Reset state is removed if "H" level is applied to the $\overline{\text{RESET}}$ pin while in reset state.

When reset state is removed, program execution starts from the address formed by using the contents of address FFFF_{16} at bank 0_{16} as high-order and address FFFE_{16} at bank 0_{16} as low-order. Figure 3.1.1 shows the internal processing sequence after removing reset state.

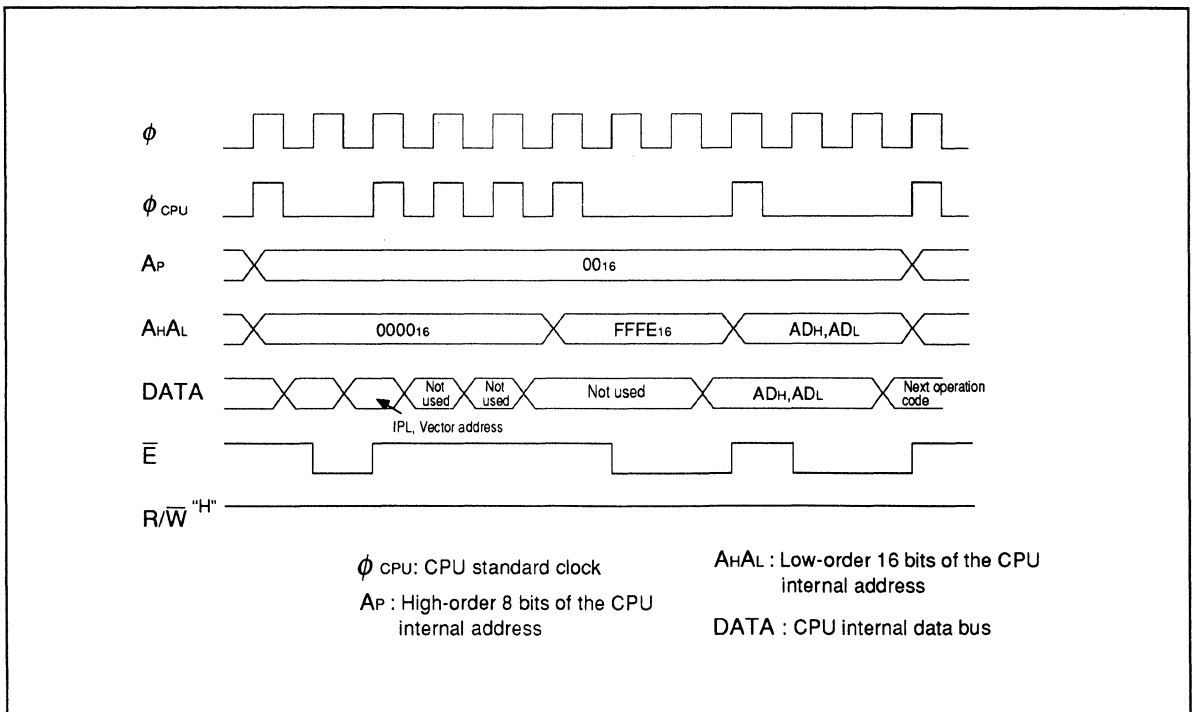


Fig. 3.1.1 Internal processing sequence after removing reset state

3.1.2 Internal registers state at reset

Figure 3.1.2 shows the state of internal registers at reset.

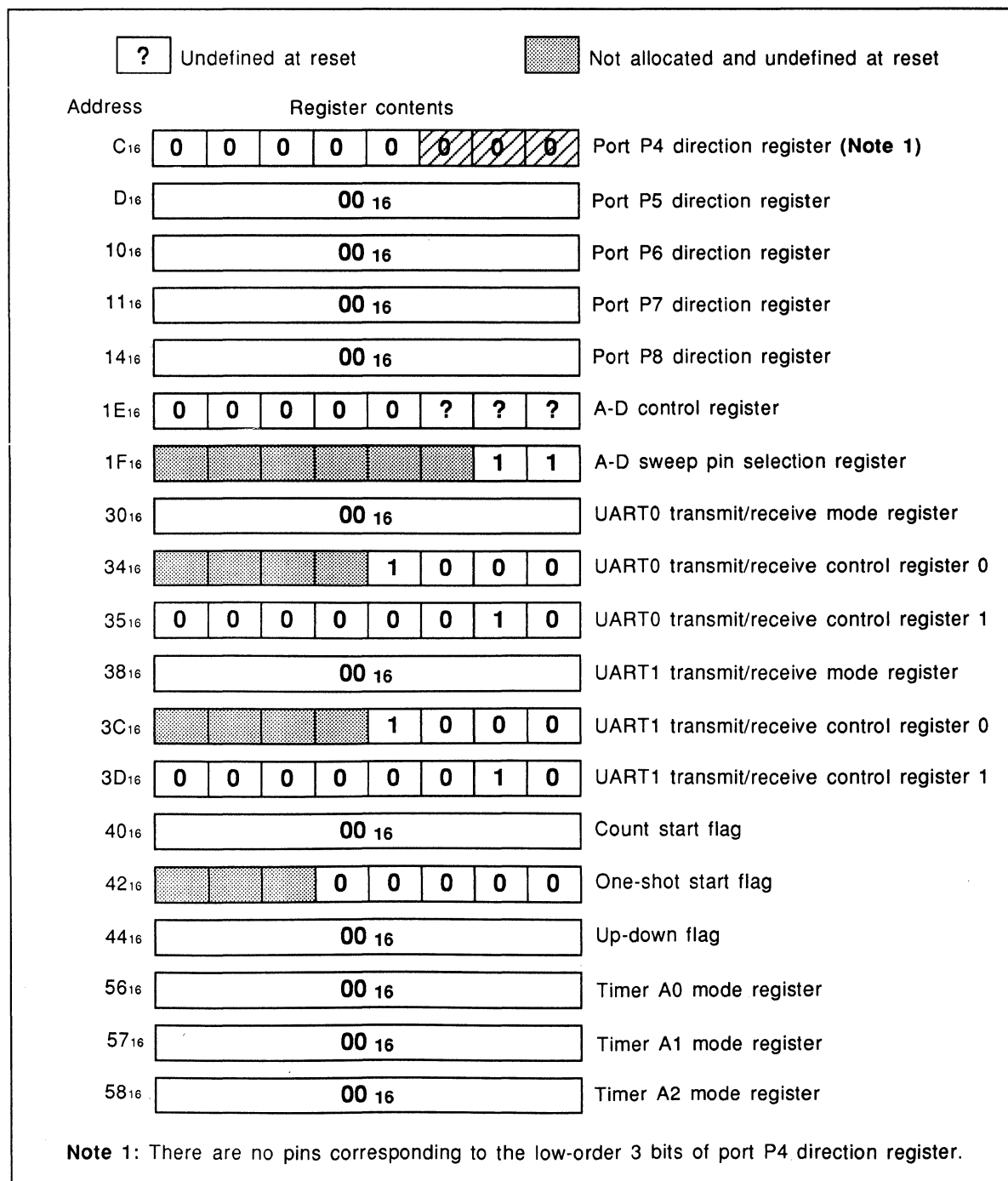


Fig. 3.1.2 Internal registers state at reset (1)

Address	Register contents	
59 ₁₆	00 ₁₆	Timer A3 mode register
5A ₁₆	00 ₁₆	Timer A4 mode register
5B ₁₆	0 0 1 [shaded] 0 0 0 0	Timer B0 mode register
5C ₁₆	0 0 1 [shaded] 0 0 0 0	Timer B1 mode register
5D ₁₆	0 0 1 [shaded] 0 0 0 0	Timer B2 mode register
5E ₁₆	[shaded] 0 0 0 0 0 1 0	Processor mode register
60 ₁₆	FFF ₁₆	Watchdog timer
61 ₁₆	[shaded] [shaded] [shaded] [shaded] [shaded] [shaded] [shaded] 0	Watchdog timer frequency selection flag
62 ₁₆	[shaded] [shaded] 0 0 0 [shaded] 0 0	Waveform output mode register
70 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	A-D conversion interrupt control register
71 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	UART0 transmission interrupt control register
72 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	UART0 receive interrupt control register
73 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	UART1 transmission interrupt control register
74 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	UART1 receive interrupt control register
75 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer A0 interrupt control register
76 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer A1 interrupt control register
77 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer A2 interrupt control register
78 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer A3 interrupt control register
79 ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer A4 interrupt control register
7A ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer B0 interrupt control register
7B ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer B1 interrupt control register
7C ₁₆	[shaded] [shaded] [shaded] [shaded] 0 0 0 0	Timer B2 interrupt control register

Fig. 3.1.2 Internal registers state at reset (2)

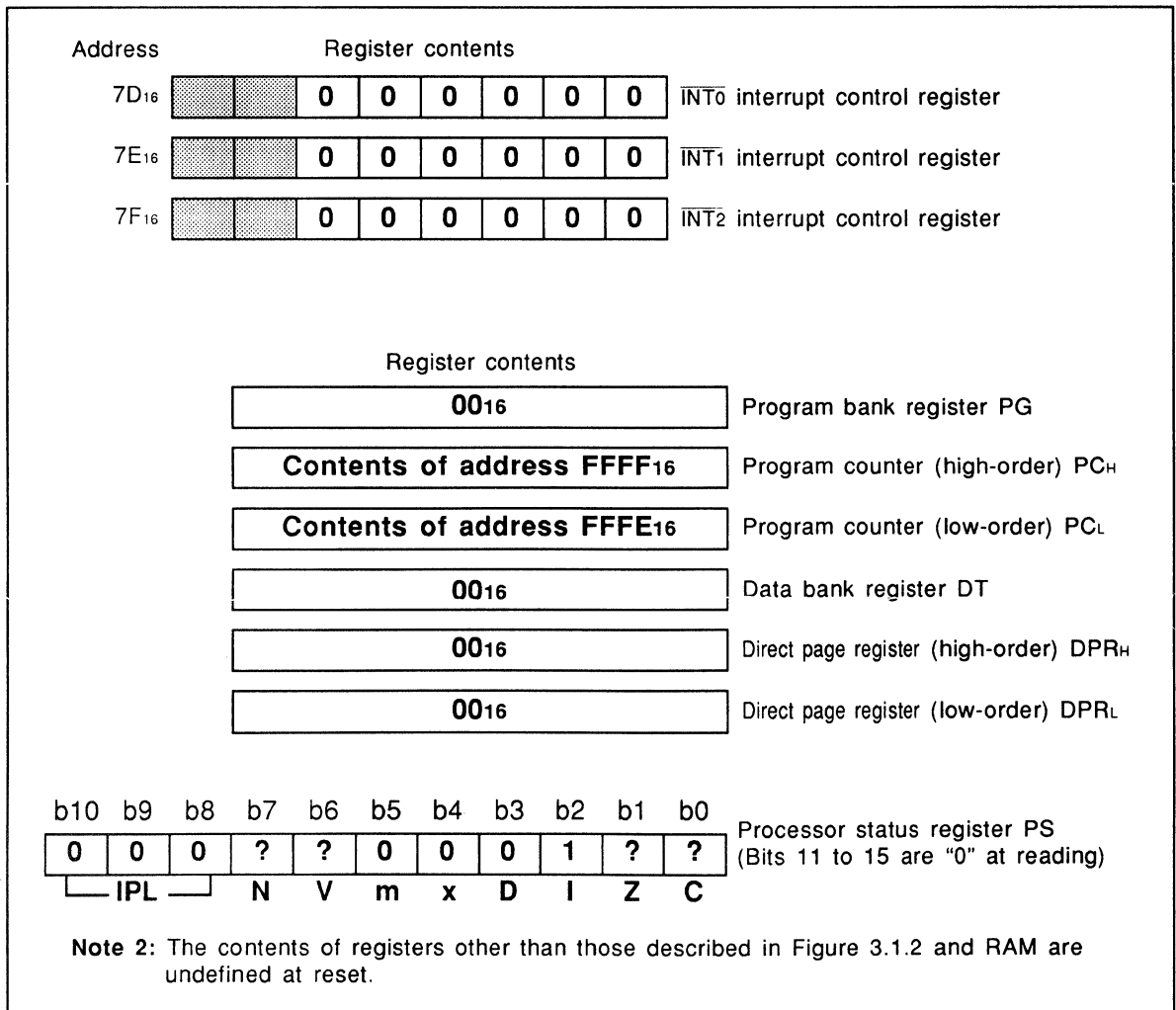


Fig. 3.1.2 Internal registers state at reset (3)

3.1.3 Bus state during reset

Table 3.1.1 shows the state of the address bus, data bus, and control bus during reset ("L" level is applied to RESET pin).

Table 3.1.1 Bus state during reset

Pin	Bus state during reset
A ₀ –A ₇ , A ₈ /D ₈ –A ₁₅ /D ₁₅ , A ₁₆ /D ₀ –A ₂₃ /D ₇	Address output ("H" or "L" level)
E, R/W	"H" level output
BHE	"H" or "L" level output (depends on address output)
ALE	"L" level output
Clock ϕ_1	Operating (output)

3.2 Reset circuit

The reset circuit must be designed so that the reset input voltage is 0.9V or lower when the supply voltage reaches 4.5V as shown in Figure 3.2.1.

Figure 3.2.2 shows the example of power-on reset circuit using a system reset IC M51957AL.

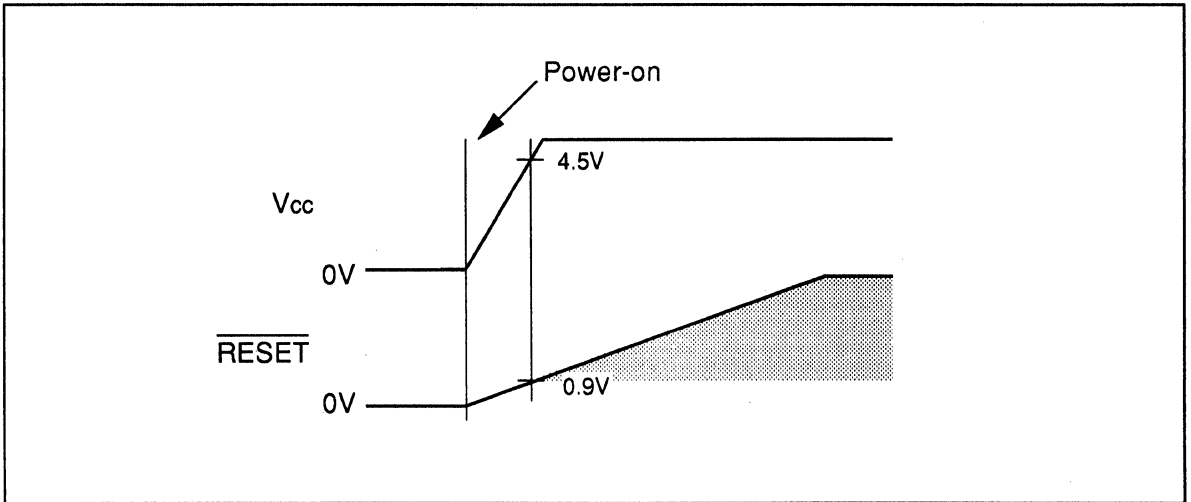


Fig. 3.2.1 Power-on reset condition

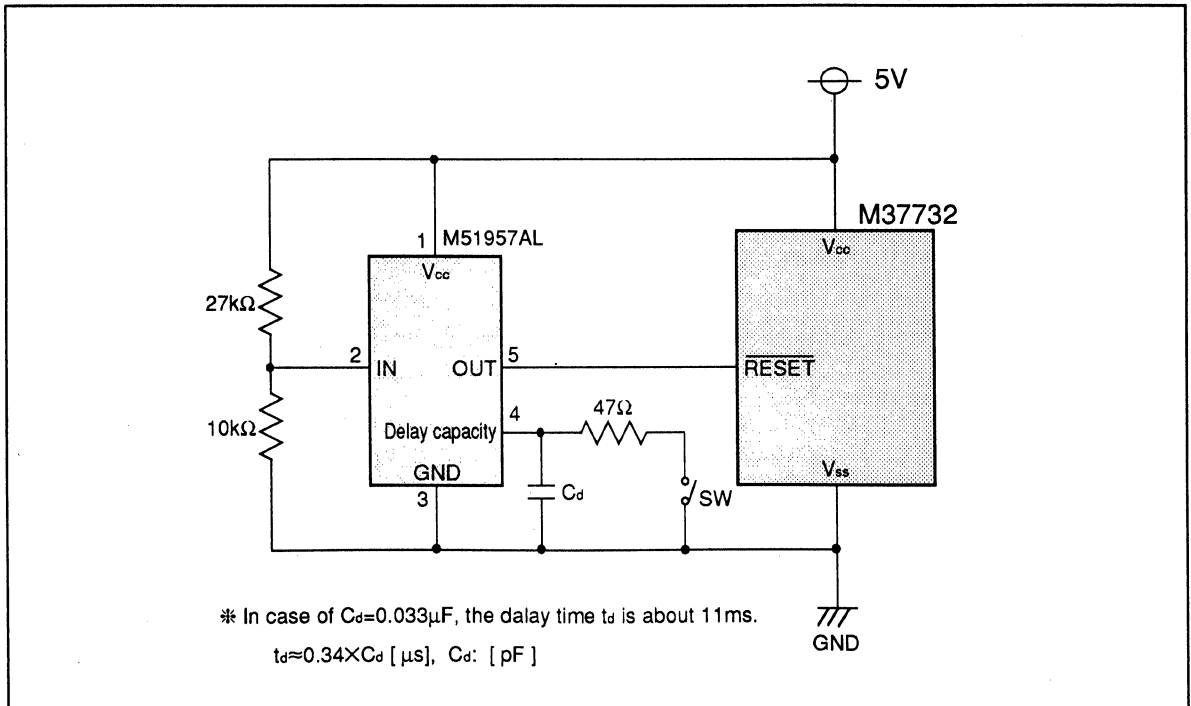


Fig. 3.2.2 Power-on reset circuit example

RESET

3.3 Software reset

3.3 Software reset

The M37732 group can be reset internally by program. Software reset occurs by writing "1" to the software reset bit of the processor mode register (bit 3 of address 5E16). Figure 3.3.1 shows the structure of the processor mode register.

Software reset is the same as hardware reset (when the $\overline{\text{RESET}}$ pin level is returned to "H" after applying "L" level) except that the contents of the internal RAM are preserved. Therefore, the contents of each register just after software reset is initialized to values shown in Figure 3.1.2 except for internal RAM.

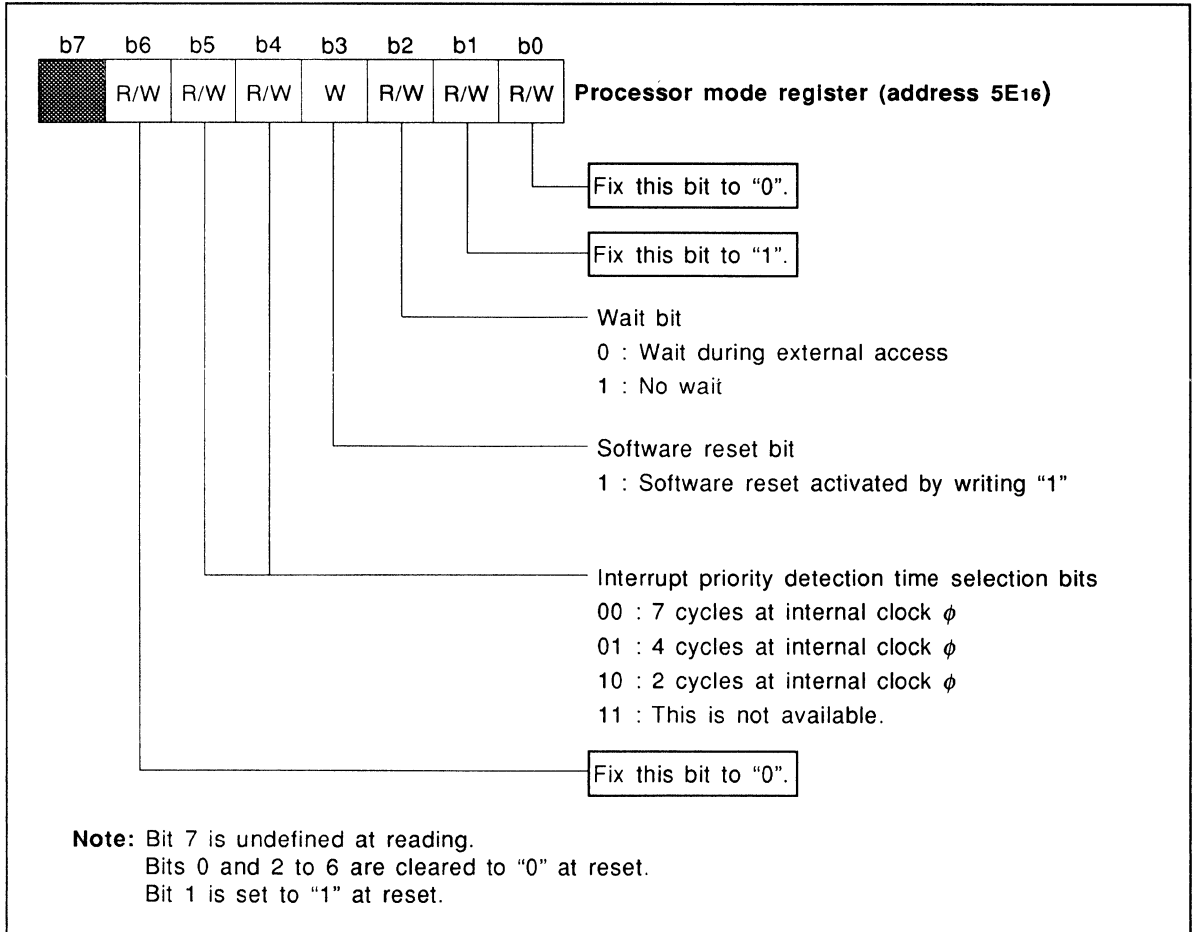
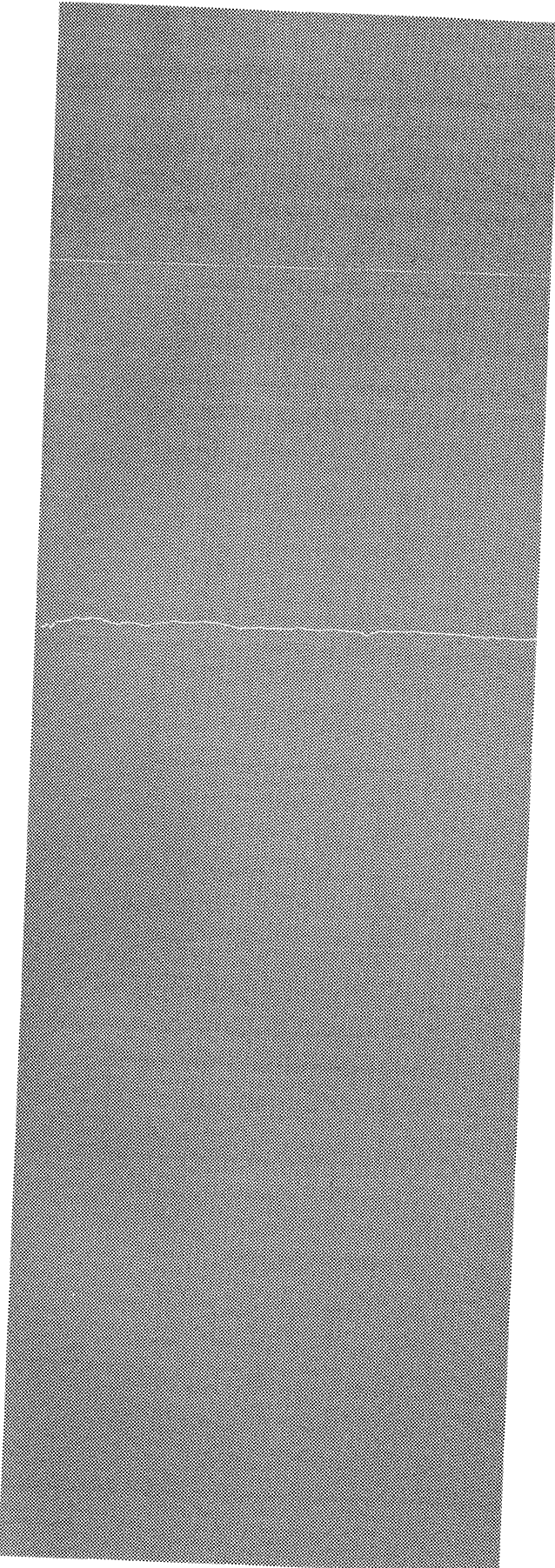


Fig. 3.3.1 Processor mode register structure

MEMORANDUM



CHAPTER 4

OSCILLATING CIRCUIT

- 4.1 Oscillating circuit
- 4.2 Clock generating circuit

OSCILLATING CIRCUIT

4.1 Oscillating circuit

4.1 Oscillating circuit

The M37732 group is equipped with an oscillating circuit to generate the necessary clock. The frequency input to the clock input pin X_{IN} is divided in half to obtain the internal clock ϕ . This ϕ is further divided in half to obtain the bus cycle. Either a ceramic resonator or a crystal resonator can be connected externally to the internal oscillating circuit.

4.1.1 Circuit using a ceramic resonator or a crystal resonator

Figure 4.1.1 shows the circuit example using a ceramic resonator and Figure 4.1.2 shows the circuit example using a crystal resonator. An oscillating circuit is formed by connecting the resonator between X_{IN} pin and X_{OUT} pin as shown in the figures. The circuit constants such as R_i , R_d , C_{IN} , and C_{OUT} must be set to the resonator manufacturer's recommended values.

4.1.2 External clock input circuit

An external clock signal can be supplied to the internal oscillating circuit. Figure 4.1.3 shows the circuit example of external clock input. Note that the external clock must be input from X_{IN} pin, and X_{OUT} pin must be left open.

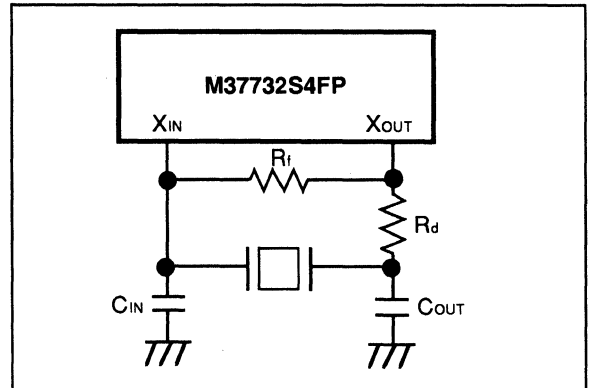


Fig. 4.1.1 Oscillating circuit using a ceramic resonator

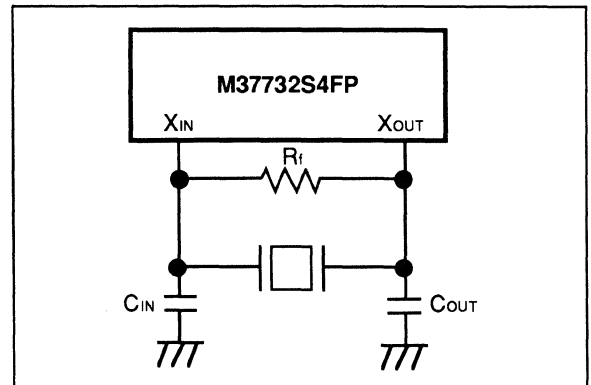


Fig. 4.1.2 Oscillating circuit using a crystal resonator

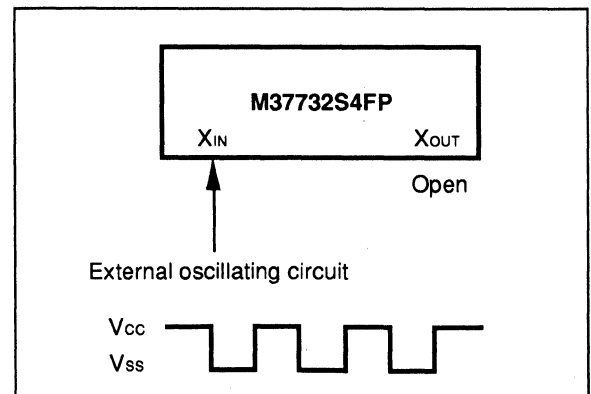


Fig. 4.1.3 External clock input circuit

OSCILLATING CIRCUIT

4.2 Clock generating circuit

4.2 Clock generating circuit

The oscillating circuit consists of oscillating gate which functions as an amplifier to obtain the necessary gain and oscillation control flip-flop to control this amplifier. Therefore, oscillation can be stopped and restarted as necessary. The clock generating circuit shown in Figure 4.2.1 is built in the M37732 group.

When the **STP** instruction is executed, the internal clock ϕ stops oscillation at the "L" level. At the same time, FFF_{16} is set in the watchdog timer and the input to the watchdog timer is connected to f_{32} . This connection is cancelled and connected to the input determined by the contents of the watchdog timer frequency selection flag when the most significant bit of the watchdog timer becomes "0" or when there is a reset. Oscillation is resumed when an interrupt is accepted, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer becomes "0". This is done to avoid the initial unstable interval when using a ceramic resonator.

When the **WIT** instruction is executed, the internal clock ϕ stops at the "L" level, but the oscillator does not stop. Therefore, the timer, serial I/O, and A-D converter can be used. The stopping of internal clock ϕ is cancelled when an interrupt is accepted. An instruction can be executed immediately because the oscillator does not stop.

Refer to "Appendix 3. Stop, wait, one-wait, Ready, Hold state" and "MELPS 7700 SOFTWARE MANUAL" for information concerning the **STP** and **WIT** instructions.

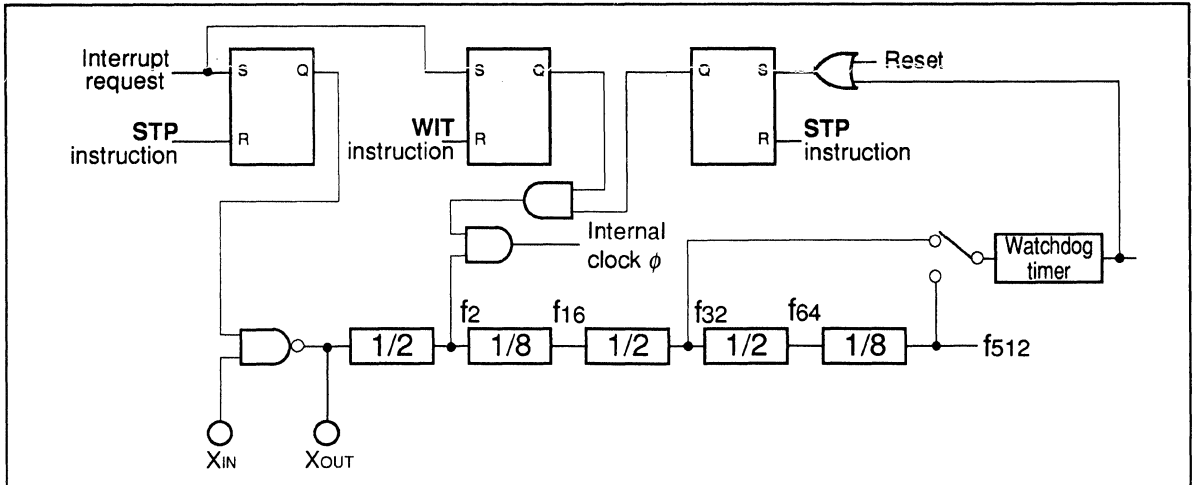
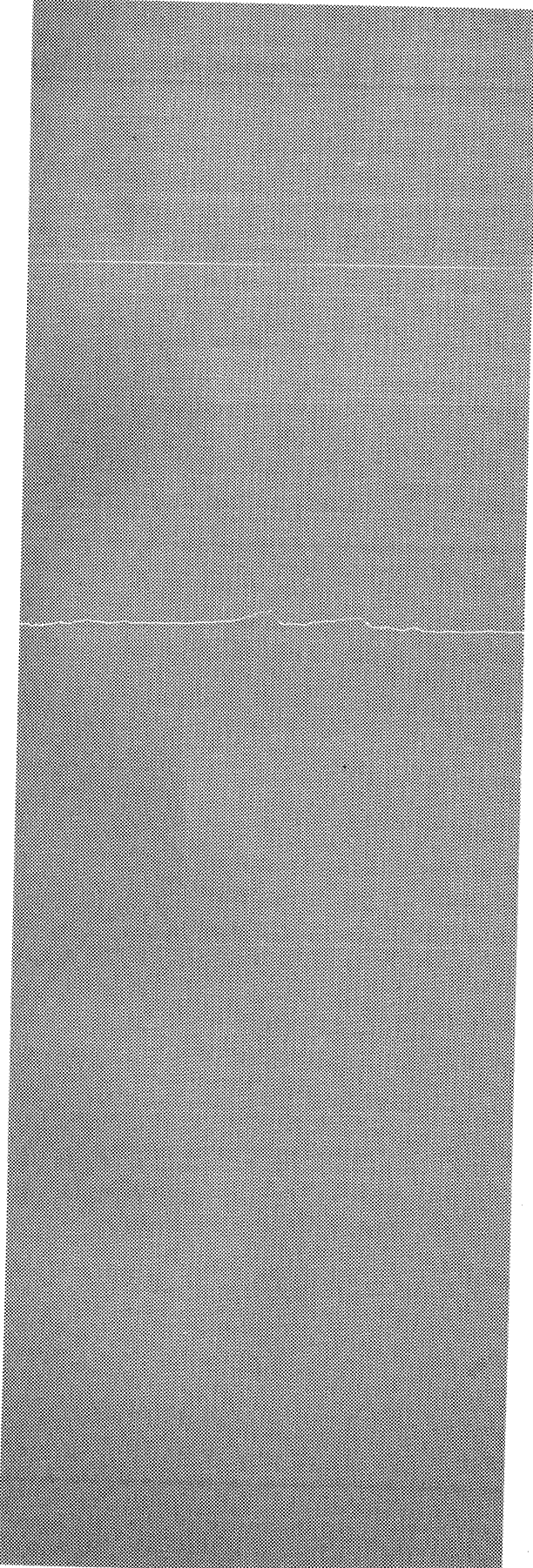


Fig. 4.2.1 Clock generating circuit block diagram

OSCILLATING CIRCUIT

4.2 Clock generating circuit

MEMORANDUM



CHAPTER 5 **ELECTRICAL CHARACTERISTICS**

5.1 Electrical characteristics

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1 Electrical characteristics

5.1.1 Absolute maximum ratings

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V _{cc}	Supply voltage		-0.3 to 7	V
AV _{cc}	Analog supply voltage		-0.3 to 7	V
V _I	Input voltage RESET, CNV _{SS} , BYTE		-0.3 to 12	V
V _I	Input voltage A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₇₀ -P ₇₇ , P ₈₀ -P ₈₇ , V _{REF} , X _{IN} , HOLD, RDY		-0.3 to V _{cc} +0.3	V
V _O	Output voltage A ₀ -A ₇ , A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₇₀ -P ₇₇ , P ₈₀ -P ₈₇ , X _{OUT} , E, ϕ_1 , HLDA, ALE, BHE, R/W		-0.3 to V _{cc} +0.3	V
P _d	Power dissipation	T _a =25°C	300	mW
T _{opr}	Operating temperature		-20 to 85	°C
T _{stg}	Storage temperature		-40 to 150	°C

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1.2 Recommended operating conditions

Recommended operating conditions ($V_{CC}=5V\pm 10\%$, $T_a=-20$ to $85^\circ C$, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V_{CC}	Supply voltage		4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage			V_{CC}		V
V_{SS}	Supply voltage			0		V
AV_{SS}	Analog supply voltage			0		V
V_{IH}	High-level input voltage	P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	0.8 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage	A ₈ /D ₈ –A ₂₃ /D ₇	0.5 V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage	P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage	A ₈ /D ₈ –A ₂₃ /D ₇	0		0.16 V_{CC}	V
I_{OH} (peak)	High-level peak output current	A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , ϕ_1 , HLDA, ALE, BHE, R/W			–10	mA
I_{OH} (avg)	High-level average output current	A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , ϕ_1 , HLDA, ALE, BHE, R/W			–5	mA
I_{OL} (peak)	Low-level peak output current	A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , ϕ_1 , HLDA, ALE, BHE, R/W			10	mA
I_{OL} (avg)	Low-level average output current	A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P4 ₃ –P4 ₇ , P5 ₀ –P5 ₇ , P6 ₀ –P6 ₇ , P7 ₀ –P7 ₇ , P8 ₀ –P8 ₇ , ϕ_1 , HLDA, ALE, BHE, R/W			5	mA
$f(X_{IN})$	External clock input frequency		M37732S4FP		8	MHz
			M37732S4AFP		16	
			M37732S4BFP		25	

Note 1: Average output current is the average value of a 100ms interval.

2: The sum of I_{OL} (peak) for A₀–A₇, A₈/D₈–A₂₃/D₇, HLDA, ALE, BHE, R/W, and port P8 must be 80mA or less, the sum of I_{OH} (peak) for A₀–A₇, A₈/D₈–A₂₃/D₇, HLDA, ALE, BHE, R/W, and port P8 must be 80mA or less, the sum of I_{OL} (peak) for ports P4, P5, P6, P7, and ϕ_1 must be 80mA or less, and the sum of I_{OH} (peak) for ports P4, P5, P6, P7, and ϕ_1 must be 80mA or less.

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1.3 Electrical characteristics and A-D converter characteristics

M37732S4FP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_7_0-P_7_7$, $P_8_0-P_8_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$	3.1			V
		$I_{OH}=-400\mu A$	4.8			
V_{OH}	High-level output voltage E	$I_{OH}=-10mA$	3.4			V
		$I_{OH}=-400\mu A$	4.8			
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_7_0-P_7_7$, $P_8_0-P_8_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$			1.9	V
		$I_{OL}=2mA$			0.43	
V_{OL}	Low-level output voltage E	$I_{OL}=10mA$			1.6	V
		$I_{OL}=2mA$			0.4	
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA_{0IN}-TA_{4IN}$, $TB_{0IN}-TB_{2IN}$, INT_0-INT_2 , AD_{TRG} , CTS_0 , CTS_1 , CLK_0 , CLK_1		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_7_0-P_7_7$, $P_8_0-P_8_7$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=5V$			5	μA
I_{IL}	Low-level input current $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_7_0-P_7_7$, $P_8_0-P_8_7$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=0V$			-5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during re- set.	$f(X_{IN})=8MHz$, square waveform $T_a=25^\circ C$ when clock is stopped.	6	12	mA
			$T_a=85^\circ C$ when clock is stopped.		1	μA
					20	

A-D converter characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			± 3	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
t_{CONV}	Conversion time		28.5			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

M37732S4AFP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=16MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₇₀ –P ₇₇ , P ₈₀ –P ₈₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.1 4.8			V
V_{OH}	High-level output voltage \bar{E}	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.4 4.8			V
V_{OL}	Low-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₇₀ –P ₇₇ , P ₈₀ –P ₈₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$ $I_{OL}=2mA$			1.9 0.43	V
V_{OL}	Low-level output voltage \bar{E}	$I_{OL}=10mA$ $I_{OL}=2mA$			1.6 0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA _{0IN} –TA _{4IN} , TB _{0IN} –TB _{2IN} , INT ₀ –INT ₂ , ADTRG, CTS ₀ , CTS ₁ , CLK ₀ , CLK ₁		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X _{IN}		0.1		0.3	V
I_{IH}	High-level input current A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₇₀ –P ₇₇ , P ₈₀ –P ₈₇ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	$V_I=5V$			5	μA
I_{IL}	Low-level input current A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₇₀ –P ₇₇ , P ₈₀ –P ₈₇ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	$V_I=0V$			–5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V _{SS} during re- set.	$f(X_{IN})=16MHz$, square waveform Ta=25°C when clock is stopped. Ta=85°C when clock is stopped.	12	24	μA
					1	μA
					20	μA

A-D converter characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=16MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			±3	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	2		10	k Ω
t_{CONV}	Conversion time		14.25			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

M37732S4BFP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=25MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{70}-P_{77}$, $P_{80}-P_{87}$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$	3.1			V
		$I_{OH}=-400\mu A$	4.8			
V_{OH}	High-level output voltage \bar{E}	$I_{OH}=-10mA$	3.4			V
		$I_{OH}=-400\mu A$	4.8			
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{70}-P_{77}$, $P_{80}-P_{87}$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$			1.9	V
		$I_{OL}=2mA$			0.43	
V_{OL}	Low-level output voltage \bar{E}	$I_{OL}=10mA$			1.6	V
		$I_{OL}=2mA$			0.4	
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA_{0IN}-TA_{4IN}$, $TB_{0IN}-TB_{2IN}$, INT_0-INT_2 , ADTRG, CTS ₀ , CTS ₁ , CLK ₀ , CLK ₁		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{70}-P_{77}$, $P_{80}-P_{87}$, X_{IN} , \bar{RESET} , CNV_{SS} , BYTE, HOLD, RDY	$V_i=5V$			5	μA
I_{IL}	Low-level input current $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{70}-P_{77}$, $P_{80}-P_{87}$, X_{IN} , \bar{RESET} , CNV_{SS} , BYTE, HOLD, RDY	$V_i=0V$			-5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during re- set.	$f(X_{IN})=25MHz$, square waveform $T_a=25^\circ C$ when clock is stopped.	19	38	mA
			$T_a=85^\circ C$ when clock is stopped.		20	μA

A-D converter characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=25MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			± 3	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
t_{CONV}	Conversion time		9.12			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1.4 Timing requirements

Timing requirements ($V_{CC}=5V\pm 10\%$, $V_{SS}=0V$, $T_a=25^\circ C$, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_c	External clock input cycle time	125		62		40		ns
$t_{w(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{w(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{su(DH-E)}$	Data high-order input setup time	60		45		30		ns
$t_{su(DL-E)}$	Data low-order input setup time	60		45		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{su(RDY-\phi)}$	\overline{RDY} input setup time	70		60		55		ns
$t_{su(HOLD-\phi)}$	HOLD input setup time	70		60		55		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		0		0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns
$t_{h(\phi-RDY)}$	\overline{RDY} input hold time	0		0		0		ns
$t_{h(\phi-HOLD)}$	HOLD input hold time	0		0		0		ns

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	250		125		80		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	125		62		40		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	125		62		40		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	1000		500		320		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	500		250		160		ns

Timer A input (external trigger input in one-shot pulse mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	500		250		160		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		125		80		ns

Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		125		80		ns

Timer A input (up-down input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(UP)}$	TA _{IOUT} input cycle time	5000		2500		2000		ns
$t_{w(UPH)}$	TA _{IOUT} input high-level pulse width	2500		1250		1000		ns
$t_{w(UPL)}$	TA _{IOUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-TIN)}$	TA _{IOUT} input setup time	1000		500		400		ns
$t_{h(TIN-UP)}$	TA _{IOUT} input hold time	1000		500		400		ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB _{IN} input cycle time (one edge count)	250		125		80		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{c(TB)}$	TB _{IN} input cycle time (both edges count)	500		250		160		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width (both edges count)	250		125		80		ns

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

Timer B Input (pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB _{IN} input cycle time	1000		500		320		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width	500		250		160		ns

Timer B Input (pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB _{IN} input cycle time	1000		500		320		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width	500		250		160		ns

A-D trigger input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(AD)}$	AD _{TRG} input cycle time (minimum allowable trigger)	2000		1000		1000		ns
$t_{w(ADL)}$	AD _{TRG} input low-level pulse width	250		125		125		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(CLK)}$	CLK _i input cycle time	500		250		200		ns
$t_{w(CLKH)}$	CLK _i input high-level pulse width	250		125		100		ns
$t_{w(CLKL)}$	CLK _i input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD _i output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD _i hold time	30		30		30		ns
$t_{su(D-C)}$	RxD _i input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD _i input hold time	90		90		90		ns

External interrupt INT_i input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(INH)}$	INT _i input high-level pulse width	250		250		250		ns
$t_{w(INL)}$	INT _i input low-level pulse width	250		250		250		ns

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1.5 Switching characteristics

Switching characteristics ($V_{CC}=5V\pm 10\%$, $V_{SS}=0V$, $T_a=25^\circ C$, unless otherwise noted)

Microprocessor mode (when wait bit is "1")

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")		110		70		45	ns
$t_{pzz(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time	100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time	80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70		45	ns
$t_{pzz(E-DLZ)}$	Floating start delay time		5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		5		ns
$t_{d(\phi_1-HLDA)}$	HLDA output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_w(ALE)$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	BHE output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/W output delay time	100		30		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_h(E-AL)$	Address low-order hold time	50		25		18		ns
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")	9		9		9		ns
$t_h(E-DHQ)$	Data high-order hold time (BYTE="L")	50		25		18		ns
$t_{pzz(E-DHZ)}$	Floating release delay time (BYTE="L")	95		36		18		ns
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")	50		25		18		ns
$t_h(ALE-AH)$	Address high-order hold time	9		9		9		ns
$t_h(E-DLQ)$	Data low-order hold time	50		25		18		ns
$t_{pzz(E-DLZ)}$	Floating release delay time	95		36		18		ns
$t_h(E-BHE)$	BHE hold time	18		18		18		ns
$t_h(E-R/W)$	R/W hold time	18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100		80	ns
$t_w(EL)$	\bar{E} pulse width	220		95		50		ns

Note: Test conditions are shown in Figure 5.1.1.

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

Microprocessor mode (when wait bit = "0", and external memory area is accessed)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		12		ns
$t_{d(E-DH0)}$	Data high-order output delay time (BYTE="L")		110		70		45	ns
$t_{pxz(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time	100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time	80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70		45	ns
$t_{pxz(E-DLZ)}$	Floating start delay time		5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		5		ns
$t_{d(\phi-HLDA)}$	HLDA output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_w(ALE)$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	BHE output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/W output delay time	100		30		20		ns
$t_{d(E-\phi)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_h(E-AL)$	Address low-order hold time	50		25		18		ns
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")	9		9		9		ns
$t_h(E-DH0)$	Data high-order hold time (BYTE="L")	50		25		18		ns
$t_{pxz(E-DHZ)}$	Floating release delay time (BYTE="L")	95		36		18		ns
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")	50		25		18		ns
$t_h(ALE-AH)$	Address high-order hold time	9		9		9		ns
$t_h(E-DLQ)$	Data low-order hold time	50		25		18		ns
$t_{pxz(E-DLZ)}$	Floating release delay time	95		36		18		ns
$t_h(E-BHE)$	BHE hold time	18		18		18		ns
$t_h(E-R/W)$	R/W hold time	18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100		80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time		200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100		80	ns
$t_w(EL)$	E pulse width	470		220		130		ns

Note: Test conditions are shown in Figure 5.1.1.

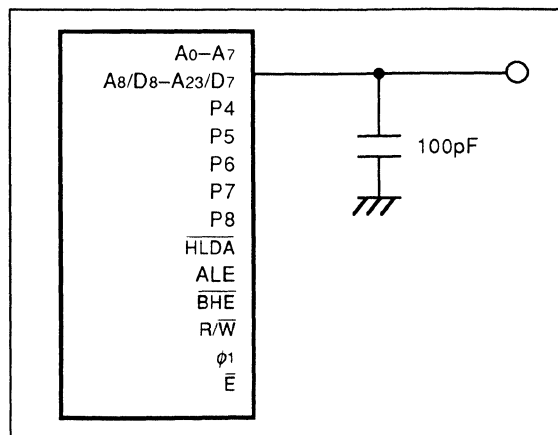


Fig. 5.1.1 Testing circuit for ports

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

5.1.6 Equations for calculating the parameters

Table 5.1.1 shows the equations for calculating the following parameters. Use the equations corresponding to the external clock input frequency $f(XIN)$.

$t_w(ALE)$ ALE pulse width	$t_d(R/W-E)$ R/\bar{W} output delay time
$t_w(EL)$ \bar{E} pulse width	$t_{pzx}(E-DLZ/DHZ)$ Floating release delay time
$t_d(AL/AM/AH-E)$ Address output delay time	$t_h(E-AL/AM)$ Address hold time
$t_d(AM/AH-ALE)$ Address output delay time	$t_h(E-DLQ/DHQ)$ Data hold time
$t_d(BHE-E)$ \bar{BHE} output delay time	

Table 5.1.1 Equations for calculating parameters

	M37732S4FP	M37732S4AFP	M37732S4BFP
$f(XIN)$	$f(XIN) \leq 8MHz$	$8MHz < f(XIN) \leq 16MHz$	$16MHz < f(XIN) \leq 25MHz$
$t_w(ALE)$	$\frac{1 \times 10^9}{f(XIN)} - 35$	$\frac{1 \times 10^9}{f(XIN)} - 27.5$	$\frac{1 \times 10^9}{f(XIN)} - 18$
$t_w(EL)$ Wait bit = "1"	$\frac{2 \times 10^9}{f(XIN)} - 30$		
$t_w(EL)$ Wait bit = "0"	$\frac{4 \times 10^9}{f(XIN)} - 30$		
$t_d(AL/AM/AH-E)$	$100 + \frac{1 \times 10^9}{f(XIN)} - 125$	$30 + \frac{1.2 \times 10^9}{f(XIN)} - 75$	$12 + \frac{1 \times 10^9}{f(XIN)} - 40$
$t_d(BHE-E)$ $t_d(R/W-E)$			$20 + \frac{2 \times 10^9}{f(XIN)} - 40$
$t_{pzx}(E-DLZ/DHZ)$	$\frac{1 \times 10^9}{f(XIN)} - 30$	$\frac{1 \times 10^9}{f(XIN)} - 26$	$\frac{1 \times 10^9}{f(XIN)} - 22$
$t_h(E-AL/AM)$ $t_h(E-DLQ/DHQ)$	$\frac{1 \times 10^9}{2 \times f(XIN)} - 12.5$	$\frac{1 \times 10^9}{2 \times f(XIN)} - 6.25$	$\frac{1 \times 10^9}{2 \times f(XIN)} - 2$
$t_d(AM/AH-ALE)$	$\frac{1 \times 10^9}{f(XIN)} - 45$	$\frac{1 \times 10^9}{f(XIN)} - 38.5$	$\frac{1 \times 10^9}{f(XIN)} - 35$

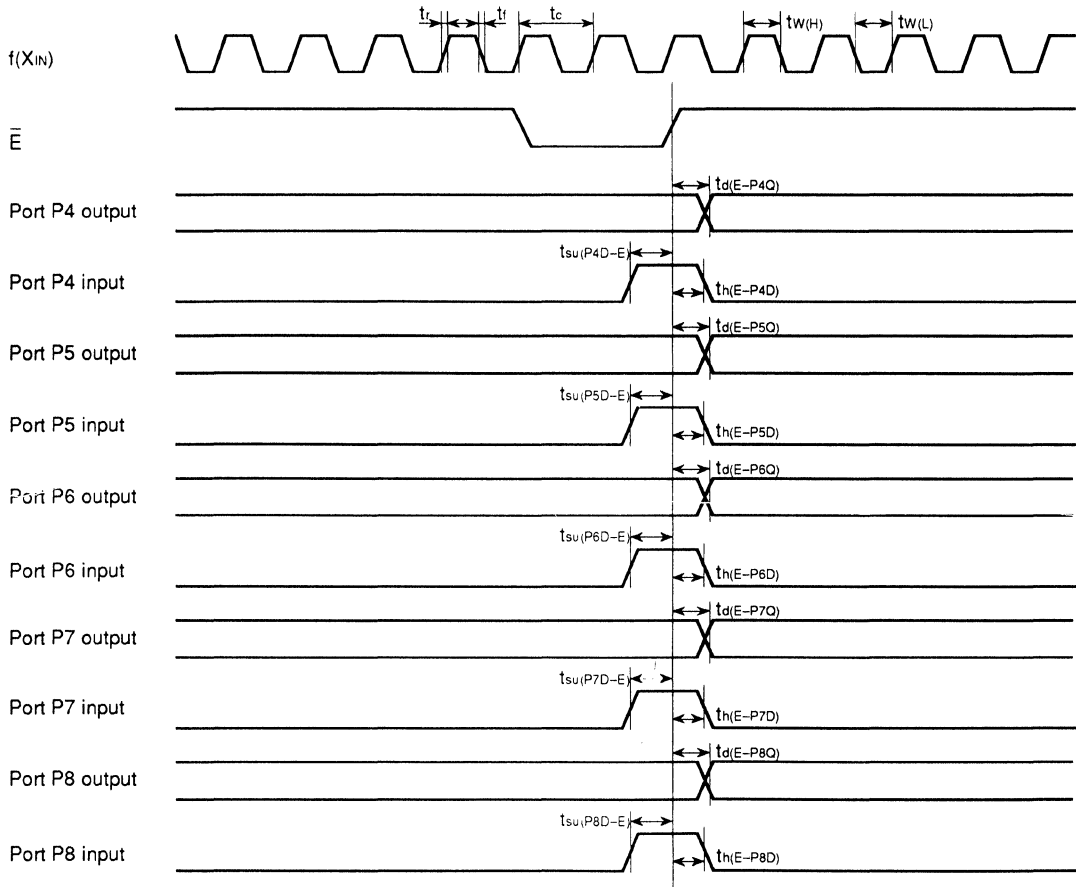
Unit : ns

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

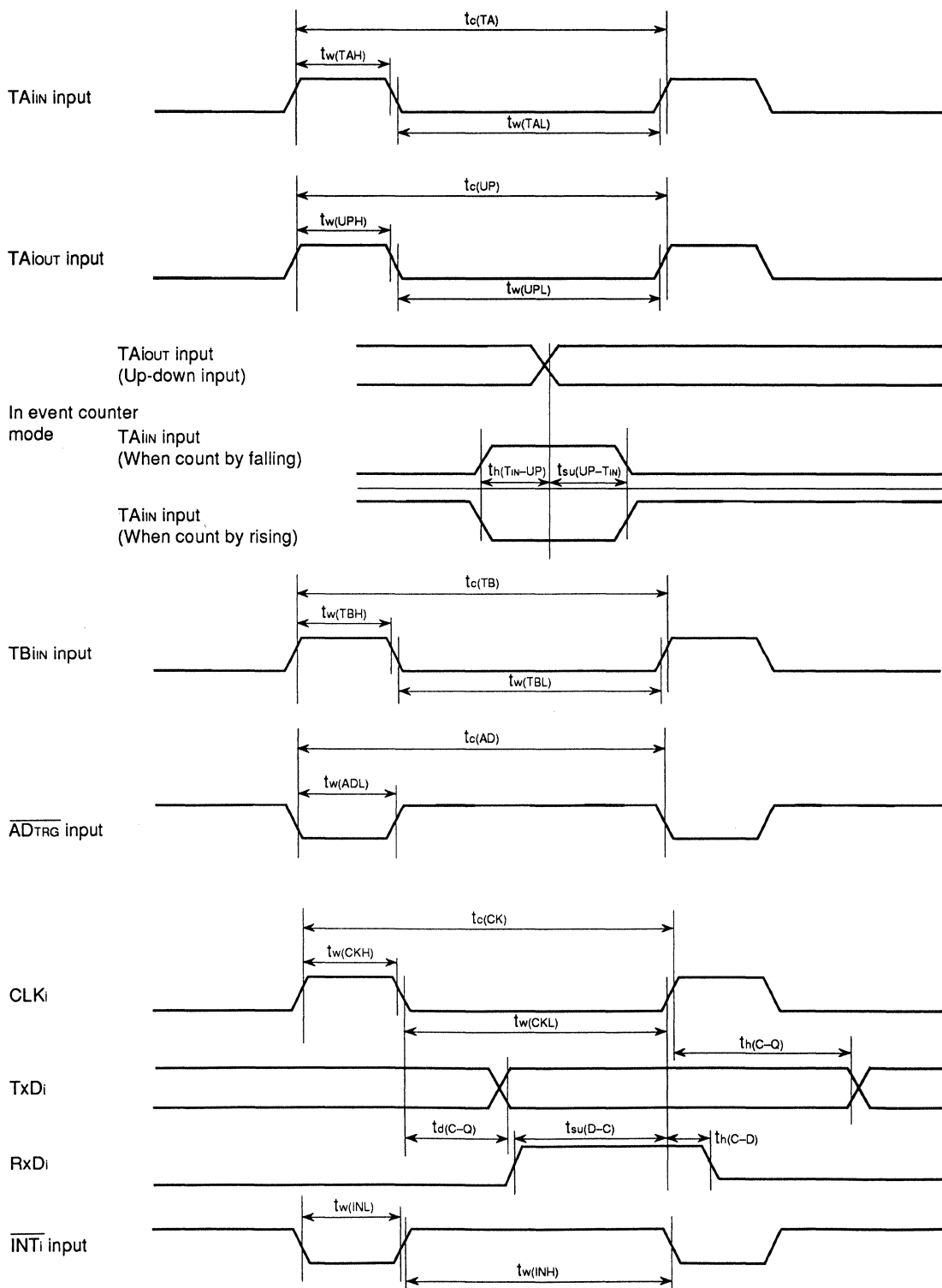
5.1.7 Timing diagram

Timing diagram



ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

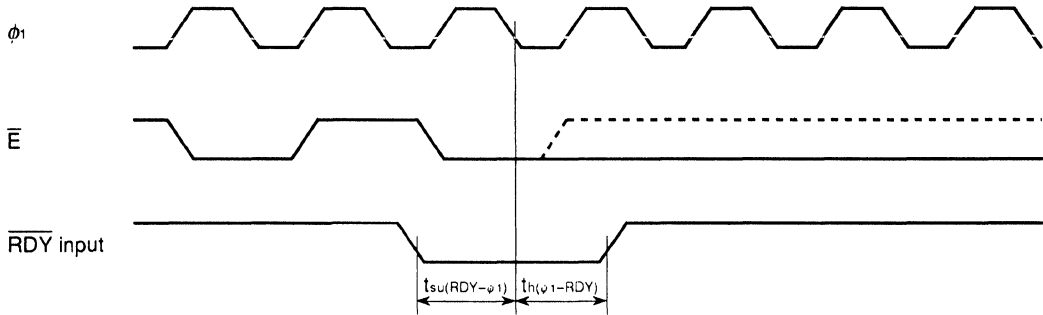


ELECTRICAL CHARACTERISTICS

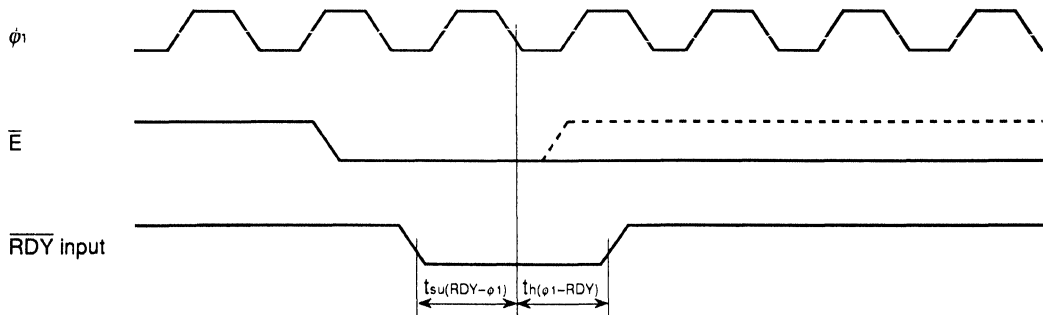
5.1 Electrical characteristics

Microprocessor mode

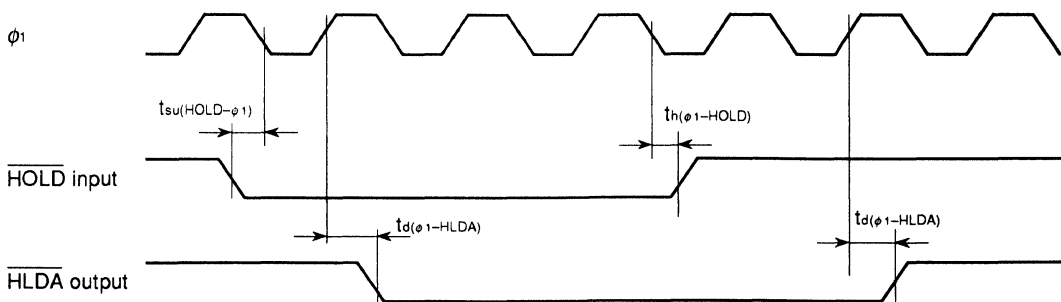
(When wait bit="1")



(When wait bit="0")



(When wait bit="1" or "0" in common)



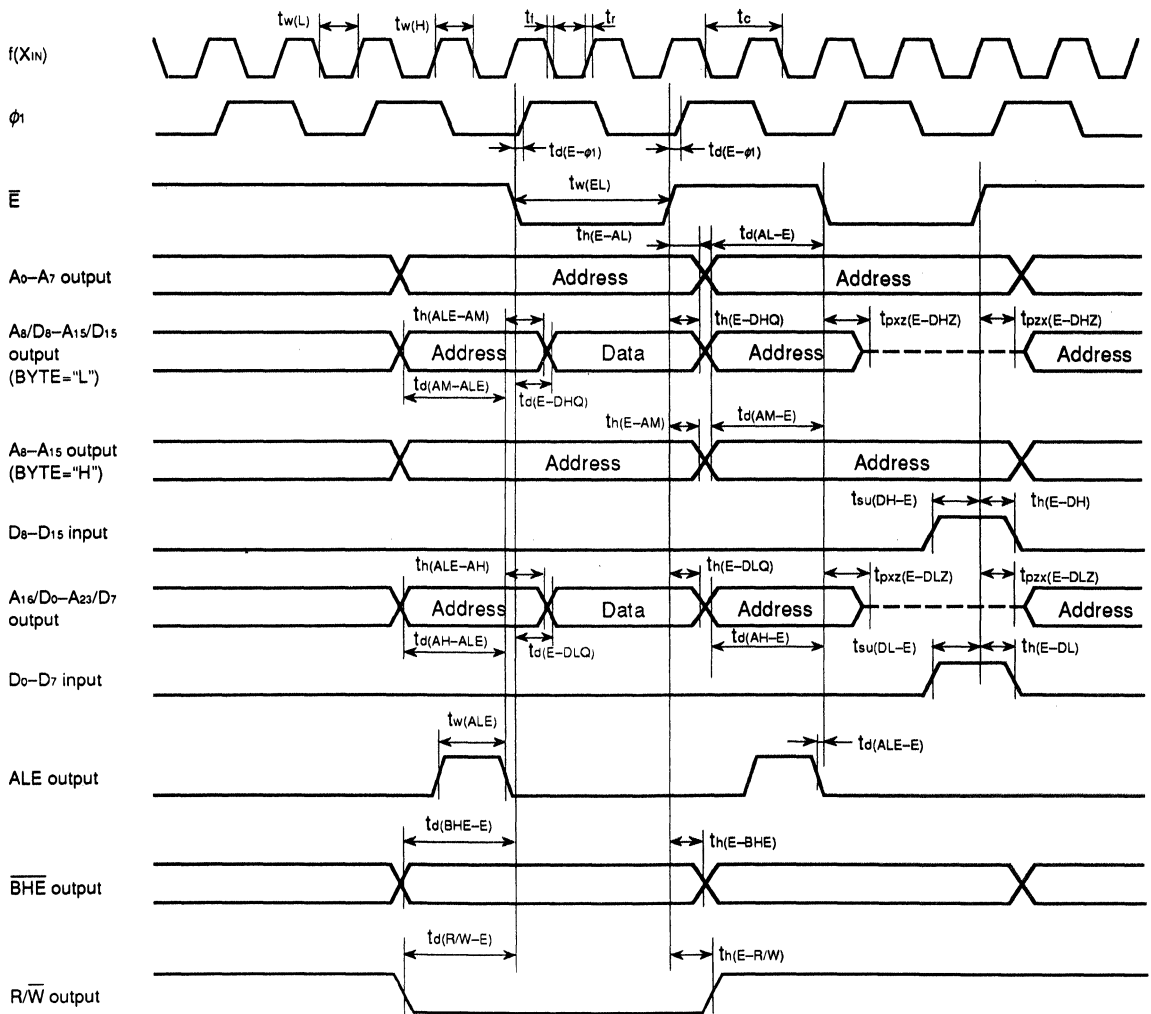
Test conditions

- $V_{CC}=5V \pm 10\%$
- Input timing voltage : $V_{IL}=1.0V$, $V_{IH}=4.0V$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

Microprocessor mode (When wait bit="1")



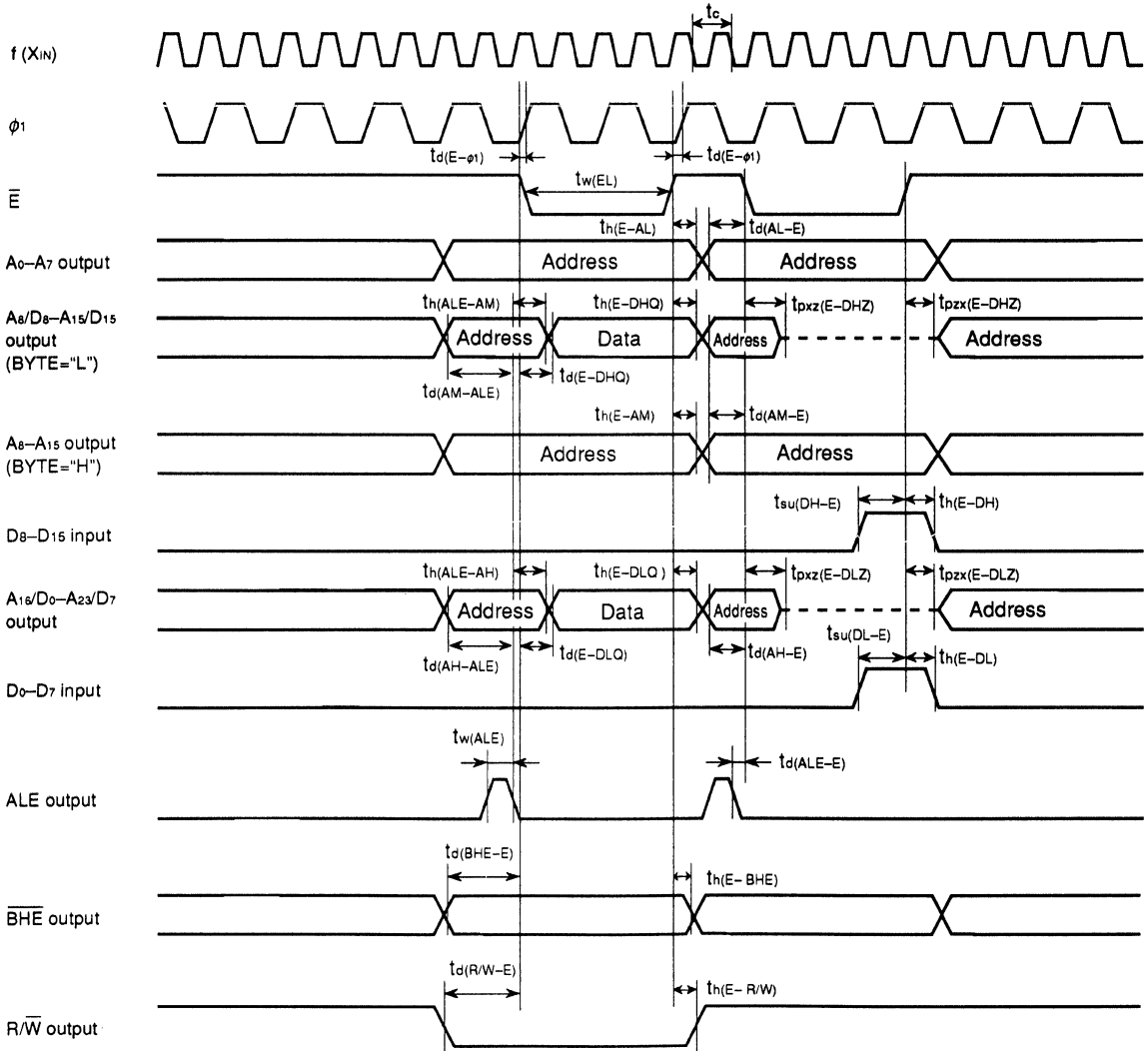
Test conditions

- $V_{CC}=5V \pm 10\%$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$
- D_0-D_{15} input : $V_{IL}=0.8V$, $V_{IH}=2.5V$

ELECTRICAL CHARACTERISTICS

5.1 Electrical characteristics

Microprocessor mode (When wait bit="0", and external memory area is accessed)



Test conditions

- $V_{CC}=5V \pm 10\%$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$
- D_0-D_{15} input : $V_{IL}=0.8V$, $V_{IH}=2.5V$

ELECTRICAL CHARACTERISTICS

MEMORANDUM



CHAPTER 6

STANDARD CHARACTERISTICS

6.1 Standard characteristics

STANDARD CHARACTERISTICS

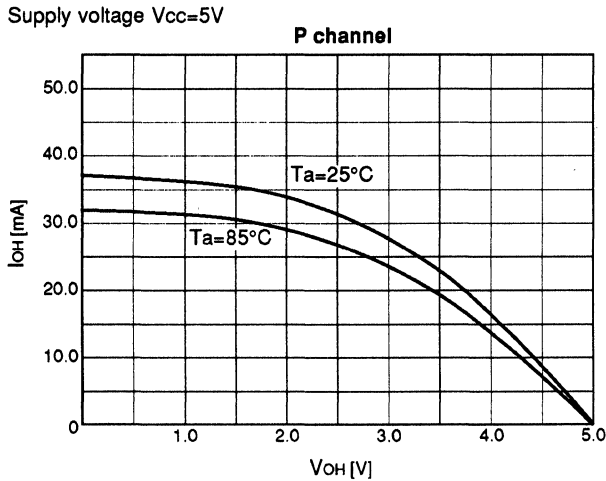
6.1 Standard characteristics

6.1 Standard characteristics

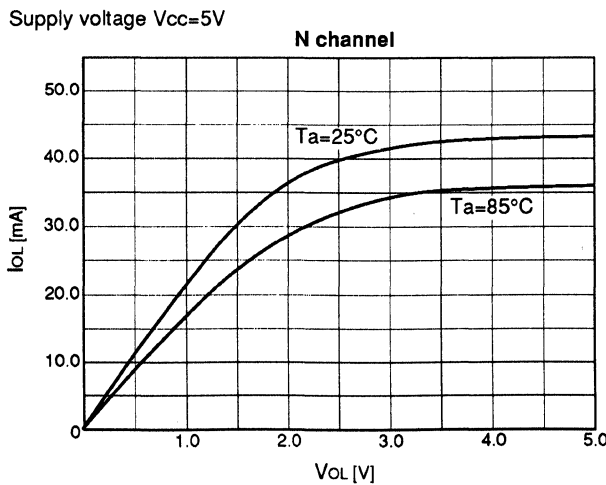
The data described in this chapter are characteristic examples for M37732S4BFP. The data is not guaranteed value. Refer to "Chapter 5. Electrical characteristics" for rated values.

6.1.1 Standard port characteristics

(1) Programmable I/O port (CMOS output) P channel $I_{OH}-V_{OH}$ characteristics



(2) Programmable I/O port (CMOS output) N channel $I_{OL}-V_{OL}$ characteristics



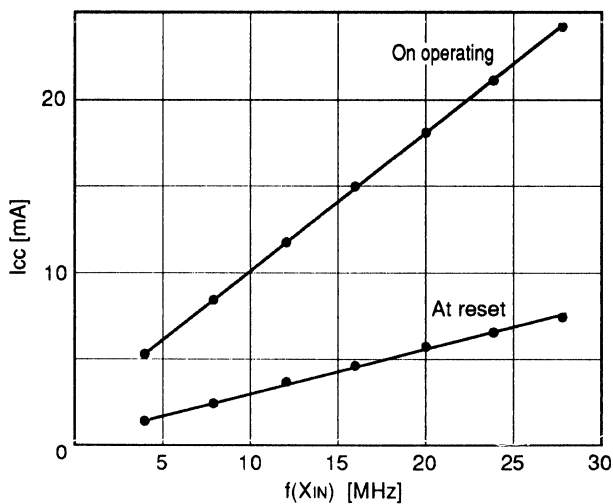
STANDARD CHARACTERISTICS

6.1 Standard characteristics

6.1.2 I_{CC} - $f(X_{IN})$ standard characteristics

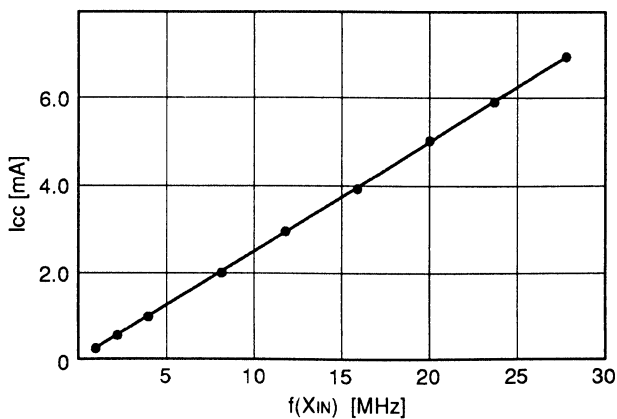
(1) I_{CC} - $f(X_{IN})$ characteristics on operating and at reset

Measurement condition ($V_{CC}=5V$, $T_a=25^\circ C$, $f(X_{IN})$: square wave input, microprocessor mode)



(2) I_{CC} - $f(X_{IN})$ characteristics during wait

Measurement condition ($V_{CC}=5V$, $T_a=25^\circ C$, $f(X_{IN})$: square wave input, microprocessor mode)



STANDARD CHARACTERISTICS

6.1 Standard characteristics

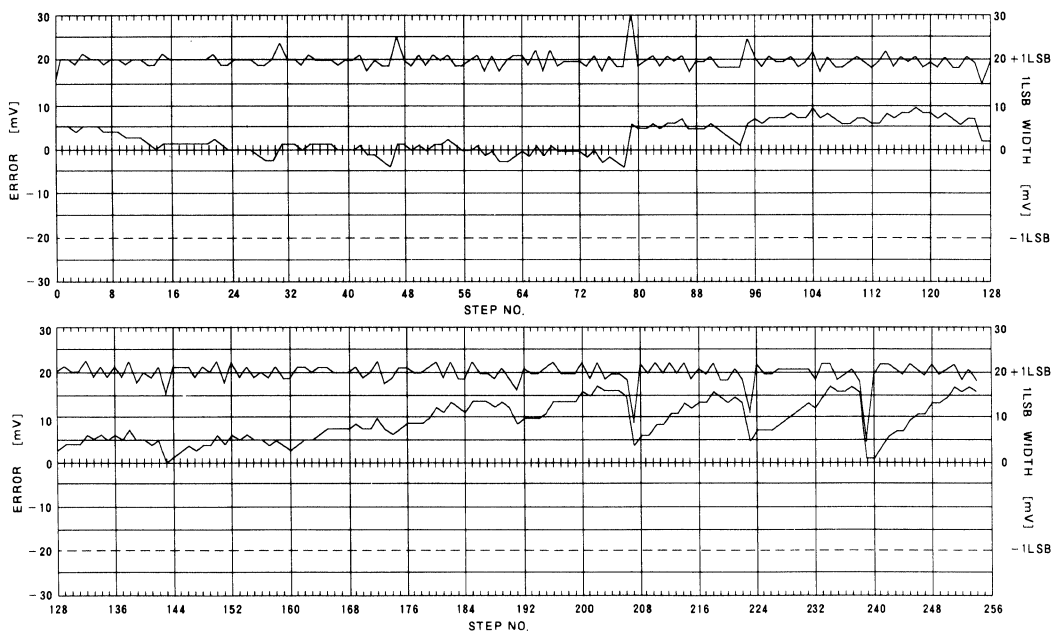
6.1.3 A-D converter standard characteristics

The lines at the bottom of the graph indicate the absolute precision errors. These are expressed as the deviation from the ideal value when the output code changes. For example, the change in output code from 00_{16} to 01_{16} should occur at $AN_i=10\text{mV}$, but the measured value is 5mV . Therefore, the measured point of change is $10+5=15\text{mV}$.

The lines at the top of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is $0F_{16}$ is 22mV . Therefore, the differential non-linear error is $22-20=2\text{mV}$ (0.1LSB).

[Measurement condition]

- $V_{CC} = 5.12\text{V}$
- $V_{REF} = 5.12\text{V}$
- $X_{IN} = 25\text{MHz}$
- $\text{Temp.} = 25^\circ\text{C}$





CHAPTER 7

LOW SUPPLY VOLTAGE VERSION

- 7.1 Description
- 7.2 Functional description
- 7.3 Electrical characteristics
- 7.4 Standard characteristics

LOW SUPPLY VOLTAGE VERSION

7.1 Description

7.1 Description

The M37732S4LGP and M37732S4LHP are low supply voltage versions of the M37732S4FP. In addition to lowering the supply voltage to 2.7–5.5V, they feature a wide operating temperature range between –40 and 85°C, and small packages with pin pitches of 0.65mm and 0.5mm.

These microprocessors are suitable for controlling battery driven portable equipment and small office equipment that must process large amounts of data using low power .

7.1.1 Low supply voltage version

The low supply voltage version of the M37732 group is available in two types as shown in Figure 7.1.1. These types feature low supply voltage, small package, and wide operating temperature range. Only the containing package is different between the two types. Therefore, the description in this chapter is focused on the M37732S4LGP unless noted otherwise.

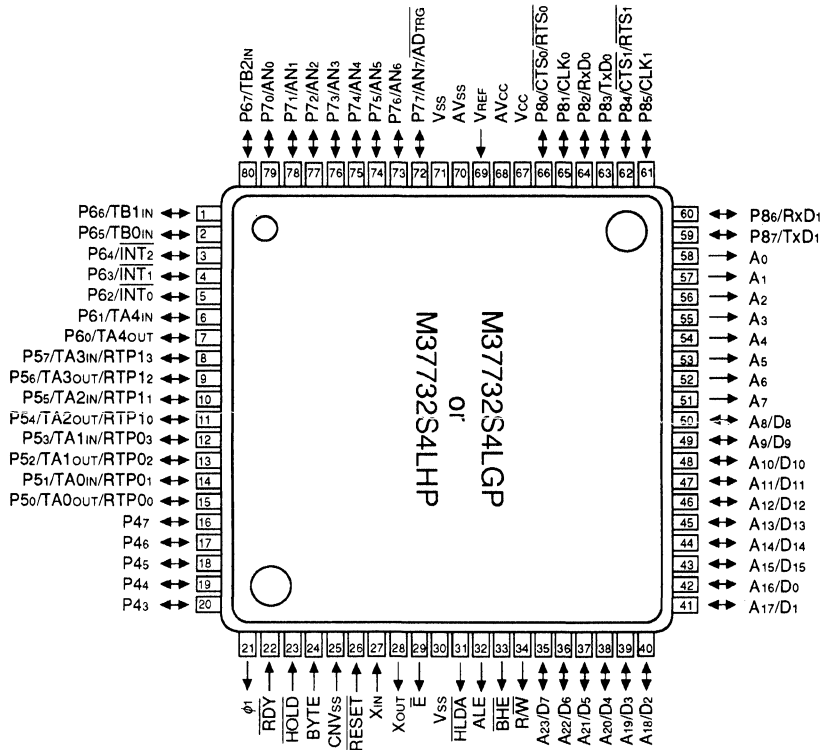
Figure 7.1.1 shows the pin configuration of the M37732S4LGP.

Table 7.1.1 Low supply voltage version

Type name	Supply voltage (V)	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)	Package
M37732S4LGP	2.7 to 5.5	—	2048	8	80-pin plastic molded QFP (80P6S–A)
M37732S4LHP					80-pin plastic molded fine-pitch QFP (80P6D–A)

LOW SUPPLY VOLTAGE VERSION

7.1 Description



Outline M37732S4LGP 80P6S-A

Outline M37732S4LHP 80P6D-A

* The pin allocation of M37732S4LGP and M37732S4LHP is shifted by two pins compared with that of M37732S4FFP.

Fig. 7.1.1 M37732S4LGP pin configuration (top view)

LOW SUPPLY VOLTAGE VERSION

7.1 Description

7.1.2 Performance overview

Table 7.1.2 shows the performance overview of the M37732S4LGP.

Table 7.1.2 M37732S4LGP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time		500ns (the fastest instruction at 8MHz frequency)
Operating clock frequency		8MHz (maximum)
Memory size	RAM	2048 bytes
Input/Output ports	Ports P5–P8	8 bits × 4
	Port P4	5 bits × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits × 5
	TB0, TB1, TB2	16 bits × 3
Serial I/O		(UART or clock synchronous serial I/O) × 2
A-D converter		8 bits × 1 (8 channels)
Watchdog timer		12 bits × 1
Interrupts		3 external, 16 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic or crystal resonator)
Supply voltage		2.7 to 5.5V
Power dissipation (Typ.)		12mW (at 3V supply voltage, 8MHz frequency)
		30mW (at 5V supply voltage, 8MHz frequency)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		16M bytes (maximum)
Operating temperature range		–40 to 85°C
Device structure		High-performance silicon gate CMOS process
Package	M37732S4LGP	80-pin plastic molded QFP (80P6S–A: lead pitch 0.65mm)
	M37732S4LHP	80-pin plastic molded fine-pitch QFP (80P6D–A: lead pitch 0.5mm)

LOW SUPPLY VOLTAGE VERSION

7.1 Description

7.1.3 Pin description

Table 7.1.3 shows the pin description.

Table 7.1.3 Pin description (1)

Pin	Name	Input/Output	Functions
V _{CC} , V _{SS}	Power supply		Supply 2.7–5.5V to V _{CC} and 0V to V _{SS} .
CNV _{SS}	CNV _{SS} input	Input	Connect to V _{CC} .
RESET	Reset input	Input	The microcomputer is reset when this pin is set to "L" level.
X _{IN}	Clock input	Input	These are the I/O pins of the internal clock generating circuit. Connect a ceramic or crystal resonator between X _{IN} and X _{OUT} . When an external clock is used, the clock source should be connected to the X _{IN} pin and the X _{OUT} pin should be left open.
X _{OUT}	Clock output	Output	
\bar{E}	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L" level.
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when the signal level is "L" and 8 bits when the signal level is "H".
AV _{CC} , AV _{SS}	Analog supply input		Power supply for the A-D converter. Externally connect AV _{CC} to V _{CC} and AV _{SS} to V _{SS} .
V _{REF}	Reference voltage input	Input	This is a reference voltage input pin for the A-D converter.
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X _{IN} pin by 2.
\overline{RDY}	Ready input	Input	The microcomputer is in Ready state while "L" level is input to this pin.
\overline{HOLD}	Hold input	Input	The microcomputer is in Hold state while "L" level is input to this pin.
HLDA	Hold acknowledge output	Output	This pin outputs "L" level when microcomputer is in Hold state.
R/W	Read/Write output	Output	The read/write signal indicates the data bus state. The data bus state is read when this signal is "H" level and write when "L" level.
\overline{BHE}	Byte high enable output	Output	"L" level is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to obtain only the address from address and data multiplex signals.
A ₀ –A ₇	Address bus low-order output	Output	Low-order 8 bits (A ₀ –A ₇) of the address bus are output. ●When the external bus width is 16 bits. (The BYTE pin is at "L" level.) High-order 8 bits (D ₈ –D ₁₅) of the data bus are input or output when the \bar{E} output is "L" level. Middle-order 8 bits (A ₈ –A ₁₅) of the address bus are output when \bar{E} output is "H" level. ●When the external bus width is 8 bits. (The BYTE pin is at "H" level.) Middle-order 8 bits (A ₈ –A ₁₅) of the address bus are output.
A ₈ /D ₈ –A ₁₅ /D ₁₅	Address bus middle-order output/ Data bus high-order I/O	I/O	
A ₁₆ /D ₀ –A ₂₃ /D ₇	Address bus high-order output/ Data bus low-order I/O	I/O	Low-order 8 bits (D ₀ –D ₇) of the data bus are input or output when the \bar{E} output is "L" level. High-order 8 bits (A ₁₆ –A ₂₃) of the address bus are output when the \bar{E} output is "H" level.

LOW SUPPLY VOLTAGE VERSION

7.1 Description

Table 7.1.3 Pin description (2)

Pin	Name	Input/Output	Functions
P4 ₃ –P4 ₇	I/O port P4	I/O	Port P4 is a 5-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output.
P5 ₀ –P5 ₇	I/O port P5	I/O	Port P5 is an 8-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output. These pins also function as I/O pins for timers A0–A3.
P6 ₀ –P6 ₇	I/O port P6	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as I/O pins for timer A4, external interrupt input pins for INT ₀ –INT ₂ , and input pins for timers B0–B2.
P7 ₀ –P7 ₇	I/O port P7	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as analog input pins AN ₀ –AN ₇ . P7 ₇ also functions as the AD _{TRG} input pin for an A-D conversion trigger.
P8 ₀ –P8 ₇	I/O port P8	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as CTS/RTS, CLK, RxD, TxD pins for UART0 and UART1.

LOW SUPPLY VOLTAGE VERSION

7.1 Description

7.1.4 Block diagram

Figure 7.1.2 shows the M37732S4LGP block diagram.

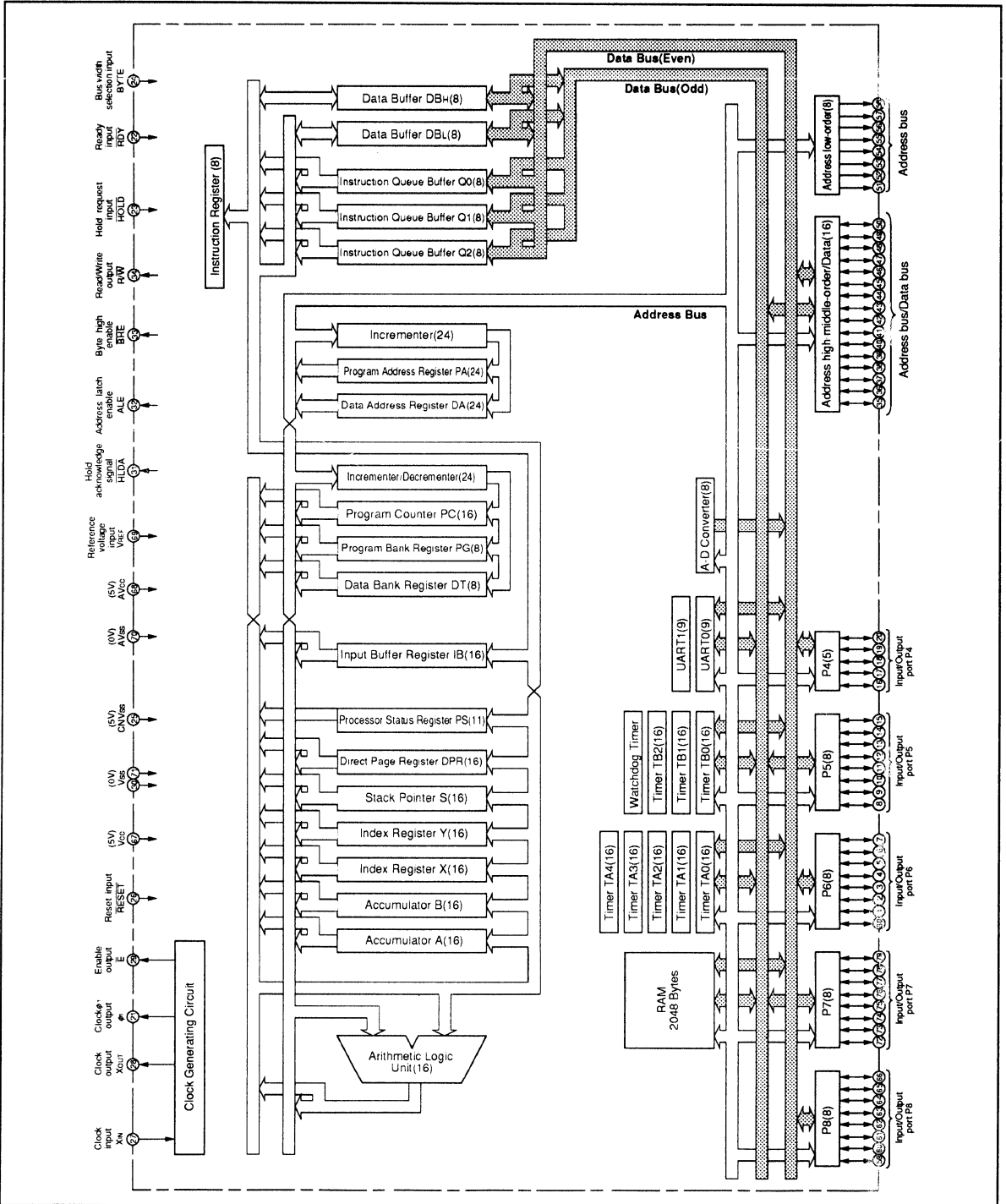


Fig. 7.1.2 M37732S4LGP block diagram

7.2 Functional description

The function of the M37732S4LGP is identical to the M37732S4FP. However, the reset voltage condition is different. Therefore, this section describes about reset. It also describes the oscillating circuit for low supply voltage operation.

For information concerning other functions, refer to "CHAPTER 2. FUNCTIONAL DESCRIPTION."

7.2.1 Reset

The CPU becomes reset state when "L" level is applied to the RESET pin while the supply voltage is between 2.7 and 5.5V. Only the reset voltage condition and the reset circuit are different from M37732S4FP. The reset operation, internal registers state at reset, and software reset of M37732S4LGP are identical to M37732S4FP. Therefore refer to "CHAPTER 3. RESET" for details.

The reset circuit is described below.

(1) Reset circuit

The reset circuit must be designed so that the reset input voltage is 0.55V or lower when the supply voltage reaches 2.7V as shown in Figure 7.2.1.

Figure 7.2.2 shows the example of a power-on reset circuit using the system reset IC M62003L/FP.

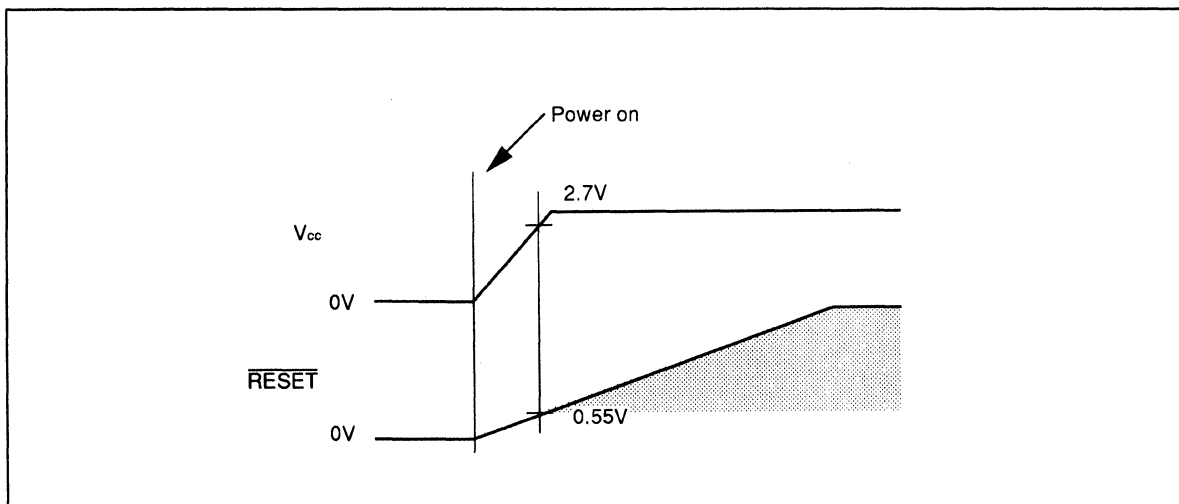
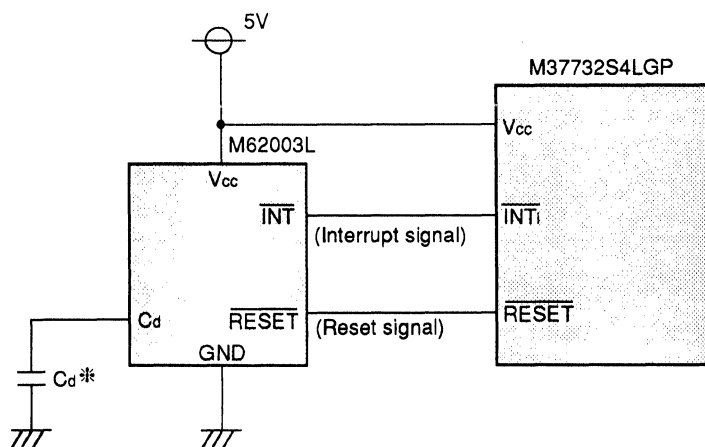


Fig. 7.2.1 Power-on reset condition

LOW SUPPLY VOLTAGE VERSION

7.2 Functional description



* In case of $C_d=0.07\mu\text{F}$, the delay time t_d is about 10ms.

$$t_d \approx 0.152C_d [\mu\text{s}], C_d: [\mu\text{F}]$$

Fig. 7.2.2 Power-on reset circuit example

LOW SUPPLY VOLTAGE VERSION

7.2 Functional description

7.2.2 Oscillating circuit

The M37732S4LGP has the same internal oscillating circuit as M37732S4FP. The frequency input to the clock input pin X_{IN} is divided in half to obtain the internal clock ϕ . This ϕ is further divided in half to obtain the bus cycle.

Either a ceramic resonator or a crystal resonator can be connected externally to the internal oscillating circuit. An external clock signal can be supplied to the internal oscillating circuit. The circuit examples using these resonators and external clock are identical to M37732S4FP.

Figure 7.2.3 shows the oscillating circuit example using a ceramic resonator.

The circuit constants such as R_f , R_d , C_{IN} , and C_{OUT} must be set to the resonator manufacturer's recommended values.

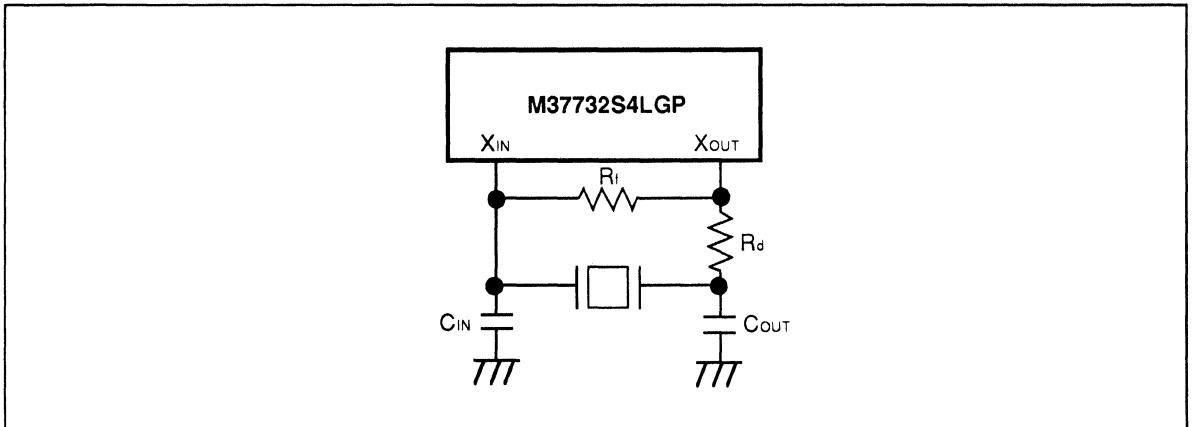


Fig. 7.2.3 Oscillating circuit example using a ceramic resonator

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

7.3 Electrical characteristics

7.3.1 Absolute maximum ratings

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V _{cc}	Supply voltage		-0.3 to 7	V
AV _{cc}	Analog supply voltage		-0.3 to 7	V
V _i	Input voltage RESET, CNV _{SS} , BYTE		-0.3 to 12	V
V _i	Input voltage A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₇₀ -P ₇₇ , P ₈₀ -P ₈₇ , V _{REF} , X _{IN} , HOLD, RDY		-0.3 to V _{cc} +0.3	V
V _o	Output voltage A ₀ -A ₇ , A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₇₀ -P ₇₇ , P ₈₀ -P ₈₇ , X _{OUT} , E, ϕ_1 , HLDA, ALE, BHE, R/W		-0.3 to V _{cc} +0.3	V
P _d	Power dissipation	T _a =25°C	300 (Note)	mW
T _{opr}	Operating temperature		-40 to 85	°C
T _{stg}	Storage temperature		-65 to 150	°C

Note: In case of M37732S4LHP, the rating of power dissipation is 200mW.

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

7.3.2 Recommended operating conditions

Recommended operating conditions ($V_{CC}=2.7$ to $5.5V$, $T_a=-40$ to $85^\circ C$, unless otherwise noted)

Symbol	Parameter	Limits			Unit	
		Min.	Typ.	Max.		
V_{CC}	Supply voltage	2.7		5.5	V	
AV_{CC}	Analog supply voltage		V_{CC}		V	
V_{SS}	Supply voltage		0		V	
AV_{SS}	Analog supply voltage		0		V	
V_{IH}	High-level input voltage	$P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, X_{IN}, RESET, CNV_{SS}, BYTE, HOLD, RDY$	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage	$A_8/D_8-A_{23}/D_7$	$0.5V_{CC}$		V_{CC}	V
V_{IL}	Low-level input voltage	$P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, X_{IN}, RESET, CNV_{SS}, BYTE, \overline{HOLD}, RDY$	0		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage	$A_8/D_8-A_{23}/D_7$	0		$0.16V_{CC}$	V
$I_{OH}(\text{peak})$	High-level peak output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, \phi_1, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$			-10	mA
$I_{OH}(\text{avg})$	High-level average output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, \phi_1, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$			-5	mA
$I_{OL}(\text{peak})$	Low-level peak output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, \phi_1, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$			10	mA
$I_{OL}(\text{avg})$	Low-level average output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P7_0-P7_7, P8_0-P8_7, \phi_1, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$			5	mA
$f(X_{IN})$	External clock frequency input				8	MHz

Note 1: Average output current is the average value of a 100ms interval.

2: The sum of $I_{OL}(\text{peak})$ for $A_0-A_7, A_8/D_8-A_{23}/D_7, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$, and port P8 must be 80mA or less, the sum of $I_{OH}(\text{peak})$ for $A_0-A_7, A_8/D_8-A_{23}/D_7, \overline{HLDA}, ALE, \overline{BHE}, R/\overline{W}$, and port P8 must be 80mA or less, the sum of $I_{OL}(\text{peak})$ for ports P4, P5, P6, P7, and ϕ_1 must be 80mA or less, and the sum of $I_{OH}(\text{peak})$ for ports P4, P5, P6, P7, and ϕ_1 must be 80mA or less.

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

7.3.3 Electrical characteristics and A-D converter characteristics

Electrical characteristics (Ta=25°C, f(XIN)=8MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit			
			Min.	Typ.	Max.				
VOH	High-level output voltage	A0-A7, A8/D8-A23/D7, P43-P47, P50-P57, P60-P67, P70-P77, P80-P87, φ1, HLDA, BHE, R/W	VCC=5V, IOH=-10mA	3			V		
			VCC=3V, IOH=-1mA	2.5					
VOH	High-level output voltage	A0-A7, A8/D8-A23/D7, φ1, HLDA, BHE, R/W	VCC=5V, IOH=-400μA	4.7			V		
VOH	High-level output voltage	ALE	VCC=5V, IOH=-10mA	3.1			V		
			VCC=5V, IOH=-400μA	4.8					
			VCC=3V, IOH=-1mA	2.6					
VOH	High-level output voltage	E	VCC=5V, IOH=-10mA	3.4			V		
			VCC=5V, IOH=-400μA	4.8					
			VCC=3V, IOH=-1mA	2.6					
VOL	Low-level output voltage	A0-A7, A8/D8-A23/D7, P43-P47, P50-P57, P60-P67, P70-P77, P80-P87, φ1, HLDA, BHE, R/W	VCC=5V, IOL=10mA			2	V		
				VCC=3V, IOL=1mA				0.5	
VOL	Low-level output voltage	A0-A7, A8/D8-A23/D7, φ1, HLDA, BHE, R/W	VCC=5V, IOL=2mA			0.45	V		
VOL	Low-level output voltage	ALE	VCC=5V, IOL=10mA			1.9	V		
			VCC=5V, IOL=2mA			0.43			
			VCC=3V, IOL=1mA			0.4			
VOL	Low-level output voltage	E	VCC=5V, IOL=10mA			1.6	V		
			VCC=5V, IOL=2mA			0.4			
			VCC=3V, IOL=1mA			0.4			
VT+ - VT-	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB2IN, INT0-INT2, ADTRG, CTS0, CTS1, CLK0, CLK1	VCC=5V	0.4		1	V		
				VCC=3V	0.1			0.7	
VT+ - VT-	Hysteresis	RESET	VCC=5V	0.2		0.5	V		
				VCC=3V	0.1			0.4	
VT+ - VT-	Hysteresis	XIN	VCC=5V	0.1		0.3	V		
				VCC=3V	0.06			0.2	
IIH	High-level input current	A8/D8-A23/D7, P43-P47, P50-P57, P60-P67, P70-P77, P80-P87, XIN, RESET, CNVSS, BYTE, HOLD, RDY	VCC=5V, VI=5V			5	μA		
				VCC=3V, VI=3V				4	
IIL	Low-level input current	A8/D8-A23/D7, P43-P47, P50-P57, P60-P67, P70-P77, P80-P87, XIN, RESET, CNVSS, BYTE, HOLD, RDY	VCC=5V, VI=0V			-5	μA		
				VCC=3V, VI=0V				-4	
V _{RAM}	RAM hold voltage		When clock is stopped.		2		V		
I _{CC}	Power supply current		Output only pin is open and other pins are V _{SS} during reset.	f(XIN)=8MHz, square waveform	VCC=5V	6	12	mA	
					VCC=3V	4	8		
					Ta=25°C when clock is stopped.			1	μA
					Ta=85°C when clock is stopped.			20	

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

A-D converter characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	$V_{REF}=V_{CC}$			8	Bits
—	Absolute accuracy	$V_{REF}=V_{CC}$			± 3	LSB
R_{LADDER}	Ladder resistance	$V_{REF}=V_{CC}$	2		10	$k\Omega$
t_{CONV}	Conversion time		28.5			μs
V_{REF}	Reference voltage		2.7		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

7.3.4 Timing requirements

Timing requirements ($V_{CC}=2.7$ to $5.5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$, unless otherwise noted)

External clock input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	125		ns
$t_{w(H)}$	External clock input high-level pulse width	50		ns
$t_{w(L)}$	External clock input low-level pulse width	50		ns
t_r	External clock rise time		20	ns
t_f	External clock fall time		20	ns

Microprocessor mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(DH-E)}$	Data high-order input setup time	80		ns
$t_{su(DL-E)}$	Data low-order input setup time	80		ns
$t_{su(P4D-E)}$	Port P4 input setup time	300		ns
$t_{su(P5D-E)}$	Port P5 input setup time	300		ns
$t_{su(P6D-E)}$	Port P6 input setup time	300		ns
$t_{su(P7D-E)}$	Port P7 input setup time	300		ns
$t_{su(P8D-E)}$	Port P8 input setup time	300		ns
$t_{su(RDY-\phi)}$	RDY input setup time	90		ns
$t_{su(HOLD-\phi)}$	HOLD input setup time	90		ns
$t_h(E-DH)$	Data high-order input hold time	0		ns
$t_h(E-DL)$	Data low-order input hold time	0		ns
$t_h(E-P4D)$	Port P4 input hold time	0		ns
$t_h(E-P5D)$	Port P5 input hold time	0		ns
$t_h(E-P6D)$	Port P6 input hold time	0		ns
$t_h(E-P7D)$	Port P7 input hold time	0		ns
$t_h(E-P8D)$	Port P8 input hold time	0		ns
$t_h(\phi-RDY)$	RDY input hold time	0		ns
$t_h(\phi-HOLD)$	HOLD input hold time	0		ns

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	250		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	125		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	125		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	1000		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	500		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	500		ns

Timer A input (external trigger input in one-shot pulse mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	500		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		ns

Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		ns

Timer A input (up-down input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TA _{IOUT} input cycle time	5000		ns
$t_{w(UPH)}$	TA _{IOUT} input high-level pulse width	2500		ns
$t_{w(UPL)}$	TA _{IOUT} input low-level pulse width	2500		ns
$t_{su(UP-TIN)}$	TA _{IOUT} input setup time	1000		ns
$t_{h(TIN-UP)}$	TA _{IOUT} input hold time	1000		ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TB _{IN} input cycle time (one edge count)	250		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width (one edge count)	125		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width (one edge count)	125		ns
$t_{c(TB)}$	TB _{IN} input cycle time (both edges count)	500		ns
$t_{w(TBH)}$	TB _{IN} input high-level pulse width (both edges count)	250		ns
$t_{w(TBL)}$	TB _{IN} input low-level pulse width (both edges count)	250		ns

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TB _{iN} input cycle time	1000		ns
$t_{w(TBH)}$	TB _{iN} input high-level pulse width	500		ns
$t_{w(TBL)}$	TB _{iN} input low-level pulse width	500		ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TB _{iN} input cycle time	1000		ns
$t_{w(TBH)}$	TB _{iN} input high-level pulse width	500		ns
$t_{w(TBL)}$	TB _{iN} input low-level pulse width	500		ns

A-D trigger input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	AD _{TRG} input cycle time (minimum allowable trigger)	2000		ns
$t_{w(ADL)}$	AD _{TRG} input low-level pulse width	250		ns

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK _i input cycle time	500		ns
$t_{w(CKH)}$	CLK _i input high-level pulse width	250		ns
$t_{w(CKL)}$	CLK _i input low-level pulse width	250		ns
$t_{d(C-Q)}$	TxD _i output delay time		170	ns
$t_h(C-Q)$	TxD _i hold time	30		ns
$t_{su(D-C)}$	RxD _i input setup time	80		ns
$t_h(C-D)$	RxD _i input hold time	100		ns

External interrupt INT_i input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	INT _i input high-level pulse width	250		ns
$t_{w(INL)}$	INT _i input low-level pulse width	250		ns

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

7.3.5 Switching characteristics

Switching characteristics ($V_{CC}=2.7$ to $5.5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$ unless otherwise noted)

Microprocessor mode (when wait bit is "1")

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 7.3.1	50		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			130	ns
$t_{pXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			10	ns
$t_{d(AM-E)}$	Address middle-order output delay time		50		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		40		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			130	ns
$t_{pXZ(E-DLZ)}$	Floating start delay time			10	ns
$t_{d(AH-E)}$	Address high-order output delay time		50		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		40		ns
$t_{d(\phi-HLDA)}$	HLDA output delay time			120	ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_w(ALE)$	ALE pulse width		60		ns
$t_{d(BHE-E)}$	BHE output delay time		50		ns
$t_{d(R/W-E)}$	R/W output delay time		50		ns
$t_{d(E-\phi)}$	ϕ : output delay time		0	40	ns
$t_h(E-AL)$	Address low-order hold time		50		ns
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")		9		ns
$t_h(E-DHQ)$	Data high-order hold time (BYTE="L")		50		ns
$t_{pXZ(E-DHZ)}$	Floating release delay time (BYTE="L")		95		ns
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")		50		ns
$t_h(ALE-AH)$	Address high-order hold time		9		ns
$t_h(E-DLQ)$	Data low-order hold time		50		ns
$t_{pXZ(E-DLZ)}$	Floating release delay time		95		ns
$t_h(E-BHE)$	BHE hold time		18		ns
$t_h(E-R/W)$	R/W hold time		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			300	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			300	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			300	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			300	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			300	ns
$t_w(EL)$	\bar{E} pulse width		210		ns

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

Microprocessor mode (when wait bit = "0", and external memory area is accessed)

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_d(\text{AL-E})$	Address low-order output delay time	Fig. 7.3.1	50		ns
$t_d(\text{E-DHQ})$	Data high-order output delay time (BYTE="L")			130	ns
$t_{pzx}(\text{E-DHZ})$	Floating start delay time (BYTE="L")			10	ns
$t_d(\text{AM-E})$	Address middle-order output delay time		50		ns
$t_d(\text{AM-ALE})$	Address middle-order output delay time		40		ns
$t_d(\text{E-DLQ})$	Data low-order output delay time			130	ns
$t_{pzx}(\text{E-DLZ})$	Floating start delay time			10	ns
$t_d(\text{AH-E})$	Address high-order output delay time		50		ns
$t_d(\text{AH-ALE})$	Address high-order output delay time		40		ns
$t_d(\phi_1\text{-HLDA})$	HLDA output delay time			120	ns
$t_d(\text{ALE-E})$	ALE output delay time		4		ns
$t_w(\text{ALE})$	ALE pulse width		60		ns
$t_d(\text{BHE-E})$	BHE output delay time		50		ns
$t_d(\text{R/W-E})$	R/W output delay time		50		ns
$t_d(\text{E-}\phi_1)$	ϕ_1 output delay time		0	40	ns
$t_h(\text{E-AL})$	Address low-order hold time		50		ns
$t_h(\text{ALE-AM})$	Address middle-order hold time (BYTE="L")		9		ns
$t_h(\text{E-DHQ})$	Data high-order hold time (BYTE="L")		50		ns
$t_{pzx}(\text{E-DHZ})$	Floating release delay time (BYTE="L")		95		ns
$t_h(\text{E-AM})$	Address middle-order hold time (BYTE="H")		50		ns
$t_h(\text{ALE-AH})$	Address high-order hold time		9		ns
$t_h(\text{E-DLQ})$	Data low-order hold time		50		ns
$t_{pzx}(\text{E-DLZ})$	Floating release delay time		95		ns
$t_h(\text{E-BHE})$	BHE hold time		18		ns
$t_h(\text{E-R/W})$	R/W hold time		18		ns
$t_d(\text{E-P4Q})$	Port P4 data output delay time			300	ns
$t_d(\text{E-P5Q})$	Port P5 data output delay time			300	ns
$t_d(\text{E-P6Q})$	Port P6 data output delay time			300	ns
$t_d(\text{E-P7Q})$	Port P7 data output delay time			300	ns
$t_d(\text{E-P8Q})$	Port P8 data output delay time			300	ns
$t_w(\text{EL})$	\bar{E} pulse width		460		ns

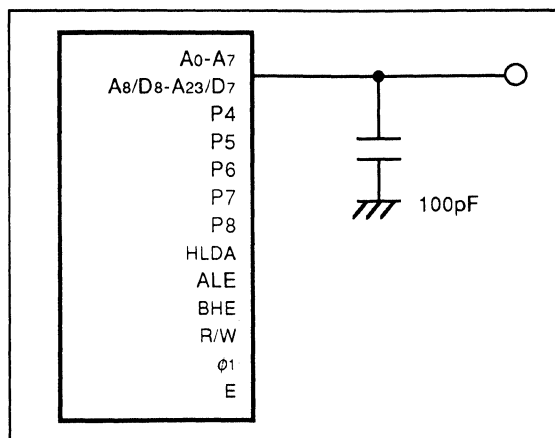


Fig. 7.3.1 Testing circuit for ports

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

7.3.6 Equations for calculating the parameters

Table 7.3.1 shows the equations for calculating the following parameters.

$t_w(\text{ALE})$ALE pulse width	$t_d(\text{R}/\overline{\text{W}}-\text{E})$ $\text{R}/\overline{\text{W}}$ output delay time
$t_w(\text{EL})$ $\overline{\text{E}}$ pulse width	$t_{\text{pzx}}(\text{E}-\text{DLZ}/\text{DHZ})$Floating release delay time
$t_d(\text{AL}/\text{AM}/\text{AH}-\text{E})$Address output delay time	$t_h(\text{E}-\text{AL}/\text{AM})$Address hold time
$t_d(\text{AM}/\text{AH}-\text{ALE})$Address output delay time	$t_h(\text{E}-\text{DLQ}/\text{DHQ})$Data hold time
$t_d(\text{BHE}-\text{E})$ $\overline{\text{BHE}}$ output delay time	

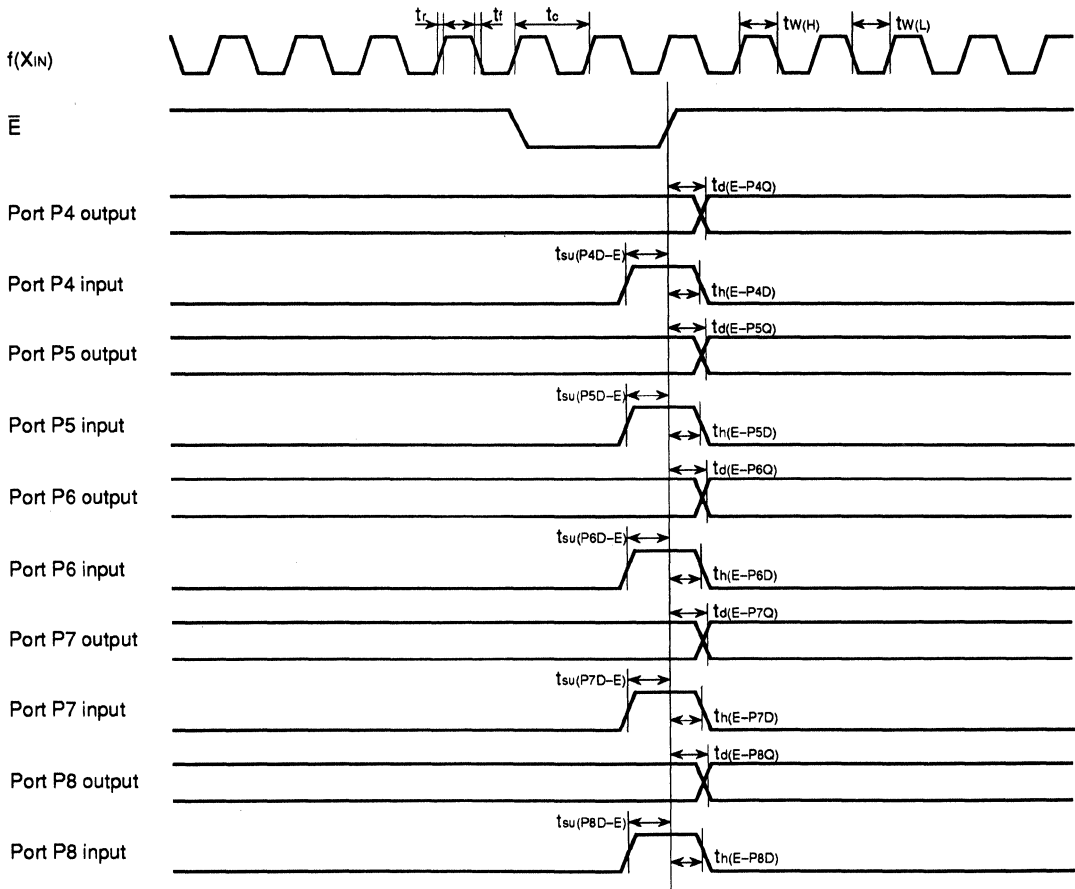
Table 7.3.1 Equations for calculating parameters

	M37732S4LGP, M37732S4LHP
$f(\text{XIN})$	$f(\text{XIN}) \leq 8\text{MHz}$
$t_w(\text{ALE})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 65$
$t_w(\text{EL})$ Wait bit = "1"	$\frac{2 \times 10^9}{f(\text{XIN})} - 40$
$t_w(\text{EL})$ Wait bit = "0"	$\frac{4 \times 10^9}{f(\text{XIN})} - 40$
$t_d(\text{AL}/\text{AM}/\text{AH}-\text{E})$	$50 + \frac{1 \times 10^9}{f(\text{XIN})} - 125$
$t_d(\text{BHE}-\text{E})$ $t_d(\text{R}/\overline{\text{W}}-\text{E})$	
$t_{\text{pzx}}(\text{E}-\text{DLZ}/\text{DHZ})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 30$
$t_h(\text{E}-\text{AL}/\text{AM})$ $t_h(\text{E}-\text{DLQ}/\text{DHQ})$	$\frac{1 \times 10^9}{2 \times f(\text{XIN})} - 12.5$
$t_d(\text{AM}/\text{AH}-\text{ALE})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 85$

Unit : ns

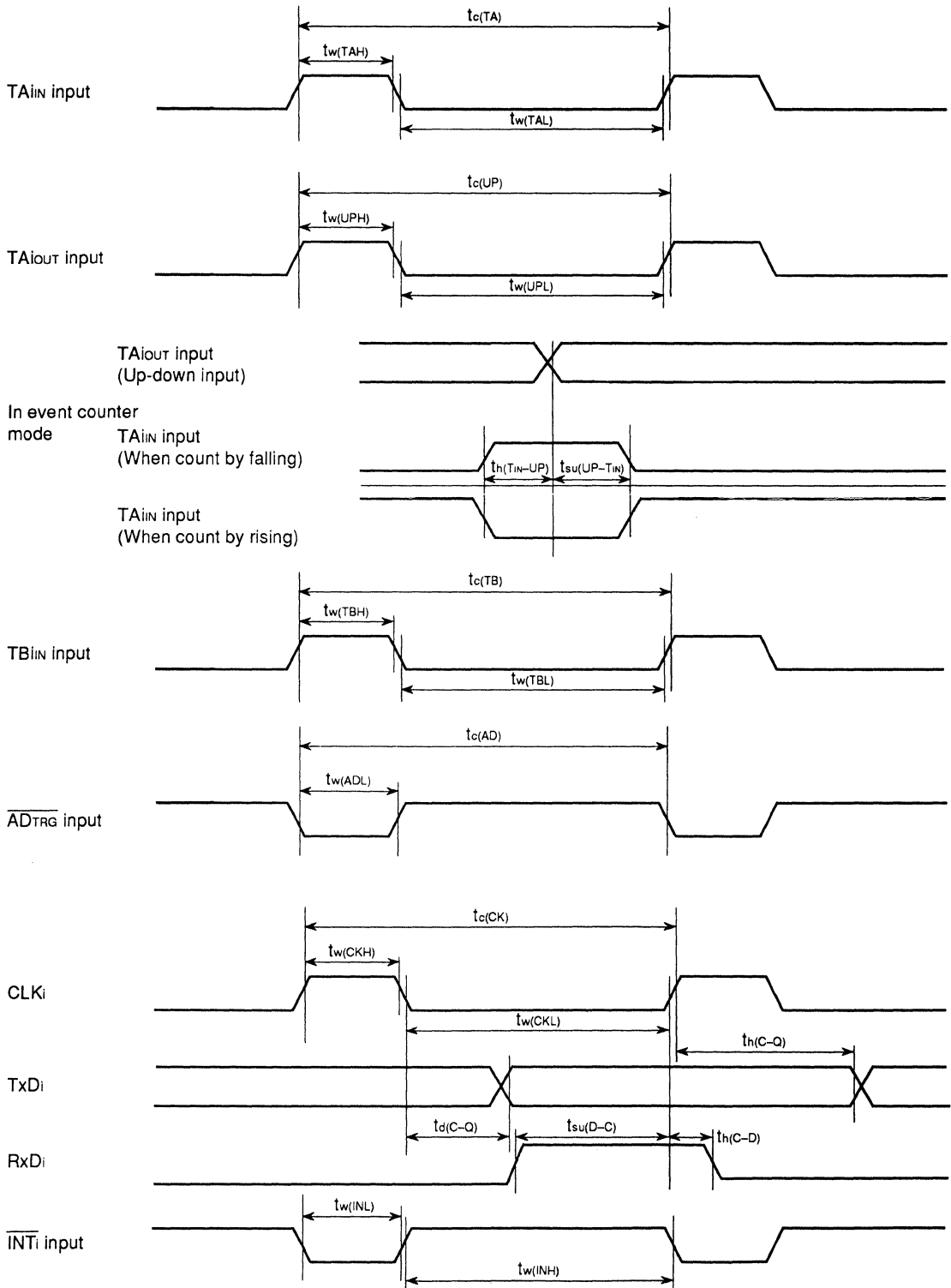
7.3.7 Timing diagram

Timing diagram



LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

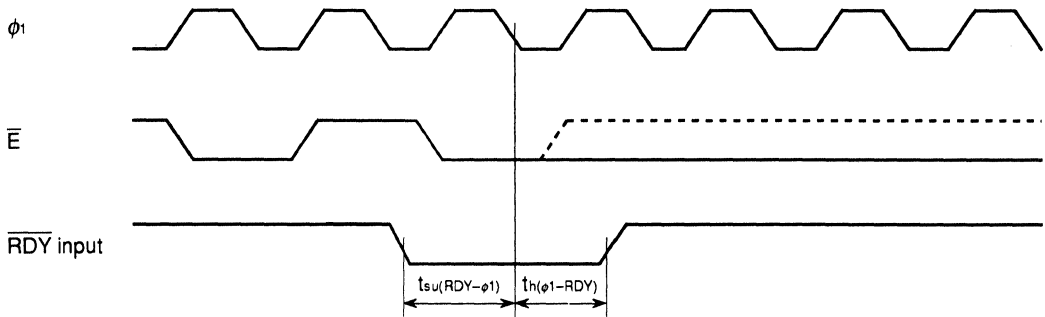


LOW SUPPLY VOLTAGE VERSION

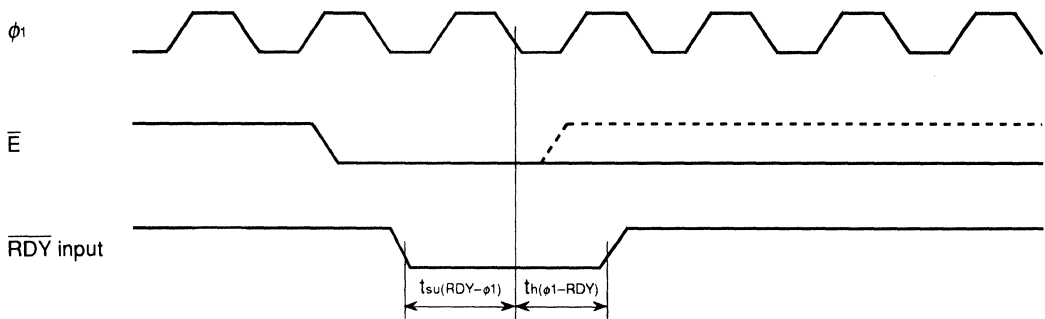
7.3 Electrical characteristics

Microprocessor mode

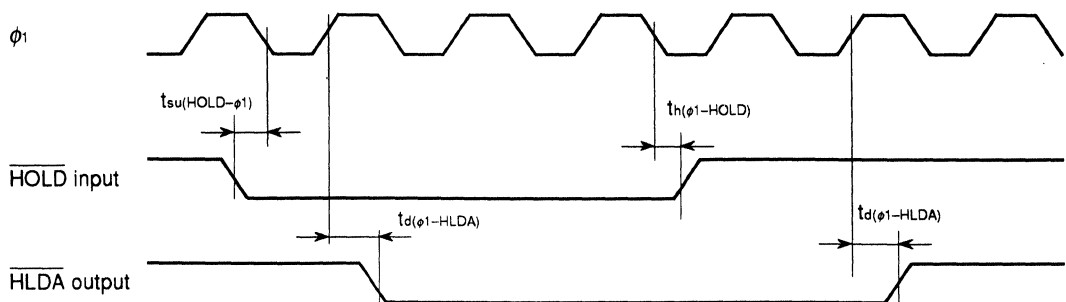
(When wait bit="1")



(When wait bit="0")



(When wait bit="1" or "0" in common)



Test conditions

• $V_{CC}=2.7$ to $5.5V$

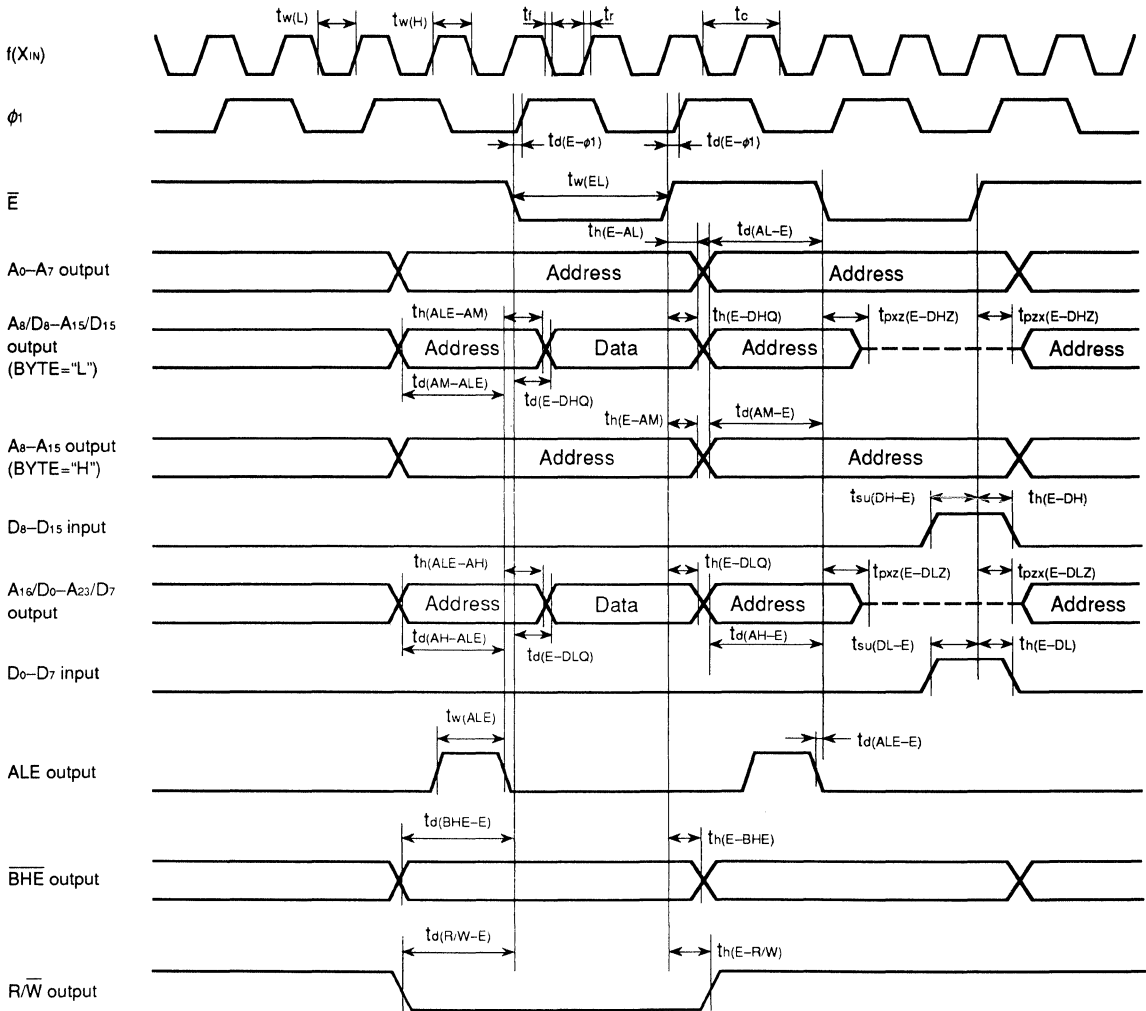
• Input timing voltage : $V_{IL}=0.2V_{CC}$, $V_{IH}=0.8V_{CC}$

• Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

Microprocessor mode (When wait bit="1")



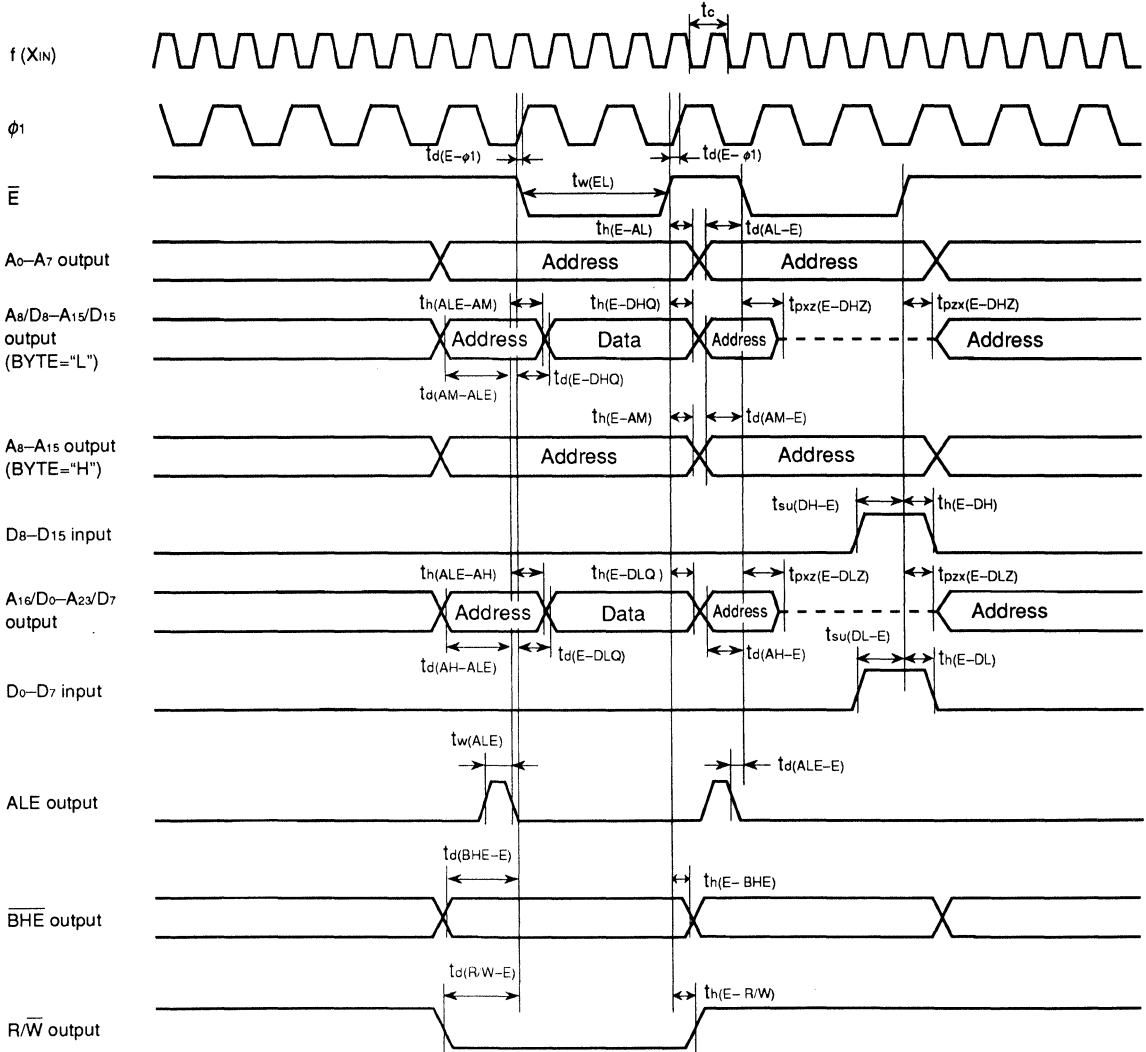
Test conditions

- $V_{CC}=2.7$ to $5.5V$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$
- D_0-D_{15} input : $V_{IL}=0.16V_{CC}$, $V_{IH}=0.5V_{CC}$

LOW SUPPLY VOLTAGE VERSION

7.3 Electrical characteristics

Microprocessor mode (When wait bit="0", and external memory area is accessed)



Test conditions

- $V_{CC}=2.7$ to $5.5V$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$
- D_0-D_{15} input : $V_{IL}=0.16V_{CC}$, $V_{IH}=0.5V_{CC}$

LOW SUPPLY VOLTAGE VERSION

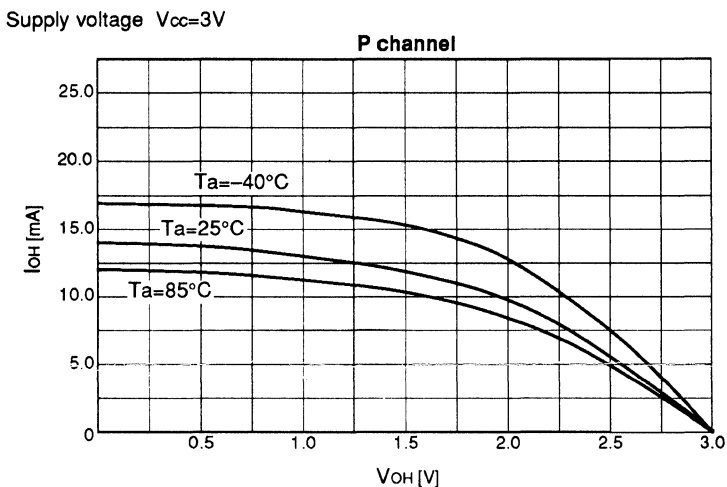
7.4 Standard characteristics

7.4 Standard characteristics

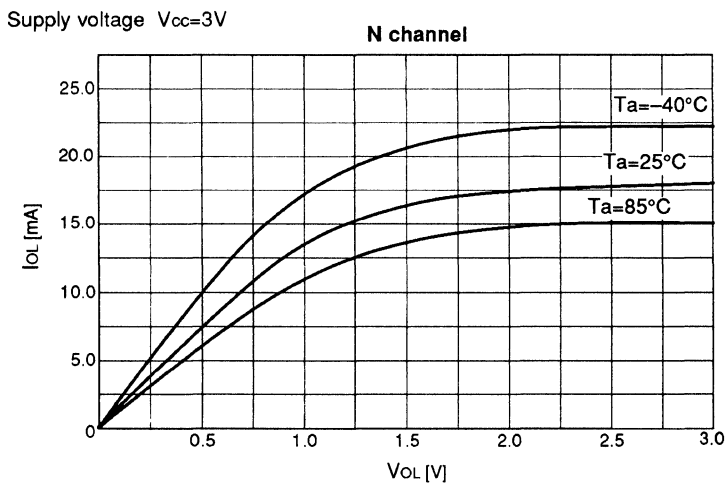
The data described in this section are characteristic examples for M37732S4LHP. The data is not guaranteed value. Refer to section "7.3 Electrical characteristics" for rated values.

7.4.1 Standard port characteristics

(1) Programmable I/O port (CMOS output) P channel $I_{OH}-V_{OH}$ characteristics



(2) Programmable I/O port (CMOS output) N channel $I_{OL}-V_{OL}$ characteristics



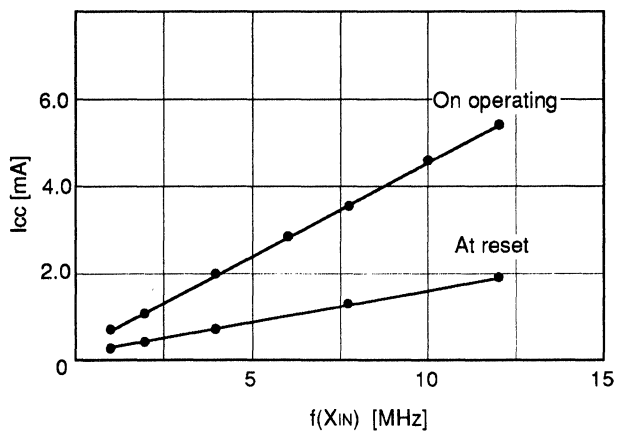
LOW SUPPLY VOLTAGE VERSION

7.4 Standard characteristics

7.4.2 $I_{CC}-f(X_{IN})$ standard characteristics

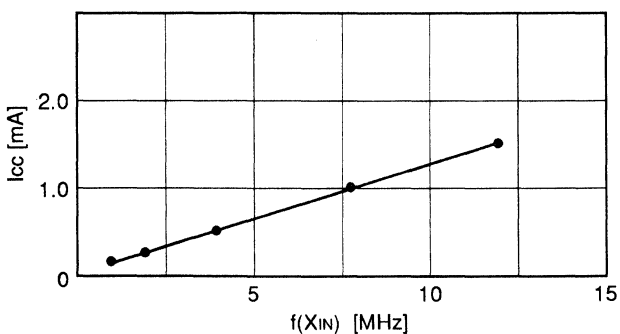
(1) $I_{CC}-f(X_{IN})$ characteristics on operating and at reset

Measurement condition ($V_{CC}=3V, T_a=25^\circ C, f(X_{IN})$: square wave input, microprocessor mode)



(2) $I_{CC}-f(X_{IN})$ characteristics during wait

Measurement condition ($V_{CC}=3V, T_a=25^\circ C, f(X_{IN})$: square wave input, microprocessor mode)



LOW SUPPLY VOLTAGE VERSION

7.4 Standard characteristics

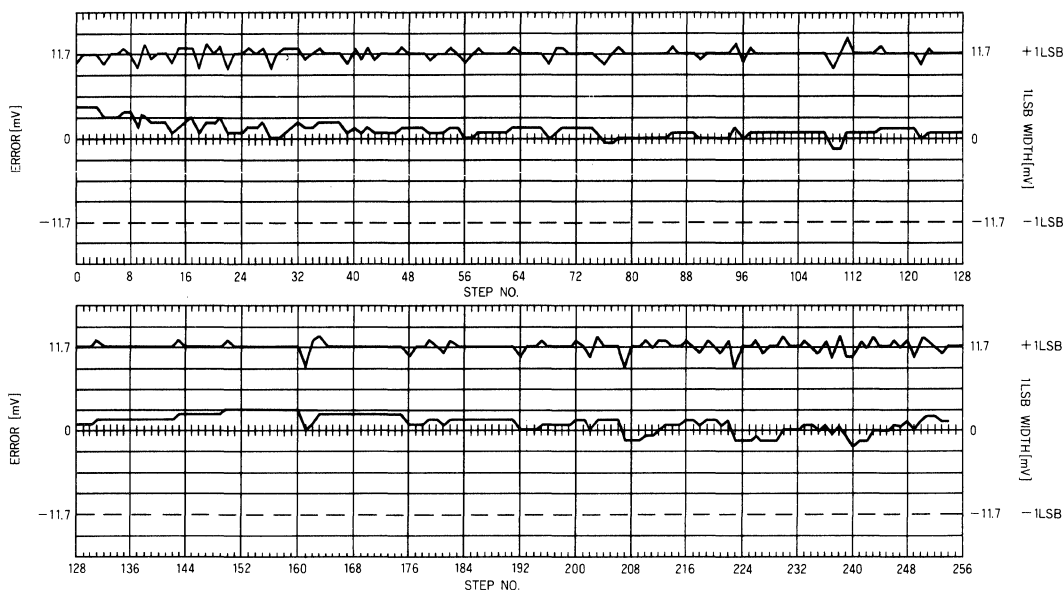
7.4.3 A-D converter standard characteristics

The lines at the bottom of the graph indicate the absolute precision errors. These are expressed as the deviation from the ideal value when the output code changes. For example, the change in output code from 04_{16} to 05_{16} should occur at $AN_i=52.7\text{mV}$, but the measured value is 2.9mV . Therefore, the measured point of change is $52.7+2.9=55.6\text{mV}$.

The lines at the top of the graph indicate the input voltage width for which the output code is constant. For example, the measured input voltage width for which the output code is $0F_{16}$ is 12.4mV . Therefore, the differential non-linear error is $12.4-11.7=0.7\text{mV}$ (0.06LSB).

[Measurement condition]

- $V_{CC} = 3\text{V}$
- $V_{REF} = 3\text{V}$
- $X_{IN} = 8\text{MHz}$
- Temp. = 25°C



LOW SUPPLY VOLTAGE VERSION

7.4 Standard characteristics

MEMORANDUM



CHAPTER 8

M37730 GROUP

- 8.1 Description
- 8.2 Functional description
- 8.3 Electrical characteristics

8.1 Description

The M37730 group is an external ROM version 16-bit microcomputer designed with high-performance silicon gate CMOS technology. It is housed in an 64-pin plastic molded QFP or 64-pin shrink plastic molded DIP. This microcomputer has a large 16M-byte address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controllers that require high-speed processing of large amounts of data.

M37730 group has the same functions as M37732 group except for a few parts. Therefore, the differences between M37730 group and M37732 group are mainly described in this chapter. M37730 group has the same functions as M37732 group except for the functions described in this chapter. Therefore, refer to "CHAPTER 2. FUNCTIONAL DESCRIPTION" for them.

8.1.1 M37730 group

The M37730 group consists of chips shown in Table 8.1.1 with the M37730S2FP as the base chip. The differences between these chips are the operating clock frequencies and the package outlines. Hereafter, the M37730 group microcomputers will be referred to simply as the M37730S2FP unless there is a specific difference by version.

Table 8.1.1 M37730 group

Type name	ROM size (bytes)	RAM size (bytes)	Clock frequency (MHz)	Package	Remarks
M37730S2FP	External	1024	8	64P6N-A	High-speed version of M37730S2FP
M37730S2AFP			16		
M37730S2BFP			25		
M37730S2SP			8	64P4B	High-speed version of M37730S2SP
M37730S2ASP			16		
M37730S2BSP			25		

8.1.2 Performance overview

Table 8.1.2 shows the performance overview of M37730S2FP.

Table 8.1.2 M37730S2FP performance overview

Parameters		Functions
Number of basic instructions		103
Instruction execution time	M37730S2FP, M37730S2SP	500ns (the fastest instruction at 8MHz frequency)
	M37730S2AFP, M37730S2ASP	250ns (the fastest instruction at 16MHz frequency)
	M37730S2BFP, M37730S2BSP	160ns (the fastest instruction at 25MHz frequency)
Operating clock frequency	M37730S2FP, M37730S2SP	8MHz (maximum)
	M37730S2AFP, M37730S2ASP	16MHz (maximum)
	M37730S2BFP, M37730S2BSP	25MHz (maximum)
Memory size	RAM	1024 bytes
Input/Output ports	Ports P5, P6	8 bits × 2
	Port P4	5 bits × 1
	Port P8	4 bits × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16 bits × 5
	TB0	16 bits × 1
Serial I/O		(UART or clock synchronous serial I/O) × 1
Watchdog timer		12 bits × 1
Interrupts		3 external, 11 internal (priority levels 0 to 7 can be set for each interrupt with software)
Clock generating circuit		Built-in (externally connected to a ceramic or crystal resonator)
Supply voltage		5V±10%
Power dissipation		30mW (operating clock frequency = 8MHz)
Input/Output characteristics	Input/Output voltage	5V
	Output current	5mA
Memory expansion		Maximum 16M bytes
Operating temperature range		-20 to 85°C
Device structure		High-performance silicon gate CMOS process
Package	M37730S2FP/AFP/BFP	64-pin plastic molded QFP
	M37730S2SP/ASP/BSP	64-pin shrink plastic molded DIP

8.1.3 Pin configuration

Figure 8.1.1 shows the M37730S2FP pin configuration and figure 8.1.2 shows the M37730S2SP pin configuration.

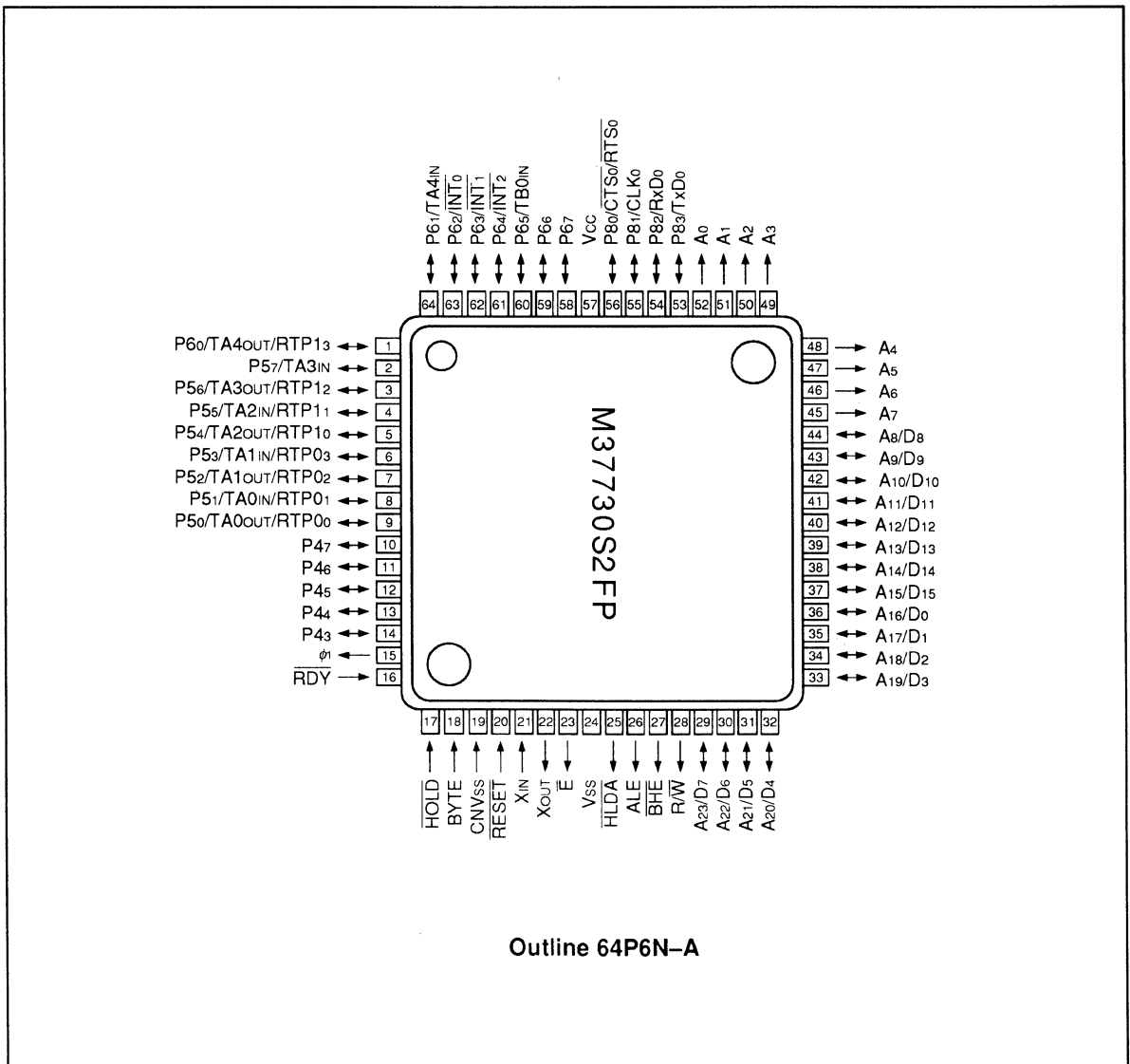


Fig. 8.1.1 M37730S2FP pin configuration (top view)

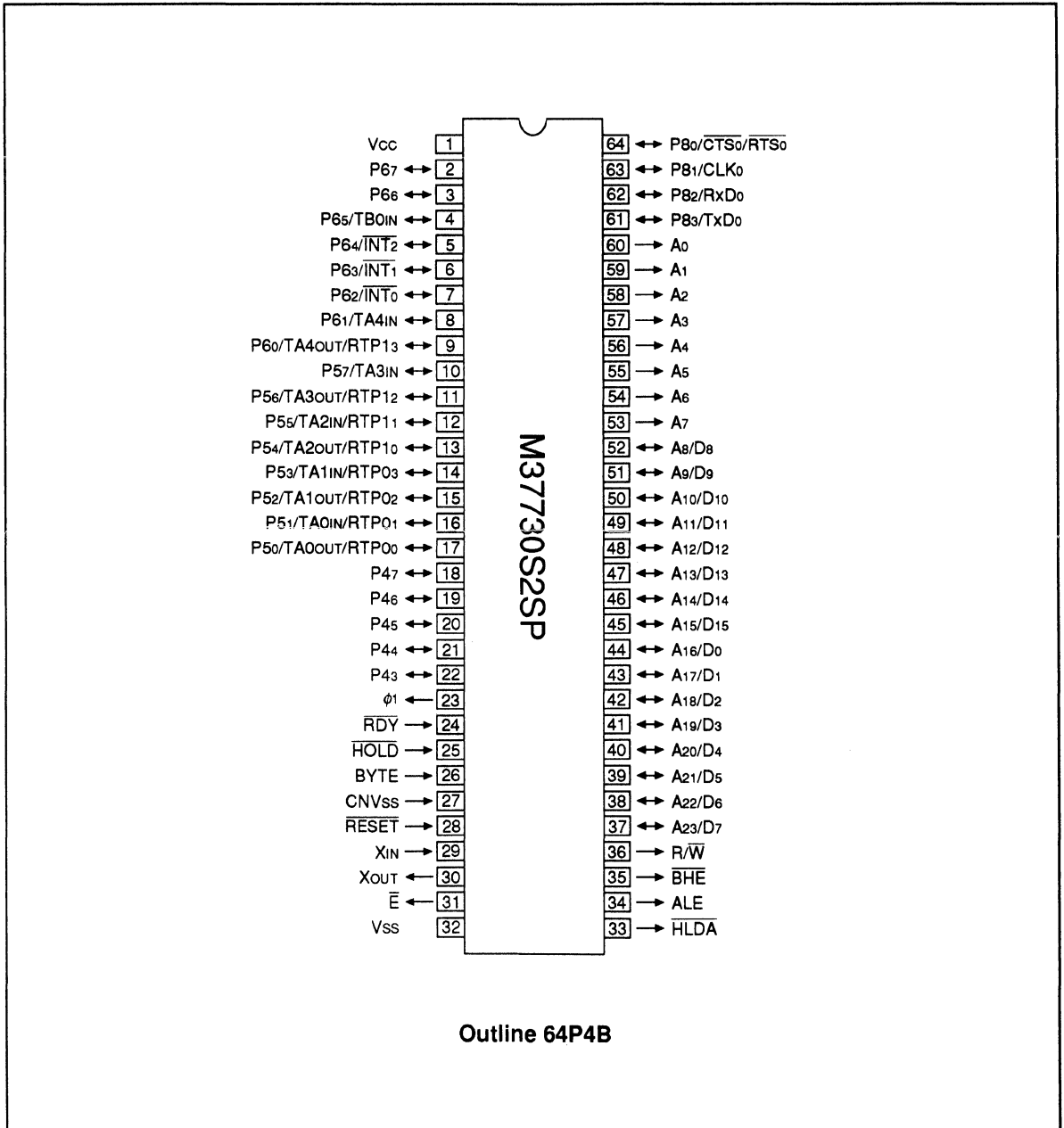


Fig. 8.1.2 M37730S2SP pin configuration (top view)

8.1.4 Pin description

Table 8.1.3 shows the pin description.

Table 8.1.3 Pin description (1)

Pin	Name	Input/Output	Functions
V _{CC} , V _{SS}	Power supply		Supply 5V±10% to V _{CC} and 0V to V _{SS} .
CNV _{SS}	CNV _{SS} input	Input	Connect to V _{CC} .
RESET	Reset input	Input	The microcomputer is reset when this pin is set to "L" level.
X _{IN}	Clock input	Input	These are the I/O pins of the internal clock generating circuit. Connect a ceramic or crystal resonator between X _{IN} and X _{OUT} . When an external clock is used, the clock source should be connected to the X _{IN} pin and the X _{OUT} pin should be left open.
X _{OUT}	Clock output	Output	
\bar{E}	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L" level.
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when the signal level is "L" and 8 bits when the signal level is "H".
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X _{IN} pin by 2.
\overline{RDY}	Ready input	Input	The microcomputer is in Ready state while "L" level is input to this pin.
\overline{HOLD}	Hold input	Input	The microcomputer is in Hold state while "L" level is input to this pin.
\overline{HLDA}	Hold acknowledge output	Output	This pin outputs "L" level when microcomputer is in Hold state.
R/W	Read/Write output	Output	The read/write signal indicates the data bus state. The data bus state is read when this signal is "H" level and write when "L" level.
\overline{BHE}	Byte high enable output	Output	"L" level is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to obtain only the address from address and data multiplex signals.
A ₀ –A ₇	Address bus low-order output	Output	Low-order 8 bits (A ₀ –A ₇) of the address bus are output.
A ₈ /D ₈ –A ₁₅ /D ₁₅	Address bus middle-order output/ Data bus high-order I/O	I/O	<ul style="list-style-type: none"> ●When the external bus width is 16 bits. (The BYTE pin is at "L" level.) High-order 8 bits (D₈–D₁₅) of the data bus are input or output when the \bar{E} output is "L" level. Middle-order 8 bits (A₈–A₁₅) of the address bus are output when \bar{E} output is "H" level. ●When the external bus width is 8 bits. (The BYTE pin is at "H" level.) Middle-order 8 bits (A₈–A₁₅) of the address bus are output.
A ₁₆ /D ₀ –A ₂₃ /D ₇	Address bus high-order output/ Data bus low-order I/O	I/O	Low-order 8 bits (D ₀ –D ₇) of the data bus are input or output when the \bar{E} output is "L" level. High-order 8 bits (A ₁₆ –A ₂₃) of the address bus are output when the \bar{E} output is "H" level.

Table 8.1.3 Pin description (2)

Pin	Name	Input/Output	Functions
P4 ₃ –P4 ₇	I/O port P4	I/O	Port P4 is a 5-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output.
P5 ₀ –P5 ₇	I/O port P5	I/O	Port P5 is an 8-bit CMOS I/O port. This port has a data direction register and each pin can be programmed for input or output. These pins also function as I/O pins for timers A0–A3.
P6 ₀ –P6 ₇	I/O port P6	I/O	This port is an 8-bit I/O port with the same function as P5. These pins can be programmed as I/O pins for timer A4, external interrupt input pins for $\overline{\text{INT}}_0$ – $\overline{\text{INT}}_2$, and input pin for timer B0.
P8 ₀ –P8 ₃	I/O port P8	I/O	This port is an 4-bit I/O port with the same function as P5. These pins can be programmed as $\overline{\text{CTS}}/\overline{\text{RTS}}$, CLK, RxD, TxD pins for UART0.

8.1.5 Block diagram

Figure 8.1.3 shows the M37730S2FP block diagram.

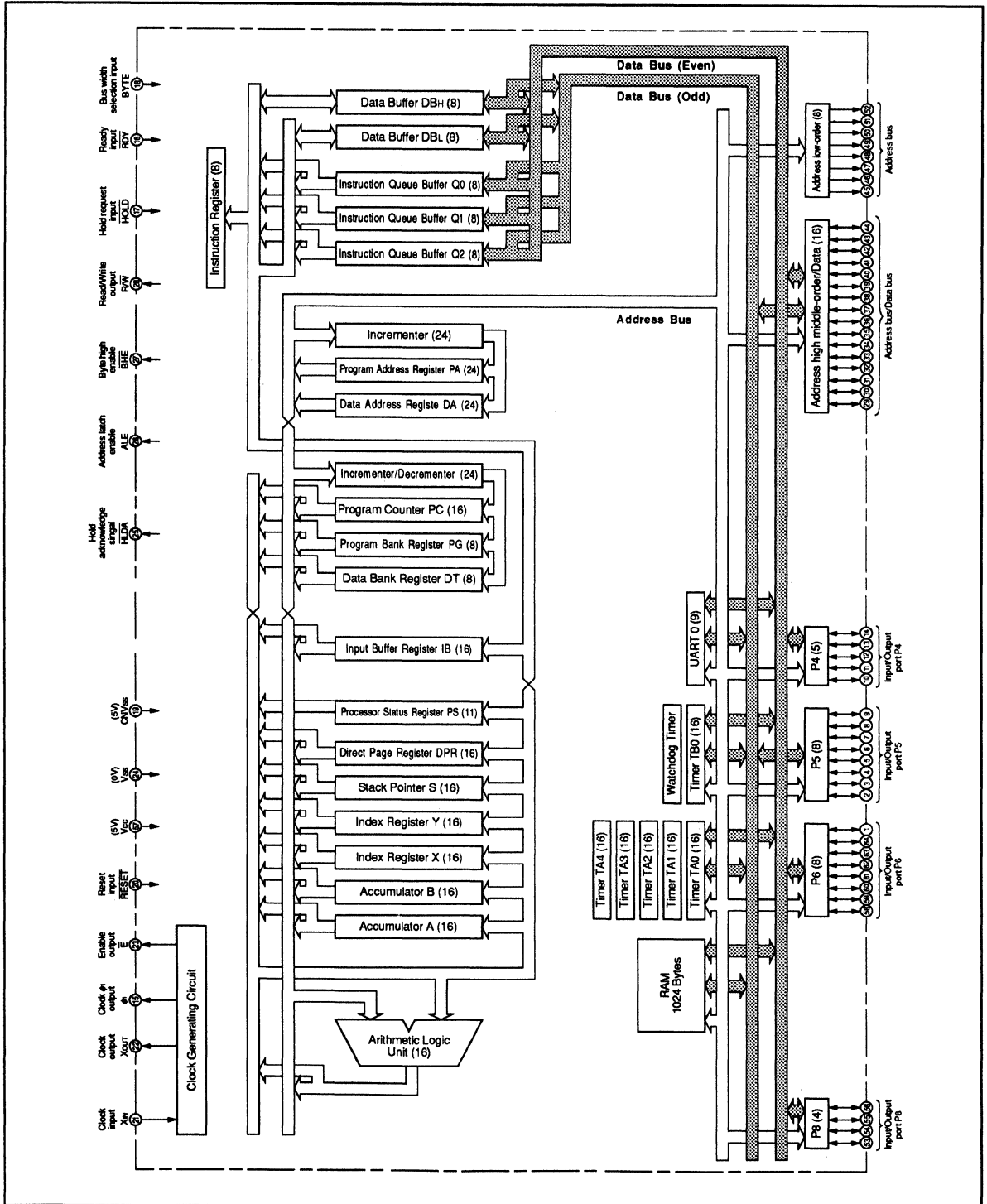


Fig. 8.1.3 M37730S2FP block diagram

M37730 GROUP

8.2 Functional description

8.2 Functional description

The functions of the M37730S2FP are described below. The differences between the M37730S2FP and M37732S4FP are mainly described in this section. M37730S2FP has the same functions as M37732S4FP except for the functions described in this section. Therefore, refer to "CHAPTER 2. FUNCTIONAL DESCRIPTION" for them.

Table 8.2.1 shows the differences between M37730 group and M37732 group.

Table 8.2.1 Differences between M37730 group and M37732 group

Functions	M37730 group	M37732 group	
Memory size	RAM1024 bytes	RAM2048 bytes	
Interrupts	14 sources	19 sources	
	External	3 sources	3 sources
	Internal	11 sources	16 sources
I/O ports	25	37	
	Port P4	5 bits	5 bits
	Port P5	8 bits	8 bits
	Port P6	8 bits	8 bits
	Port P7	8 bits	8 bits
	Port P8	4 bits	8 bits
Timer	16 bits × 6	16 bits × 8	
	Timer A	16 bits × 5 (Timer A0 to A4)	16 bits × 5 (Timer A0 to A4)
	Timer B	16 bits × 1 (Timer B0)	16 bits × 3 (Timer B0 to B2)
Pulse output function	2 (It is possible to control pulse output (enable/disable) by setting of register or external input signal.)	2	
Serial I/O	1	2	
	UART0	Clock asynchronous/synchronous	Clock asynchronous/synchronous
	UART1		Clock asynchronous/synchronous
A-D converter		One 8-bit resolution 8-channel analog input pin	
Package	64-pin plastic molded QFP (64P6N)	80-pin plastic molded QFP (80P6N-A)	
	64-pin shrink plastic molded DIP (64P4B)	80-pin plastic molded QFP (80P6S-A) 80-pin plastic molded fine-pitch QFP (80P6D-A)	

8.2.1 Addressable memory space

The M37730S2FP allocates internal RAM, external memory, I/O, and various control registers in the same memory space. Therefore, data transfer and operation can be performed with the same instruction without distinguishing memory and I/O area.

Program counter (PC) consists of 16 bits. It is used together with an 8-bit program bank register (PG) to directly access a 16M-byte address space from 0_{16} to $FFFFFF_{16}$.

Figure 8.2.1 shows the addressable memory space. Internal RAM size, SFR memory allocation and interrupt vector table memory allocation are different from M37732S4FP.

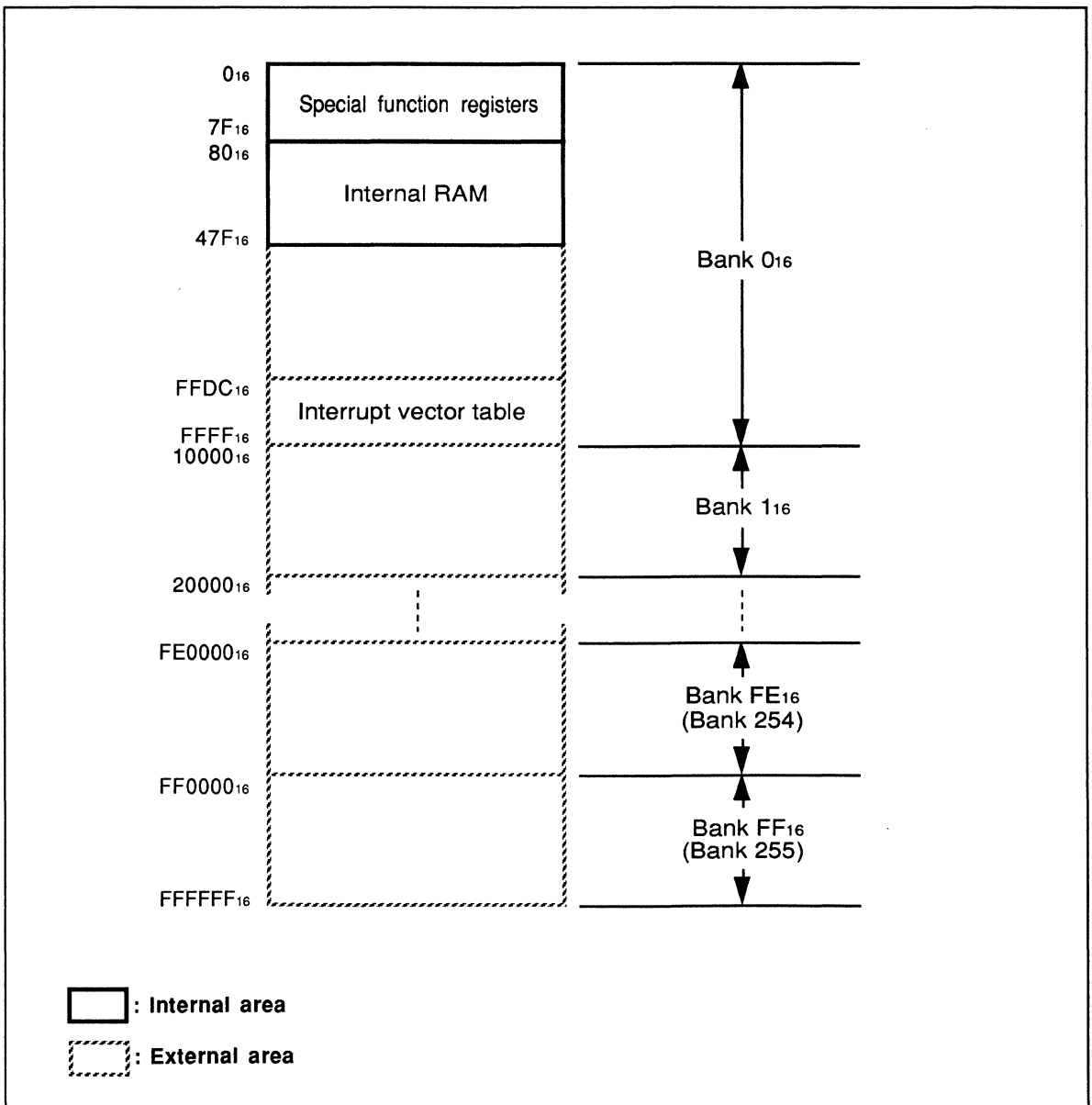


Fig. 8.2.1 Addressable memory space

8.2.2 Memory allocation

SFR (Special Function Registers) and internal RAM are allocated in internal area of bank 0₁₆. ROM must be connected to the external memory area because the M37730S2FP is an external ROM version. Figure 8.2.2 shows the memory map.

(1) SFR (Special Function Registers), Internal RAM and external memory area.

The allocated SFR, internal RAM and external memory area are described below.

- **SFR area**

Addresses 0000₁₆ to 007F₁₆ of bank 0₁₆ are the SFR (Special Function Registers) area. This area contains the control registers of internal peripheral devices, I/O ports, timers, and so on. Internal peripheral devices can be accessed through these registers. Figure 8.2.3 shows the memory map of the SFR area.

Each bit in the register can be either read-only, write-only, or read/write bit.

Refer to each block description for the register function in the SFR area.

Refer to section “8.2.9 Reset” for the state of the SFR at reset.

- **Internal RAM area**

A 1024-byte static RAM is allocated at addresses 0080₁₆ to 047F₁₆ of bank 0₁₆. In addition to storing data, the internal RAM area is used as stack area during subroutine calls and interrupts. Therefore, be careful of subroutine nesting levels and multiple interrupt levels so that important data is not destroyed.

- **External memory area**

External memory can be set to a 16M-byte space of banks 0₁₆ to FF₁₆ excluding SFR area (addresses 0000₁₆ to 007F₁₆) and internal RAM area (addresses 0080₁₆ to 047F₁₆).

The interrupt vector table allocated to addresses FFDC₁₆ to FFFF₁₆ of bank 0₁₆ must be set in ROM. This area contains the start address of the interrupt service routine for each interrupt source and reset.

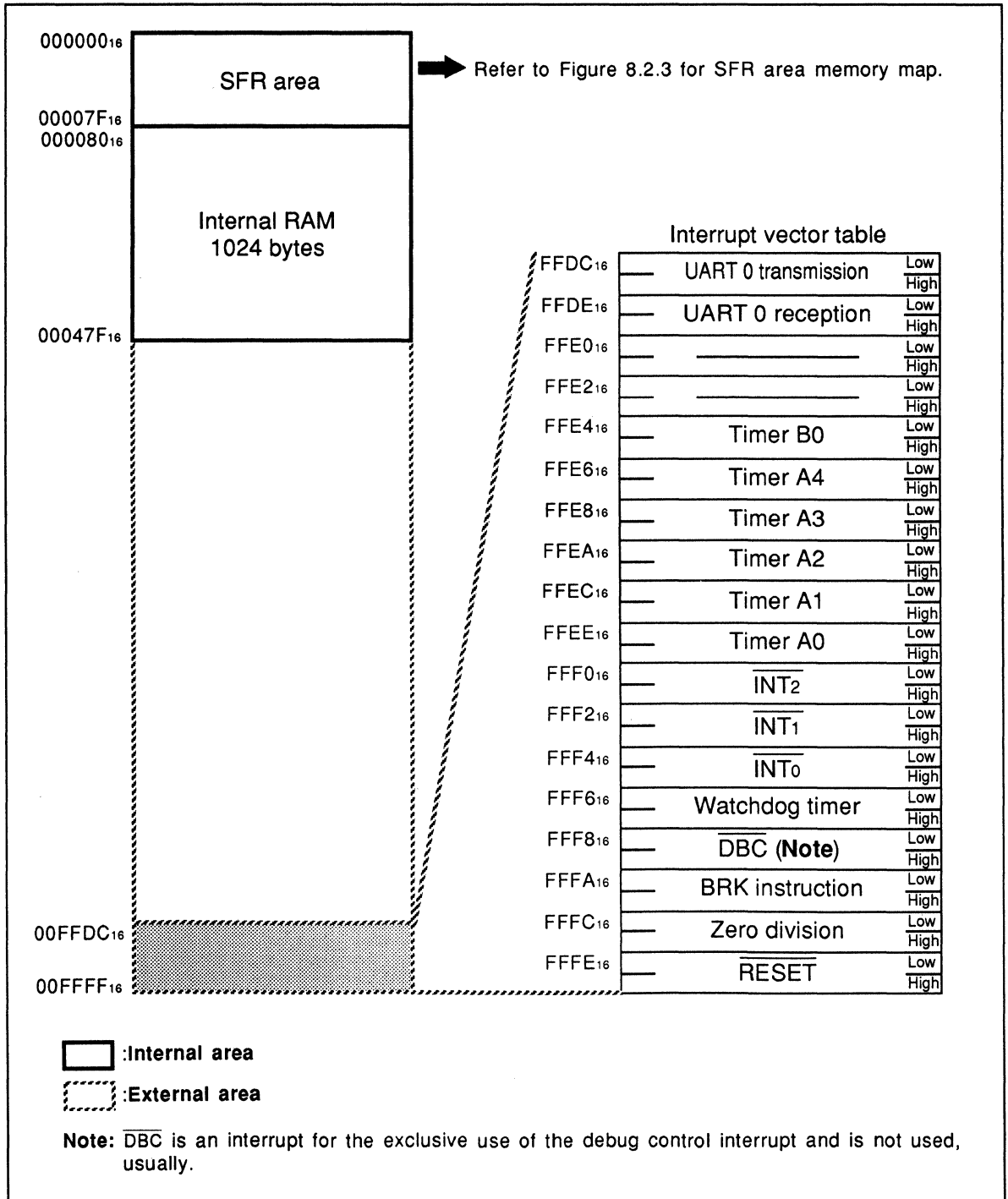


Fig. 8.2.2 Memory map

Address (Hexadecimal notation)		Address (Hexadecimal notation)	
000000		000040	Count start flag
000001		000041	
000002		000042	One-shot start flag
000003		000043	
000004		000044	Up-down flag
000005		000045	
000006		000046	Timer A0 register
000007		000047	
000008		000048	Timer A1 register
000009		000049	
00000A	Port P4 register	00004A	Timer A2 register
00000B	Port P5 register	00004B	
00000C	Port P4 direction register	00004C	Timer A3 register
00000D	Port P5 direction register	00004D	
00000E	Port P6 register	00004E	Timer A4 register
00000F		00004F	
000010	Port P6 direction register	000050	Timer B0 register
000011		000051	
000012	Port P8 register	000052	
000013		000053	
000014	Port P8 direction register	000054	
000015		000055	
000016		000056	Timer A0 mode register
000017		000057	Timer A1 mode register
000018		000058	Timer A2 mode register
000019		000059	Timer A3 mode register
00001A		00005A	Timer A4 mode register
00001B		00005B	Timer B0 mode register
00001C		00005C	
00001D		00005D	
00001E		00005E	Processor mode register
00001F		00005F	
000020		000060	Watchdog timer
000021		000061	Watchdog timer frequency selection flag
000022		000062	Waveform output mode register
000023		000063	
000024		000064	Pulse output data register1
000025		000065	Pulse output data register0
000026		000066	
000027		000067	
000028		000068	
000029		000069	
00002A		00006A	
00002B		00006B	
00002C		00006C	
00002D		00006D	
00002E		00006E	
00002F		00006F	
000030	UART 0 transmit/receive mode register	000070	
000031	UART 0 baud rate register	000071	UART 0 transmission interrupt control register
000032		000072	UART 0 receive interrupt control register
000033	UART 0 transmission buffer register	000073	
000034	UART 0 transmit/receive control register 0	000074	
000035	UART 0 transmit/receive control register 1	000075	Timer A0 interrupt control register
000036		000076	Timer A1 interrupt control register
000037	UART 0 receive buffer register	000077	Timer A2 interrupt control register
000038		000078	Timer A3 interrupt control register
000039		000079	Timer A4 interrupt control register
00003A		00007A	Timer B0 interrupt control register
00003B		00007B	
00003C		00007C	
00003D		00007D	INT0 interrupt control register
00003E		00007E	INT1 interrupt control register
00003F		00007F	INT2 interrupt control register

Fig. 8.2.3 SFR area memory map

(2) Processor mode

M37730S2FP can operate in microprocessor mode. ROM, RAM and peripherals can be expanded to any area (except for SFR and internal RAM area) within a 16M-byte addressable memory space.

Figure 8.2.4 shows the structure of the processor mode register. Bits 2 to 5 of the processor mode register have the same function as the M37732S4FP. Therefore, refer to section "2.4.3 External memory area bus control" for wait bit, section "2.6.6 Interrupt priority level detection time" for interrupt priority detection time selection bits and section "3.3 Software reset" for software reset bit.

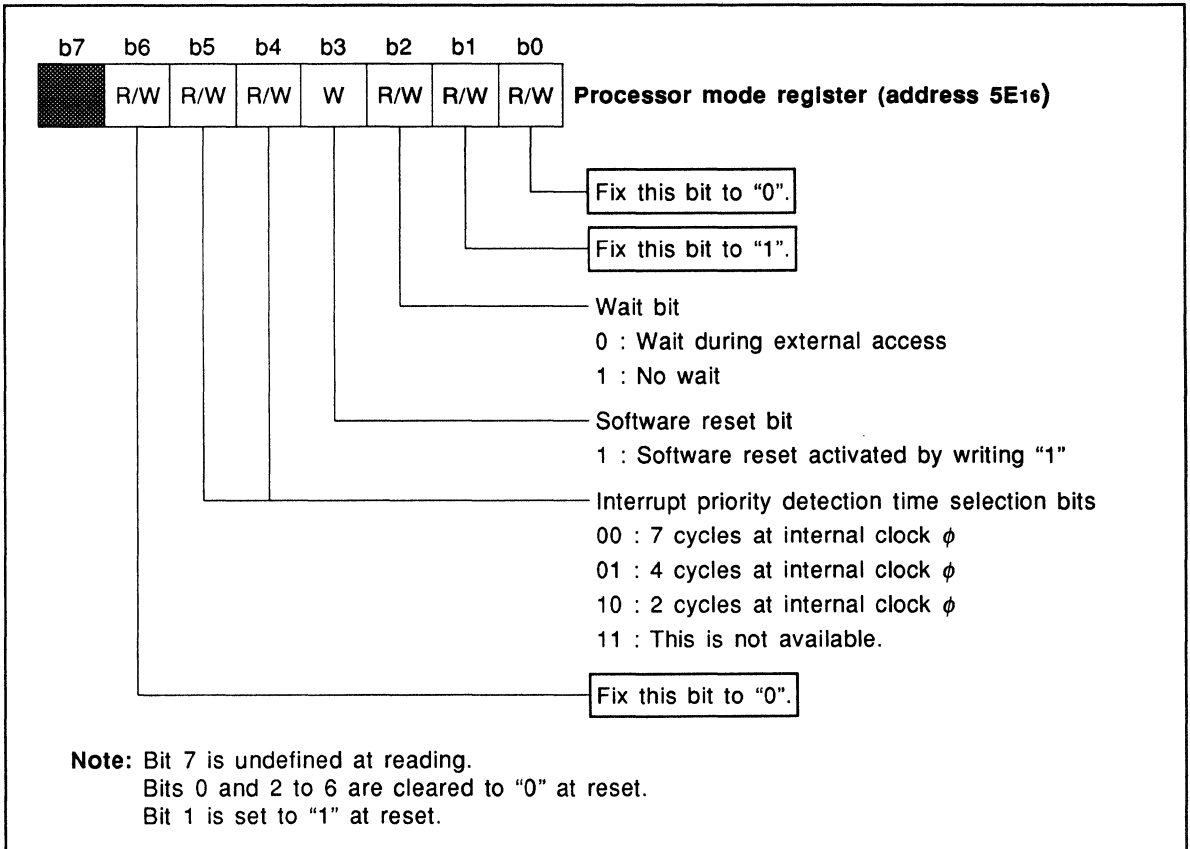


Fig. 8.2.4 Processor mode register structure

8.2.3 Input/Output pins

The M37730S2FP has 25 programmable I/O pins: ports P4–P6, and P8. Ports P5, P6, and P8 also function as I/O pins of the internal peripheral devices. The programmable I/O port and pin function are described below.

(1) Programmable I/O port

Each of the programmable I/O ports P4–P6 and P8 has a direction register and a port register in the SFR area of bank 0₁₆. The direction register can be used to program the port for input mode or output mode by one bit. Figure 8.2.5 shows the memory allocation of the direction registers and port registers. The differences with M37732S4FP are that the M37730S2FP does not have ports P7 and P8₄–P8₇. Therefore, there are no port P7 direction register nor port P7 register and the structure of the port P8 direction register and port P8 register is different. Figure 8.2.6 shows the register structure of the port P4–P6 and P8 direction registers and Figure 8.2.7 shows the relationship between port P4–P6 and P8 registers and pins.

Address	
8 ₁₆	
9 ₁₆	
A ₁₆	Port P4 register
B ₁₆	Port P5 register
C ₁₆	Port P4 direction register
D ₁₆	Port P5 direction register
E ₁₆	Port P6 register
F ₁₆	
10 ₁₆	Port P6 direction register
11 ₁₆	
12 ₁₆	Port P8 register
13 ₁₆	
14 ₁₆	Port P8 direction register
15 ₁₆	
16 ₁₆	

Fig. 8.2.5 Memory allocation of direction registers and port registers

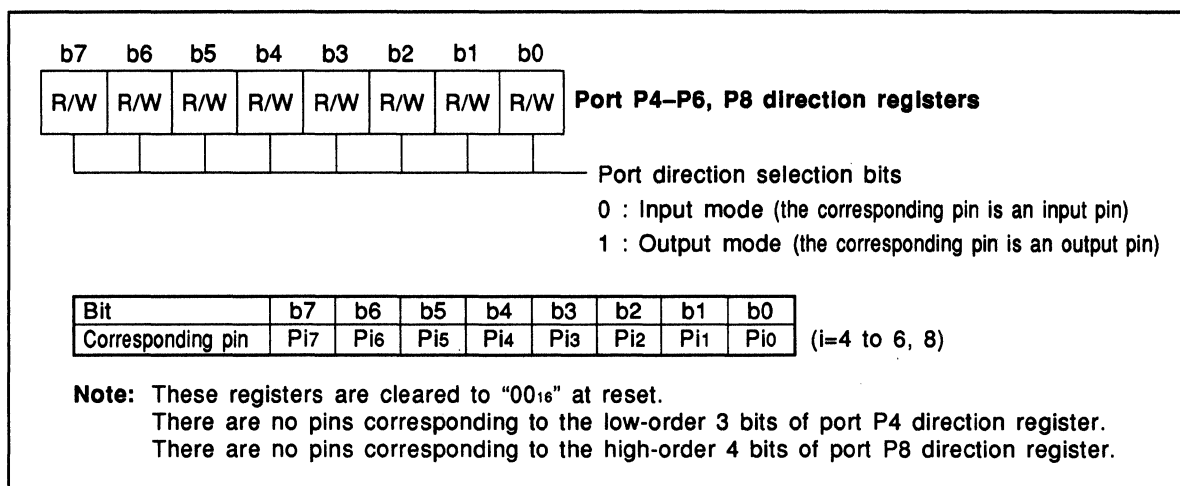


Fig. 8.2.6 Port P4–P6, P8 direction registers structure

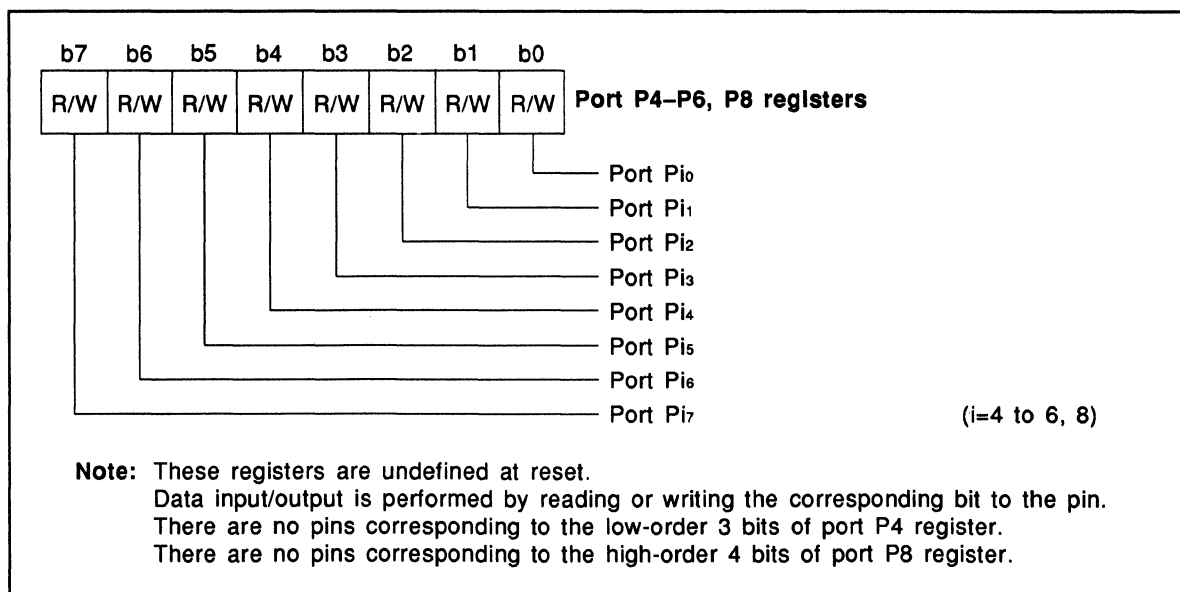


Fig. 8.2.7 Relationship between port P4–P6, P8 registers and pins

(2) Ports P4–P6, P8 function

In addition to the programmable I/O port function, the 20 pins of ports P5, P6, and P8 also function as the I/O pins for external interrupt, timer, and serial I/O. Port P4 functions only as programmable I/O port. Refer to the respective section for the peripheral device I/O pin function of ports P5, P6, and P8.

Ports P4–P6 and P8 all function as programmable I/O port after removing reset.

Figure 8.2.8 shows the port peripheral circuit and Table 8.2.2 shows the functions of ports P4–P6, and P8.

When ports P5, P6, and P8 are used as output pins for internal peripheral device, they are forced to output mode. To use them as input pins for internal peripheral device, set the port direction register to input mode because the contents of the port direction register has higher priority.

(3) Function of each pin

The function of each pin is identical to M37732S4FP. Refer to “2.5.3 Pin functions” for details. However, pins AV_{CC} , AV_{SS} , V_{REF} , AN_0 – AN_6 , and AN_7/\overline{AD}_{TRG} do not exist because the M37730S2FP is not equipped with an A-D converter.

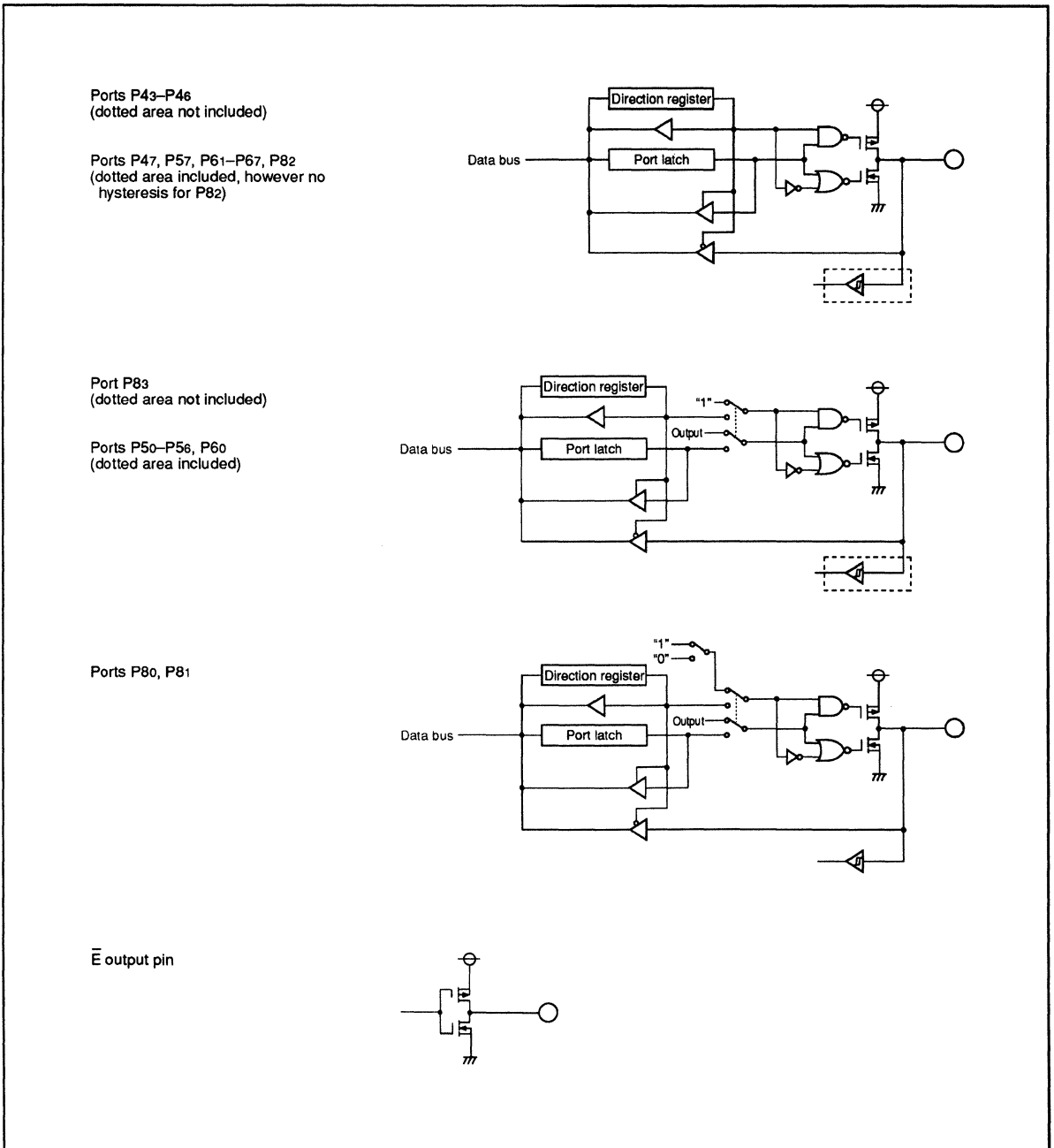


Fig. 8.2.8 Port peripheral circuit

Table 8.2.2 Function of ports P4–P6, P8

Pin	Function 1	Function 2
Port P4 (P4 ₃ –P4 ₇)	Programmable I/O ports	
Port P5 (P5 ₀ –P5 ₇)	Programmable I/O ports	<p>TA0_{OUT}/RTP0₀ (P5₀) :Timer A0 I/O pin / RTP0₀ output pin</p> <p>TA0_{IN}/RTP0₁ (P5₁) :Timer A0 input pin / RTP0₁ output pin</p> <p>TA1_{OUT}/RTP0₂ (P5₂) :Timer A1 I/O pin / RTP0₂ output pin</p> <p>TA1_{IN}/RTP0₃ (P5₃) :Timer A1 input pin / RTP0₃ output pin</p> <p>* RTP0_n (n=0 to 3) output pin is available when RTP0 function is selected.</p> <p>TA2_{OUT}/RTP1₀ (P5₄) :Timer A2 I/O pin / RTP1₀ output pin</p> <p>TA2_{IN}/RTP1₁ (P5₅) :Timer A2 input pin / RTP1₁ output pin</p> <p>TA3_{OUT}/RTP1₂ (P5₆) :Timer A3 I/O pin / RTP1₂ output pin</p> <p>* RTP1_m (m=0 to 2) output pin is available when RTP1 function is selected.</p> <p>TA3_{IN} (P5₇) :Timer A3 input pin</p>
Port P6 (P6 ₀ –P6 ₇)	Programmable I/O ports	<p>TA4_{OUT}/RTP1₃ (P6₀) :Timer A4 I/O pin / RTP1₃ output pin</p> <p>* RTP1₃ output pin is available when RTP1 function is selected.</p> <p>TA4_{IN} (P6₁) :Timer A4 input pin</p> <p>INT₀ (P6₂) :INT₀ interrupt signal input pin</p> <p>INT₁ (P6₃) :INT₁ interrupt signal input pin</p> <p>INT₂ (P6₄) :INT₂ interrupt signal input pin</p> <p>TB0_{IN} (P6₅) :Timer B0 input pin</p>
Port P8 (P8 ₀ –P8 ₃)	Programmable I/O ports	<p>CTS₀/RTS₀ (P8₀) :UART0 transmit enable signal input pin when CTS₀ function is selected</p> <p>:UART0 receive enable signal output pin when RTS₀ function is selected</p> <p>CLK₀ (P8₁) :UART0 transmit/receive clock I/O pin</p> <p>RxD₀ (P8₂) :UART0 receive data input pin</p> <p>TxD₀ (P8₃) :UART0 transmit data output pin</p>

8.2.4 Interrupts

M37730S2FP has 14 sources of interrupts (3 external and 11 internal).

The function and operation of the interrupts are identical to M37732S4FP. The structure of each interrupt control register is also identical to M37732S4FP. Use the **SEB** and **CLB** instructions when setting each interrupt control register.

Figure 8.2.9 shows the memory map of the interrupt control registers. Table 8.2.3 shows the sources of interrupts and the corresponding vector address.

Address	
70 ₁₆	
71 ₁₆	UART0 transmission interrupt control register
72 ₁₆	UART0 receive interrupt control register
73 ₁₆	
74 ₁₆	
75 ₁₆	Timer A0 interrupt control register
76 ₁₆	Timer A1 interrupt control register
77 ₁₆	Timer A2 interrupt control register
78 ₁₆	Timer A3 interrupt control register
79 ₁₆	Timer A4 interrupt control register
7A ₁₆	Timer B0 interrupt control register
7B ₁₆	
7C ₁₆	
7D ₁₆	$\overline{\text{INT}}_0$ interrupt control register
7E ₁₆	$\overline{\text{INT}}_1$ interrupt control register
7F ₁₆	$\overline{\text{INT}}_2$ interrupt control register

Fig. 8.2.9 Interrupt control registers memory map

Table 8.2.3 Interrupt sources and vector address

Interrupt source	Vector address		Remarks
	High-order address	Low-order address	
Reset (Note 1)	00FFFF ₁₆	00FFFE ₁₆	Non-maskable
Zero division	00FFFD ₁₆	00FFFC ₁₆	Non-maskable software interrupt
BRK instruction	00FFFB ₁₆	00FFFA ₁₆	Non-maskable software interrupt
DBC (Note 2)	00FFF9 ₁₆	00FFF8 ₁₆	Not used normally
Watchdog timer	00FFF7 ₁₆	00FFF6 ₁₆	Non-maskable interrupt
INT ₀	00FFF5 ₁₆	00FFF4 ₁₆	External interrupt due to INT ₀ pin input signal
INT ₁	00FFF3 ₁₆	00FFF2 ₁₆	External interrupt due to INT ₁ pin input signal
INT ₂	00FFF1 ₁₆	00FFF0 ₁₆	External interrupt due to INT ₂ pin input signal
Timer A0	00FFE1 ₁₆	00FEE ₁₆	Timer A0 internal interrupt
Timer A1	00FFED ₁₆	00FFEC ₁₆	Timer A1 internal interrupt
Timer A2	00FFEB ₁₆	00FFEA ₁₆	Timer A2 internal interrupt
Timer A3	00FFE9 ₁₆	00FFE8 ₁₆	Timer A3 internal interrupt
Timer A4	00FFE7 ₁₆	00FFE6 ₁₆	Timer A4 internal interrupt
Timer B0	00FFE5 ₁₆	00FFE4 ₁₆	Timer B0 internal interrupt
UART0 reception	00FFDF ₁₆	00FFDE ₁₆	Valid only when the UART0 function is selected.
UART0 transmission	00FFDD ₁₆	00FFDC ₁₆	

Note 1: Reset is included in this table.

Note 2: The DBC is an interrupt for the exclusive use of the debug control interrupt and is not used, usually.

8.2.5 Timer A, timer B

Timer A of M37730S2FP consists of five 16-bit timers (timer A0 to A4). Timer B consists of one 16-bit timer (timer B0).

The function and operation of each timer are identical to M37732S4FP. Refer to "2.7 Timer A" and "2.8 Timer B" for details. The difference with the M37732S4FP is described below.

- ① The timer B1 and B2 related control registers available with M37732S4FP do not exist in M37730S2FP. Figure 8.2.10 shows the memory map of timer related control registers.

Address	
39 ₁₆	
40 ₁₆	Count start flag
41 ₁₆	
42 ₁₆	One-shot start flag
43 ₁₆	
44 ₁₆	Up-down flag
45 ₁₆	
46 ₁₆	—Timer A0 register—
47 ₁₆	
48 ₁₆	—Timer A1 register—
49 ₁₆	
4A ₁₆	—Timer A2 register—
4B ₁₆	
4C ₁₆	Timer A3 register
4D ₁₆	
4E ₁₆	—Timer A4 register—
4F ₁₆	
50 ₁₆	—Timer B0 register—
51 ₁₆	
52 ₁₆	
53 ₁₆	
54 ₁₆	
55 ₁₆	
56 ₁₆	Timer A0 mode register
57 ₁₆	Timer A1 mode register
58 ₁₆	Timer A2 mode register
59 ₁₆	Timer A3 mode register
5A ₁₆	Timer A4 mode register
5B ₁₆	Timer B0 mode register
5C ₁₆	

Fig. 8.2.10 Memory map of timer related control registers

- ② The structure of port P6 direction register is different from M37732S4FP. Figure 8.2.11 shows the relationship between port P6 direction register and timer pins.

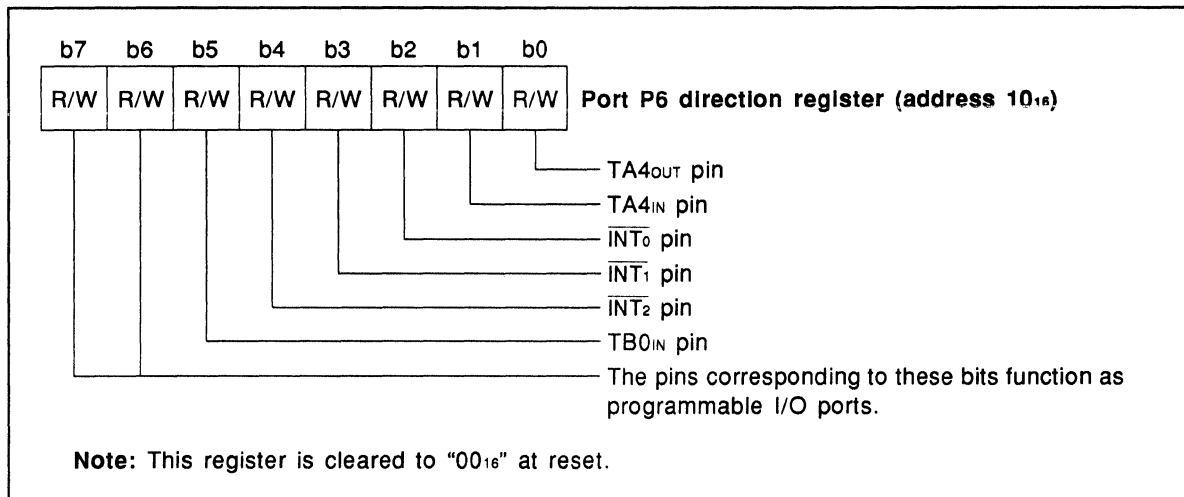


Fig. 8.2.11 Relationship between port P6 direction register and timer pins

- ③ The structure of count start flag is different from M37732S4FP. Figure 8.2.12 shows the structure of count start flag.

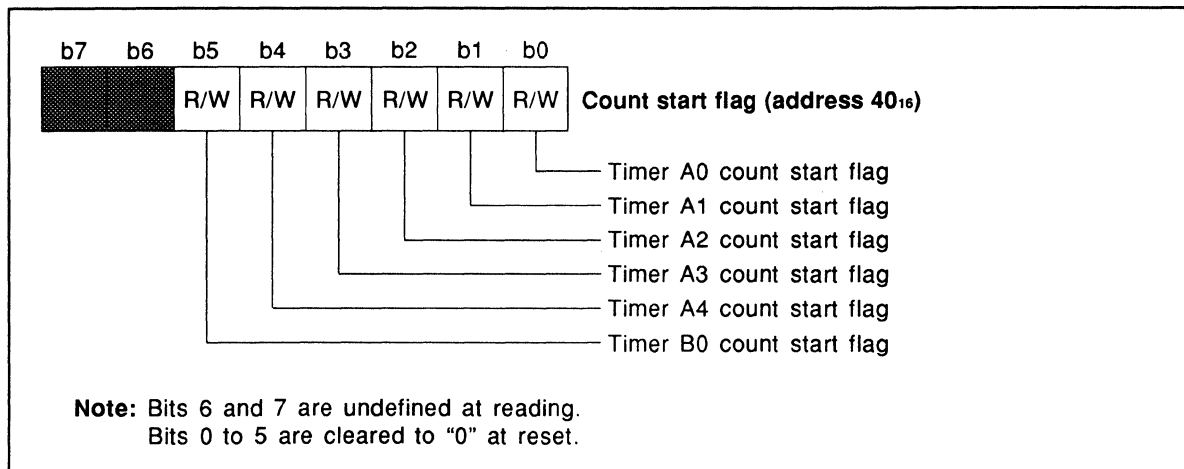


Fig. 8.2.12 Count start flag register structure

8.2.6 Pulse output function

The pulse output function (pulse output port mode) of M37730S2FP allows pulse output control (enable/disable) in addition to the functions of M37732S4FP. The pulse output operation is identical to M37732S4FP. However, pulse output pins for RTP1 are slightly different.

Therefore, the register structure and settings of some control registers are different from M37732S4FP. Figure 8.2.13 shows the block diagram of pulse output port mode. The differences are described below.

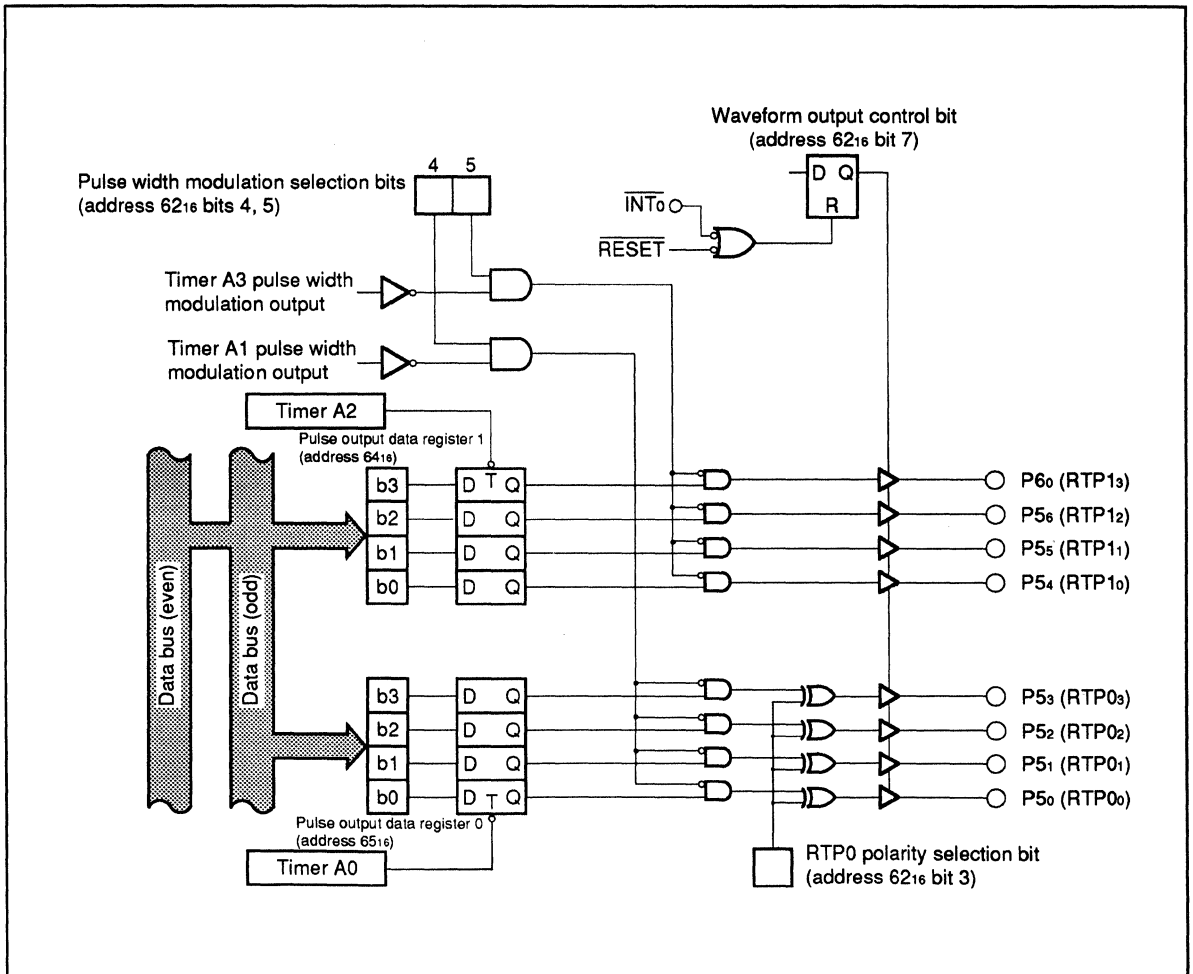


Fig. 8.2.13 Pulse output port mode block diagram

(1) Pulse output pin

The pulse output pin for the RTP1₃ of the M37730S2FP is shared with port P6₀ (TA4_{OUT}). The structure of other pulse output pins is identical to M37732S4FP.

Figure 8.2.14 shows the structure of the pulse output pins and pulse output data registers 0 and 1.

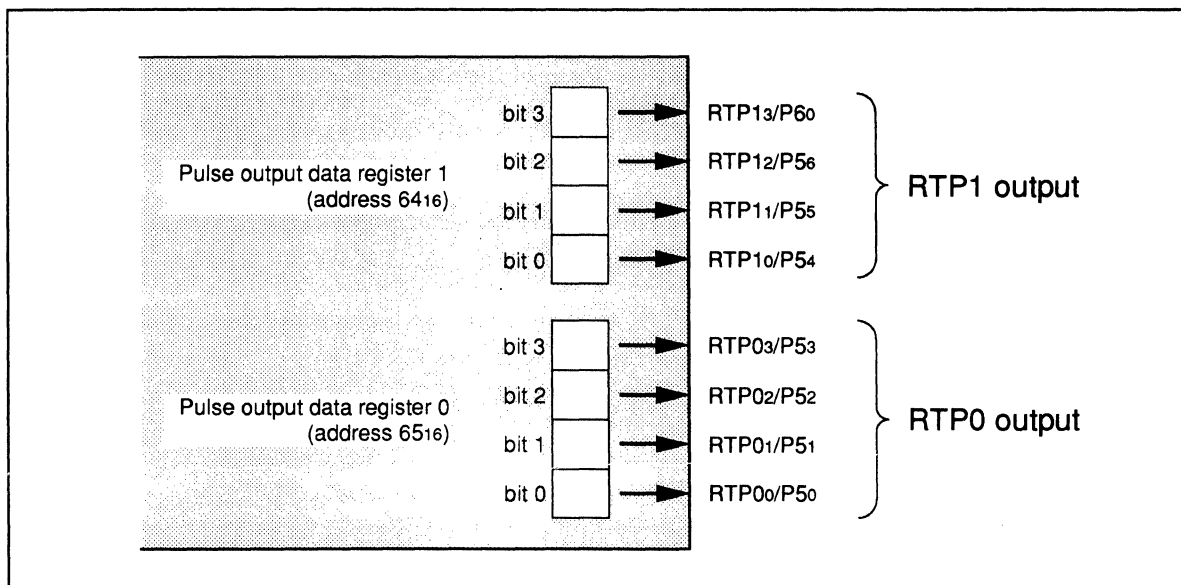


Fig. 8.2.14 Pulse output pins and pulse output data register 0, 1 structure

(2) Pulse output control

Figure 8.2.15 shows the relationship between port P6 direction register and pins. Figure 8.2.16 shows the structure of the waveform output mode register.

Pulse output is controlled by the waveform output control bit (bit 7 at address 62₁₆) of the waveform output mode register. Pulse output is enabled when the waveform output control bit is set to "1". When it is "0", pulse output is disabled and the pulse output pin becomes floating state.

When "L" level signal is input to $\overline{\text{INT}}_0$ pin (P6₂), the waveform output control bit is cleared to "0" and pulse output is disabled. Refer to page 266 for the enabling procedure after it is disabled. When using the pulse output port mode (pulse output function) disabling function with input signal to $\overline{\text{INT}}_0$ pin, set to input mode by setting bit 2 of the port P6 direction register (address 10₁₆) to "0". When the pulse output port mode disabling function with input signal to $\overline{\text{INT}}_0$ pin is not used, set to input mode by setting bit 2 of the port P6 direction register to "0" and pull up externally.

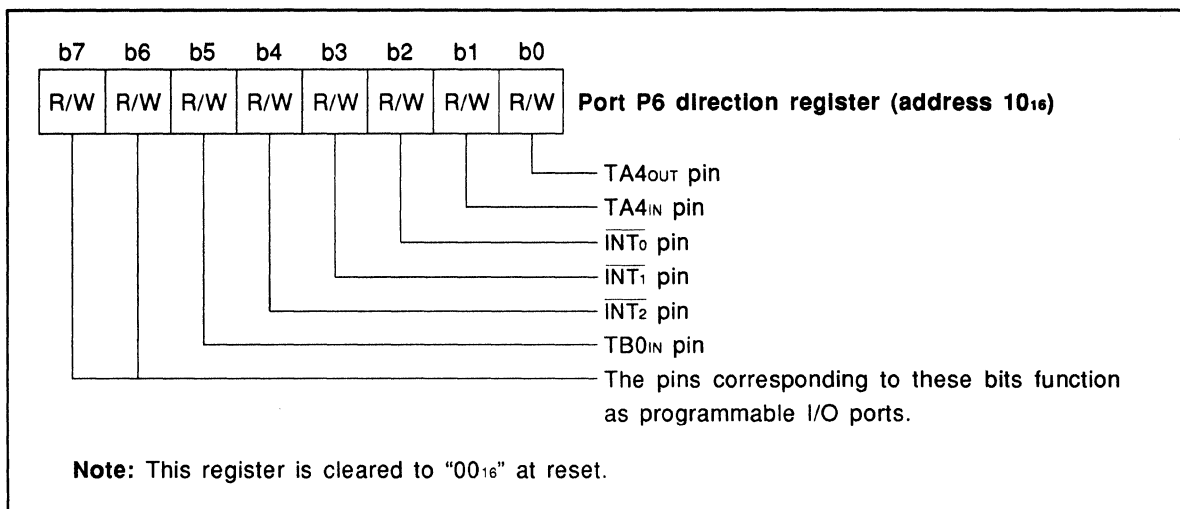


Fig. 8.2.15 Relationship between port P6 direction register and pins

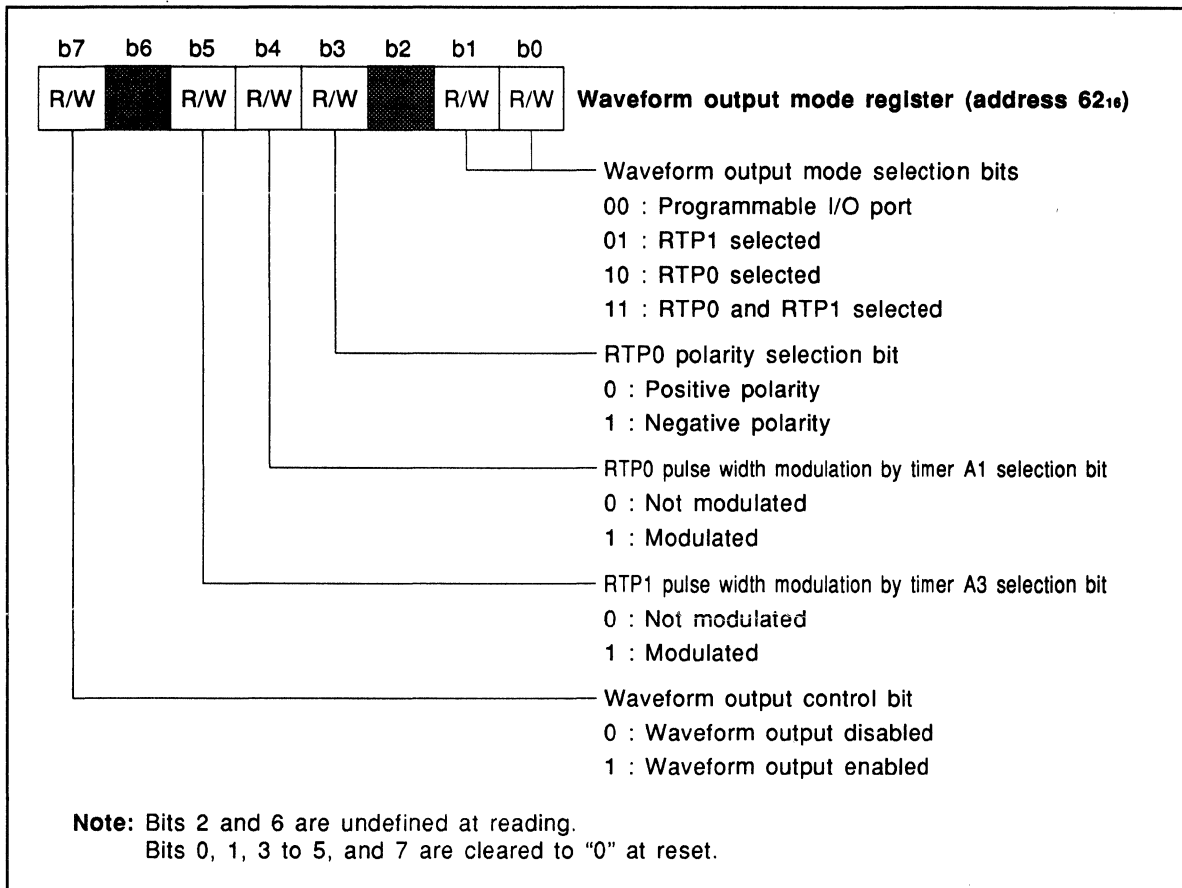


Fig. 8.2.16 Waveform output mode register structure

When pulse output is disabled and pulse output pin becomes floating state with "L" level input to $\overline{\text{INT}}_0$ pin during pulse output, pulse output is resumed when the waveform output control bit is set to "1" after returning the $\overline{\text{INT}}_0$ pin to "H" level. Pulse output is not resumed by simply returning the $\overline{\text{INT}}_0$ pin to "H" level. Furthermore, pulse output is not resumed when the waveform output control bit is set to "1" while the $\overline{\text{INT}}_0$ pin is set to "L" level. Figure 8.2.17 shows how to resume pulse output after disabling it with $\overline{\text{INT}}_0$ input signal.

The waveform output control bit is cleared to "0" when "L" level is applied to the $\overline{\text{RESET}}$ pin or when a software reset is executed.

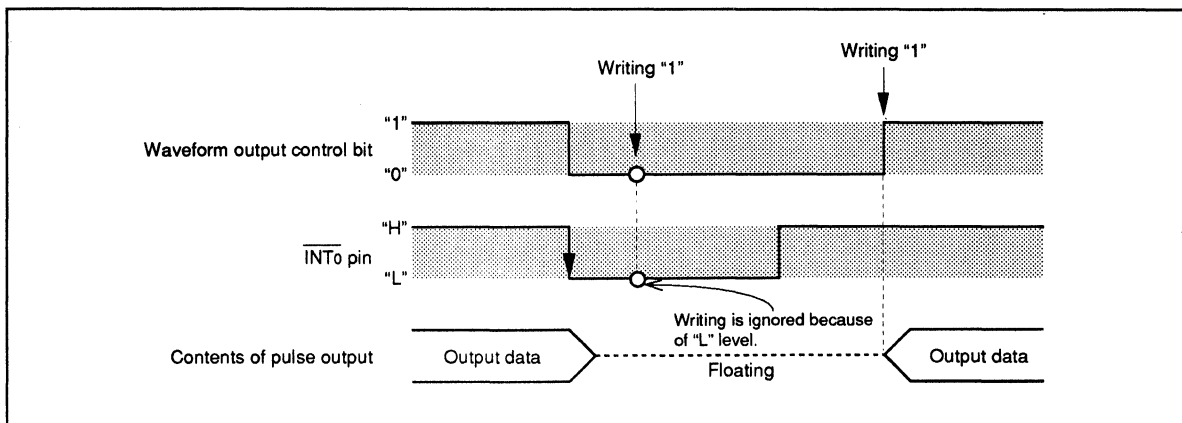


Fig. 8.2.17 How to resume pulse output after disabling it with $\overline{\text{INT}}_0$ input signal

(3) Timer A0–A4 mode registers

When using RTP0 of the M37730S2FP, the setting contents and setting method of the timer A0 and A1 mode registers are identical to M37732S4FP.

When using RTP1, the setting contents of the timer A2 mode register are identical to M37732S4FP. However, the RTP1₃ pin of M37730S2FP is shared with the TA4_{OUT} pin and the TA3_{IN} pin does not function as pulse output pin. Therefore, the operation and setting contents of the timer A3 and A4 mode registers are different from those of M37732S4FP.

● Timer A3

When performing pulse width modulation for RTP1, the setting contents of the timer A3 mode register are identical to M37732S4FP.

When pulse width modulation is not performed for RTP1, timer A3 can be used in timer mode or event counter mode. Regardless of whether timer A3 is used or not, bit 2 of the timer A3 mode register must be fixed to "1" in order to use port P5_s as RTP1₂ pin.

When operating in timer mode, the gate function can be selected. Figure 8.2.18 shows the structure of the timer A3 mode register in timer mode (not using pulse width modulation for RTP1).

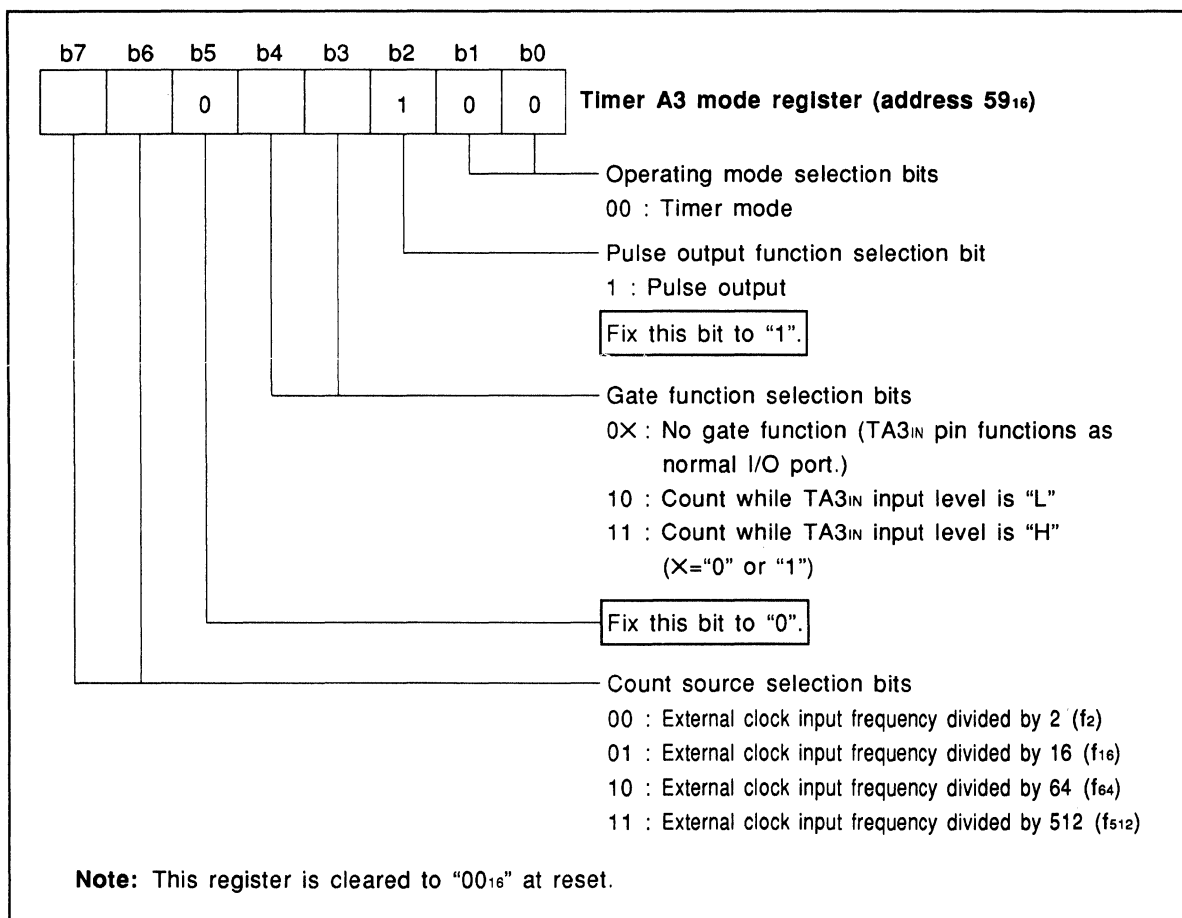


Fig. 8.2.18 Timer A3 mode register structure in timer mode (not using pulse width modulation for RTP 1)

When operating in event counter mode, only the contents of the up/down flag are used to switch the up/down factor. Therefore, bit 4 of the timer A3 mode register must be fixed to "0". Figure 8.2.19 shows the structure of the timer A3 mode register in event counter mode (not using pulse width modulation for RTP1).

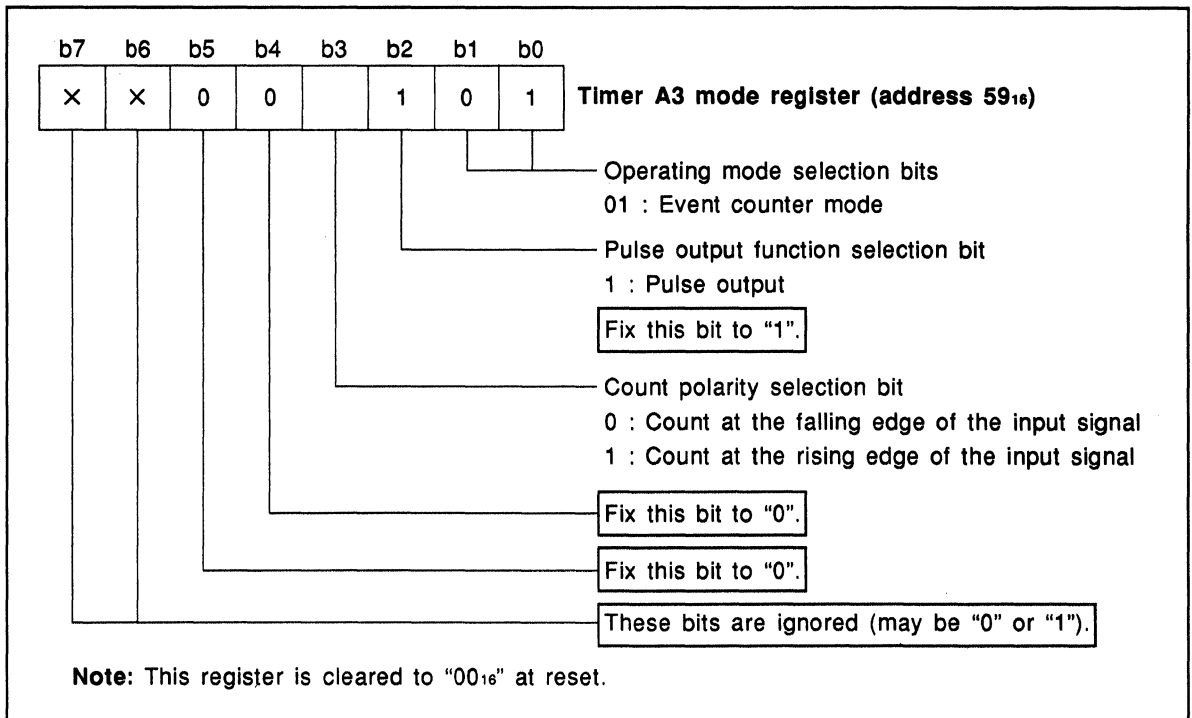


Fig. 8.2.19 Timer A3 mode register structure in event counter mode (not using pulse width modulation for RTP1)

● Timer A4

When RTP1 is selected, timer A4 can be used in timer mode or event counter mode. Regardless of whether timer A4 is used or not, bit 2 of the timer A4 mode register must be fixed to "1" in order to use port P6₀ as RTP1₃ pin.

The function and operation of timer A4 are identical to timer A3 when pulse width modulation is not performed for RTP1. Therefore, the structure of the timer A4 mode register when operating in timer mode is identical to Figure 8.2.18. The structure of the timer A4 mode register when operating in event counter mode is identical to Figure 8.2.19.

8.2.7 Serial I/O

The serial I/O of the M37730S2FP consists of one UART0. UART0 can operate as clock synchronous or asynchronous (UART) type.

The function and operation of UART0 are identical to M37732S4FP. The following are different from M37732S4FP.

- ① M37730S2FP does not have UART1 related control registers.
- ② The structure of port P8 direction register is different.

Figure 8.2.20 shows the memory map of the serial I/O related control registers and Figure 8.2.21 shows the relationship between port P8 direction register and serial I/O pins.

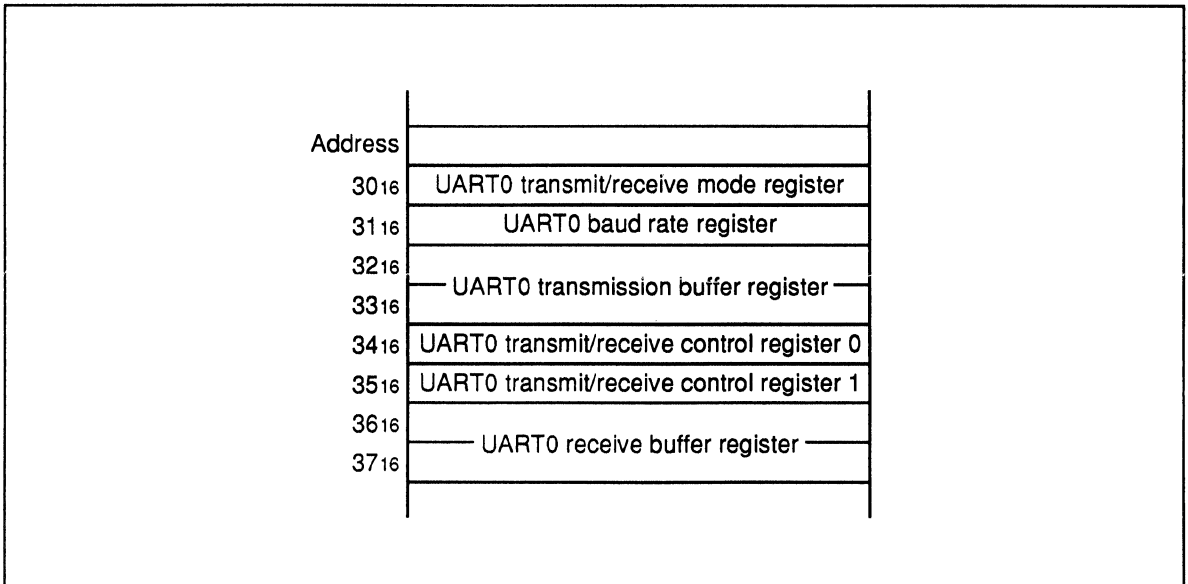


Fig. 8.2.20 Memory map of serial I/O related control registers

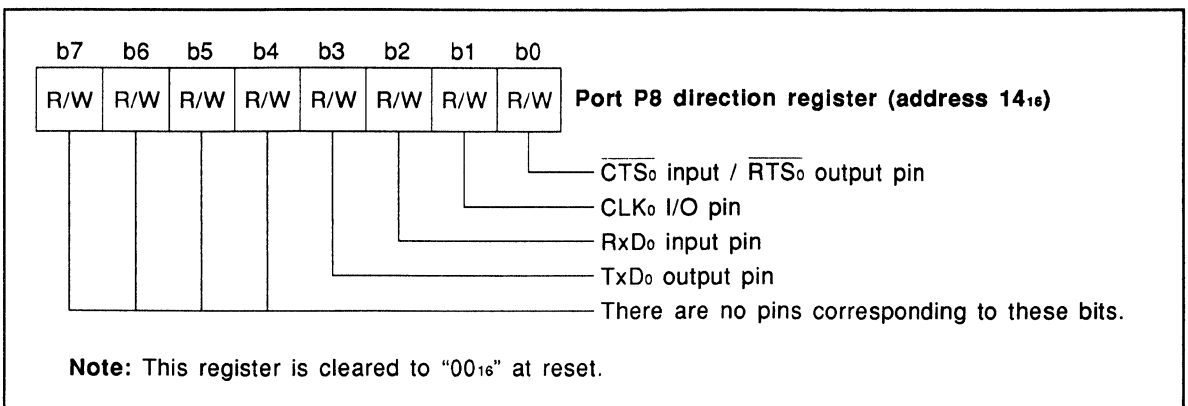


Fig. 8.2.21 Relationship between port P8 direction register and serial I/O pins

8.2.8 Watchdog timer, Hold, Ready function

Refer to "CHAPTER 2. FUNCTIONAL DESCRIPTION" for watchdog timer, Hold function and Ready function because their functions are identical to M37732S4FP.

8.2.9 Reset

The function and operation of reset are identical to M37732S4FP. Figure 8.2.22 shows the internal registers state at reset of M37730S2FP.

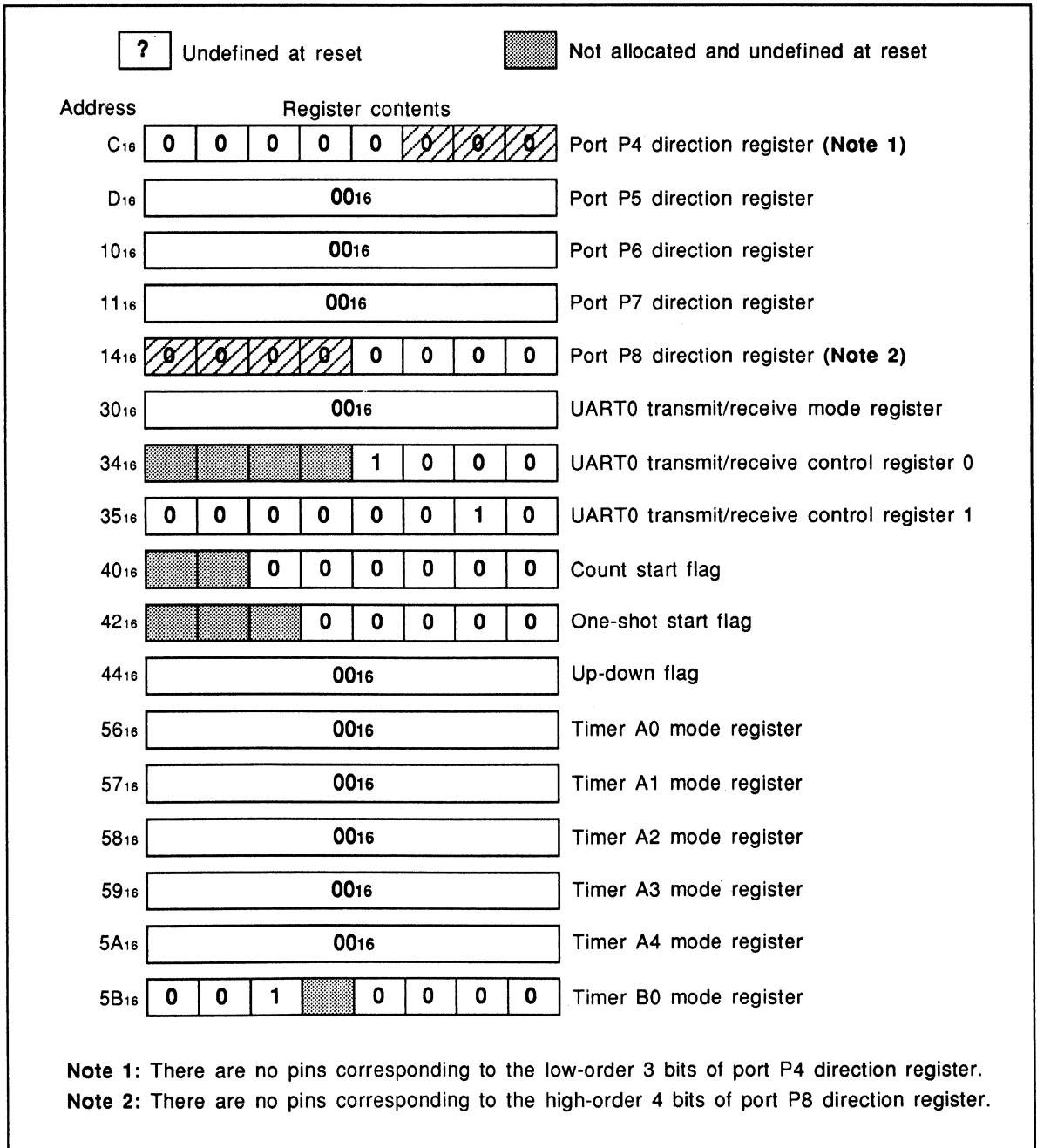


Fig. 8.2.22 Internal registers state at reset (1)

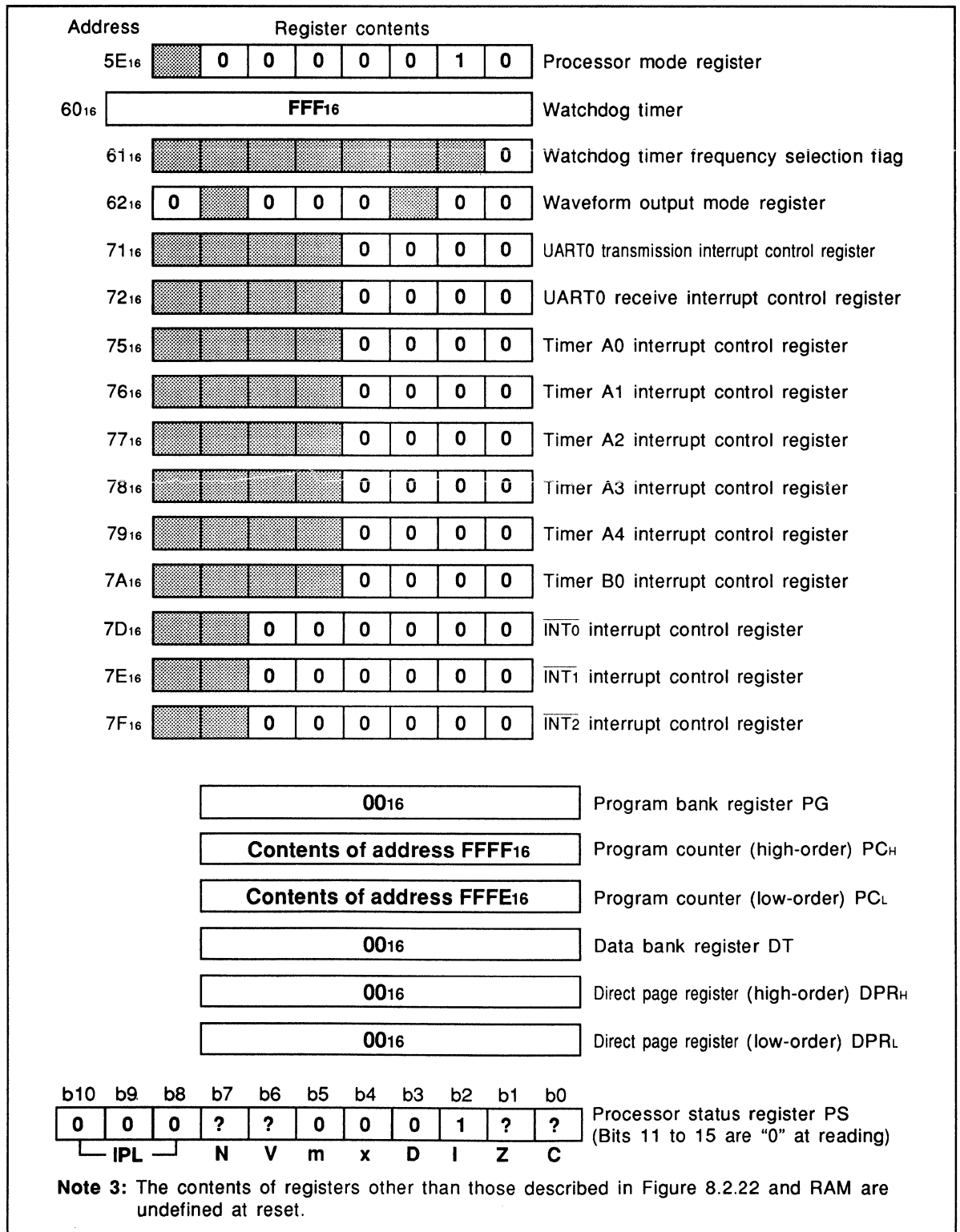


Fig. 8.2.22 Internal registers state at reset (2)

8.3 Electrical characteristics

8.3.1 Absolute maximum ratings

Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
V _{cc}	Supply voltage		-0.3 to 7	V
V _i	Input voltage RESET, CNV _{SS} , BYTE		-0.3 to 12	V
V _i	Input voltage A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₈₀ -P ₈₃ , HOLD, RDY		-0.3 to V _{cc} +0.3	V
V _o	Output voltage A ₀ -A ₇ , A ₈ /D ₈ -A ₂₃ /D ₇ , P ₄₃ -P ₄₇ , P ₅₀ -P ₅₇ , P ₆₀ -P ₆₇ , P ₈₀ -P ₈₃ , X _{OUT} , \bar{E} , ϕ_1 , HLDA, ALE, BHE, R/W		-0.3 to V _{cc} +0.3	V
P _d	Power dissipation	T _a =25°C	300 (Note)	mW
T _{opr}	Operating temperature		-20 to 85	°C
T _{stg}	Storage temperature		-40 to 150	°C

Note: In case of shrink DIP package, the rating of power dissipation is 1000mW.

M37730 GROUP

8.3 Electrical characteristics

8.3.2 Recommended operating conditions

Recommended operating conditions ($V_{CC}=5V\pm 10\%$, $T_a=-20$ to 85°C , unless otherwise noted)

Symbol	Parameter	Limits			Unit	
		Min.	Typ.	Max.		
V_{CC}	Supply voltage	4.5	5.0	5.5	V	
V_{SS}	Supply voltage		0		V	
V_{IH}	High-level input voltage	$P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, X_{IN}, \overline{\text{RESET}}, \overline{\text{CNV}}_{SS}, \overline{\text{BYTE}}, \overline{\text{HOLD}}, \overline{\text{RDY}}$	$0.8V_{CC}$		V_{CC}	V
V_{IH}	High-level input voltage	$A_8/D_8-A_{23}/D_7$	$0.5V_{CC}$		V_{CC}	V
V_{IL}	Low-level input voltage	$P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, X_{IN}, \overline{\text{RESET}}, \overline{\text{CNV}}_{SS}, \overline{\text{BYTE}}, \overline{\text{HOLD}}, \overline{\text{RDY}}$	0		$0.2V_{CC}$	V
V_{IL}	Low-level input voltage	$A_8/D_8-A_{23}/D_7$	0		$0.16V_{CC}$	V
$I_{OH}(\text{peak})$	High-level peak output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, \phi_1, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$			-10	mA
$I_{OH}(\text{avg})$	High-level average output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, \phi_1, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$			-5	mA
$I_{OL}(\text{peak})$	Low-level peak output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, \phi_1, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$			10	mA
$I_{OL}(\text{avg})$	Low-level average output current	$A_0-A_7, A_8/D_8-A_{23}/D_7, P4_3-P4_7, P5_0-P5_7, P6_0-P6_7, P8_0-P8_3, \phi_1, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$			5	mA
$f(X_{IN})$	External clock frequency input	M37730S2FP, M37730S2SP			8	MHz
		M37730S2AFP, M37730S2ASP			16	
		M37730S2BFP, M37730S2BSP			25	

Note 1: Average output current is the average value of a 100ms interval.

Note 2: The sum of $I_{OL}(\text{peak})$ for $A_0-A_7, A_8/D_8-A_{23}/D_7, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$, and port P8 must be 80mA or less, the sum of $I_{OH}(\text{peak})$ for $A_0-A_7, A_8/D_8-A_{23}/D_7, \overline{\text{HLDA}}, \overline{\text{ALE}}, \overline{\text{BHE}}, \overline{\text{R/W}}$, and port P8 must be 80mA or less, the sum of $I_{OL}(\text{peak})$ for ports P4, P5, P6, and ϕ_1 must be 80mA or less, and the sum of $I_{OH}(\text{peak})$ for ports P4, P5, P6, and ϕ_1 must be 80mA or less.

8.3.3 Electrical characteristics

M37730S2FP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=8MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_8_0-P_8_3$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.1 4.8			V
V_{OH}	High-level output voltage \bar{E}	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.4 4.8			V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_8_0-P_8_3$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$ $I_{OL}=2mA$			1.9 0.43	V
V_{OL}	Low-level output voltage \bar{E}	$I_{OL}=10mA$ $I_{OL}=2mA$			1.6 0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA_{0IN}-TA_{4IN}$, TB_{0IN} , INT_0-INT_2 , CTS, CLK ₀		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_8_0-P_8_3$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=5V$			5	μA
I_{IL}	Low-level input current $A_8/D_8-A_{23}/D_7$, $P_4_3-P_4_7$, $P_5_0-P_5_7$, $P_6_0-P_6_7$, $P_8_0-P_8_3$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=0V$			-5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during re- set.	$f(X_{IN})=8MHz$, square waveform $T_a=25^\circ C$ when clock is stopped. $T_a=85^\circ C$ when clock is stopped.	6	12	mA
					1	μA
					20	

M37730 GROUP

8.3 Electrical characteristics

M37730S2AFP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=16MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₈₀ –P ₈₃ , ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.1 4.8			V
V_{OH}	High-level output voltage \bar{E}	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.4 4.8			V
V_{OL}	Low-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₈₀ –P ₈₃ , ϕ_1 , HLDA, BHE, R/W	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A ₀ –A ₇ , A ₈ /D ₈ –A ₂₃ /D ₇ , ϕ_1 , HLDA, BHE, R/W	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$ $I_{OL}=2mA$			1.9 0.43	V
V_{OL}	Low-level output voltage \bar{E}	$I_{OL}=10mA$ $I_{OL}=2mA$			1.6 0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, TA _{0IN} –TA _{4IN} , TB _{0IN} , INT ₀ –INT ₂ , CTS ₀ , CLK ₀		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X _{IN}		0.1		0.3	V
I_{IH}	High-level input current A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₈₀ –P ₈₃ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	$V_i=5V$			5	μA
I_{IL}	Low-level input current A ₈ /D ₈ –A ₂₃ /D ₇ , P ₄₃ –P ₄₇ , P ₅₀ –P ₅₇ , P ₆₀ –P ₆₇ , P ₈₀ –P ₈₃ , X _{IN} , RESET, CNV _{SS} , BYTE, HOLD, RDY	$V_i=0V$			-5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V _{SS} during re- set.	$f(X_{IN})=16MHz$, square waveform T _a =25°C when clock is stopped. T _a =85°C when clock is stopped.	12	24	μA μA

M37730 GROUP

8.3 Electrical characteristics

M37730S2BFP

Electrical characteristics ($V_{CC}=5V$, $V_{SS}=0V$, $T_a=25^\circ C$, $f(X_{IN})=25MHz$, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{80}-P_{83}$, ϕ_1 , \overline{HLDA} , \overline{BHE} , R/\overline{W}	$I_{OH}=-10mA$	3			V
V_{OH}	High-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , \overline{HLDA} , \overline{BHE} , R/\overline{W}	$I_{OH}=-400\mu A$	4.7			V
V_{OH}	High-level output voltage ALE	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.1 4.8			V
V_{OH}	High-level output voltage \overline{E}	$I_{OH}=-10mA$ $I_{OH}=-400\mu A$	3.4 4.8			V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{80}-P_{83}$, ϕ_1 , \overline{HLDA} , \overline{BHE} , R/\overline{W}	$I_{OL}=10mA$			2	V
V_{OL}	Low-level output voltage A_0-A_7 , $A_8/D_8-A_{23}/D_7$, ϕ_1 , \overline{HLDA} , \overline{BHE} , R/\overline{W}	$I_{OL}=2mA$			0.45	V
V_{OL}	Low-level output voltage ALE	$I_{OL}=10mA$ $I_{OL}=2mA$			1.9 0.43	V
V_{OL}	Low-level output voltage \overline{E}	$I_{OL}=10mA$ $I_{OL}=2mA$			1.6 0.4	V
$V_{T+}-V_{T-}$	Hysteresis HOLD, RDY, $TA_{0IN}-TA_{4IN}$, TB_{0IN} , INT_0-INT_2 , CTS_0 , CLK_0		0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis RESET		0.2		0.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}		0.1		0.3	V
I_{IH}	High-level input current $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{80}-P_{83}$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=5V$			5	μA
I_{IL}	Low-level input current $A_8/D_8-A_{23}/D_7$, $P_{43}-P_{47}$, $P_{50}-P_{57}$, $P_{60}-P_{67}$, $P_{80}-P_{83}$, X_{IN} , RESET, CNV_{SS} , BYTE, HOLD, RDY	$V_I=0V$			-5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during re- set.	$f(X_{IN})=25MHz$, square waveform $T_a=25^\circ C$ when clock is stopped.	19	38	mA
			$T_a=85^\circ C$ when clock is stopped.		20	μA

8.3.4 Timing requirements

Timing requirements ($V_{CC}=5V\pm 10\%$, $V_{SS}=0V$, $T_a=25^\circ C$, unless otherwise noted)

External clock input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_c	External clock input cycle time	125		62		40		ns
$t_{w(H)}$	External clock input high-level pulse width	50		25		15		ns
$t_{w(L)}$	External clock input low-level pulse width	50		25		15		ns
t_r	External clock rise time		20		10		8	ns
t_f	External clock fall time		20		10		8	ns

Microprocessor mode

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{su(DH-E)}$	Data high-order input setup time	60		45		30		ns
$t_{su(DL-E)}$	Data low-order input setup time	60		45		30		ns
$t_{su(P4D-E)}$	Port P4 input setup time	200		100		60		ns
$t_{su(P5D-E)}$	Port P5 input setup time	200		100		60		ns
$t_{su(P6D-E)}$	Port P6 input setup time	200		100		60		ns
$t_{su(P7D-E)}$	Port P7 input setup time	200		100		60		ns
$t_{su(P8D-E)}$	Port P8 input setup time	200		100		60		ns
$t_{su(RDY-\phi)}$	RDY input setup time	70		60		55		ns
$t_{su(HOLD-\phi)}$	HOLD input setup time	70		60		55		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		0		0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		0		0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		0		0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		0		0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		0		0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		0		0		ns
$t_{h(\phi-RDY)}$	RDY input hold time	0		0		0		ns
$t_{h(\phi-HOLD)}$	HOLD input hold time	0		0		0		ns

Timer A input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	250		125		80		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	125		62		40		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	125		62		40		ns

Timer A input (gating input in timer mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	1000		500		320		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	500		250		160		ns

Timer A input (external trigger input in one-shot pulse mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TA)}$	TA _{IN} input cycle time	500		250		160		ns
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		125		80		ns

Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(TAH)}$	TA _{IN} input high-level pulse width	250		125		80		ns
$t_{w(TAL)}$	TA _{IN} input low-level pulse width	250		125		80		ns

Timer A input (up-down input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(UP)}$	TA _{IOUT} input cycle time	5000		2500		2000		ns
$t_{w(UPH)}$	TA _{IOUT} input high-level pulse width	2500		1250		1000		ns
$t_{w(UPL)}$	TA _{IOUT} input low-level pulse width	2500		1250		1000		ns
$t_{su(UP-TIN)}$	TA _{IOUT} input setup time	1000		500		400		ns
$t_{h(TIN-UP)}$	TA _{IOUT} input hold time	1000		500		400		ns

Timer B input (count input in event counter mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB _{0IN} input cycle time (one edge count)	250		125		80		ns
$t_{w(TBH)}$	TB _{0IN} input high-level pulse width (one edge count)	125		62		40		ns
$t_{w(TBL)}$	TB _{0IN} input low-level pulse width (one edge count)	125		62		40		ns
$t_{c(TB)}$	TB _{0IN} input cycle time (both edges count)	500		250		160		ns
$t_{w(TBH)}$	TB _{0IN} input high-level pulse width (both edges count)	250		125		80		ns
$t_{w(TBL)}$	TB _{0IN} input low-level pulse width (both edges count)	250		125		80		ns

M37730 GROUP

8.3 Electrical characteristics

Timer B input (pulse period measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB0 _{IN} input cycle time	1000		500		320		ns
$t_{w(TBH)}$	TB0 _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TBL)}$	TB0 _{IN} input low-level pulse width	500		250		160		ns

Timer B input (pulse width measurement mode)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(TB)}$	TB0 _{IN} input cycle time	1000		500		320		ns
$t_{w(TBH)}$	TB0 _{IN} input high-level pulse width	500		250		160		ns
$t_{w(TBL)}$	TB0 _{IN} input low-level pulse width	500		250		160		ns

Serial I/O

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{c(CLK)}$	CLK ₀ input cycle time	500		250		200		ns
$t_{w(CLKH)}$	CLK ₀ input high-level pulse width	250		125		100		ns
$t_{w(CLKL)}$	CLK ₀ input low-level pulse width	250		125		100		ns
$t_{d(C-Q)}$	TxD ₀ output delay time		150		90		80	ns
$t_{h(C-Q)}$	TxD ₀ hold time	30		30		30		ns
$t_{su(D-C)}$	RxD ₀ input setup time	60		30		20		ns
$t_{h(C-D)}$	RxD ₀ input hold time	90		90		90		ns

External interrupt INT_i input

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{w(INH)}$	INT _i input high-level pulse width	250		250		250		ns
$t_{w(INL)}$	INT _i input low-level pulse width	250		250		250		ns

8.3.5 Switching characteristics

Switching characteristics ($V_{CC}=5V\pm 10\%$, $V_{SS}=0V$, $T_a=25^\circ C$, unless otherwise noted)

Microprocessor mode (when wait bit is "1")

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")		110		70		45	ns
$t_{p{xz}(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time	100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time	80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70		45	ns
$t_{p{xz}(E-DLZ)}$	Floating start delay time		5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		5		ns
$t_{d(\phi-HLDA)}$	HLDA output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_w(ALE)$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	BHE output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/W output delay time	100		30		20		ns
$t_{d(E-\phi)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_h(E-AL)$	Address low-order hold time	50		25		18		ns
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")	9		9		9		ns
$t_h(E-DHQ)$	Data high-order hold time (BYTE="L")	50		25		18		ns
$t_{p{zx}(E-DHZ)}$	Floating release delay time (BYTE="L")	95		36		18		ns
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")	50		25		18		ns
$t_h(ALE-AH)$	Address high-order hold time	9		9		9		ns
$t_h(E-DLQ)$	Data low-order hold time	50		25		18		ns
$t_{p{zx}(E-DLZ)}$	Floating release delay time	95		36		18		ns
$t_h(E-BHE)$	BHE hold time	18		18		18		ns
$t_h(E-R/W)$	R/W hold time	18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100		80	ns
$t_w(EL)$	\bar{E} pulse width	220		95		50		ns

Note: Test conditions are shown in Figure 8.3.1.

M37730 GROUP

8.3 Electrical characteristics

Microprocessor mode (when wait bit = "0", and external memory area is accessed)

Symbol	Parameter	Limits						Unit
		8MHz		16MHz		25MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	100		30		12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")		110		70		45	ns
$t_{p{xz}(E-DHZ)}$	Floating start delay time (BYTE="L")		5		5		5	ns
$t_{d(AM-E)}$	Address middle-order output delay time	100		30		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time	80		24		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time		110		70		45	ns
$t_{p{xz}(E-DLZ)}$	Floating start delay time		5		5		5	ns
$t_{d(AH-E)}$	Address high-order output delay time	100		30		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time	80		24		5		ns
$t_{d(\phi-HLDA)}$	HLDA output delay time		100		50		50	ns
$t_{d(ALE-E)}$	ALE output delay time	4		4		4		ns
$t_w(ALE)$	ALE pulse width	90		35		22		ns
$t_{d(BHE-E)}$	BHE output delay time	100		30		20		ns
$t_{d(R/W-E)}$	R/W output delay time	100		30		20		ns
$t_{d(E-\phi)}$	ϕ_1 output delay time	0	30	0	20	0	18	ns
$t_h(E-AL)$	Address low-order hold time	50		25		18		ns
$t_h(ALE-AM)$	Address middle-order hold time (BYTE="L")	9		9		9		ns
$t_h(E-DHQ)$	Data high-order hold time (BYTE="L")	50		25		18		ns
$t_{p{zx}(E-DHZ)}$	Floating release delay time (BYTE="L")	95		36		18		ns
$t_h(E-AM)$	Address middle-order hold time (BYTE="H")	50		25		18		ns
$t_h(ALE-AH)$	Address high-order hold time	9		9		9		ns
$t_h(E-DLQ)$	Data low-order hold time	50		25		18		ns
$t_{p{zx}(E-DLZ)}$	Floating release delay time	95		36		18		ns
$t_h(E-BHE)$	BHE hold time	18		18		18		ns
$t_h(E-R/W)$	R/W hold time	18		18		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time		200		100		80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time		200		100		80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time		200		100		80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time		200		100		80	ns
$t_w(EL)$	E pulse width	470		220		130		ns

Note: Test conditions are shown in Figure 8.3.1.

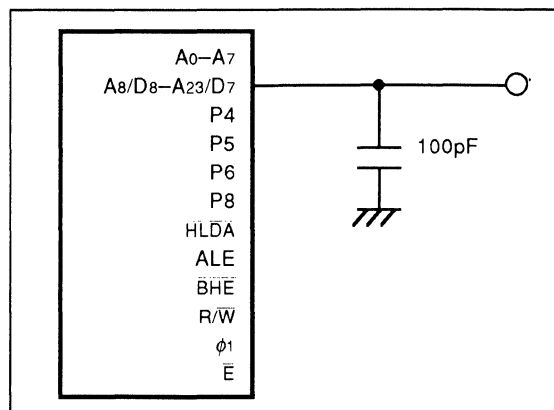


Fig. 8.3.1 Testing circuit for ports

8.3.6 Equations for calculating the parameters

Table 8.3.1 shows the equations for calculating the following parameters.

$t_w(\text{ALE})$ ALE pulse width	$t_d(\text{R/W-E})$ R/W output delay time
$t_w(\text{EL})$ \bar{E} pulse width	$t_{pzx}(\text{E-DLZ/DHZ})$ Floating release delay time
$t_d(\text{AL/AM/AH-E})$ Address output delay time	$t_h(\text{E-AL/AM})$ Address hold time
$t_d(\text{AM/AH-ALE})$ Address output delay time	$t_h(\text{E-DLQ/DHQ})$ Data hold time
$t_d(\text{BHE-E})$ $\overline{\text{BHE}}$ output delay time	

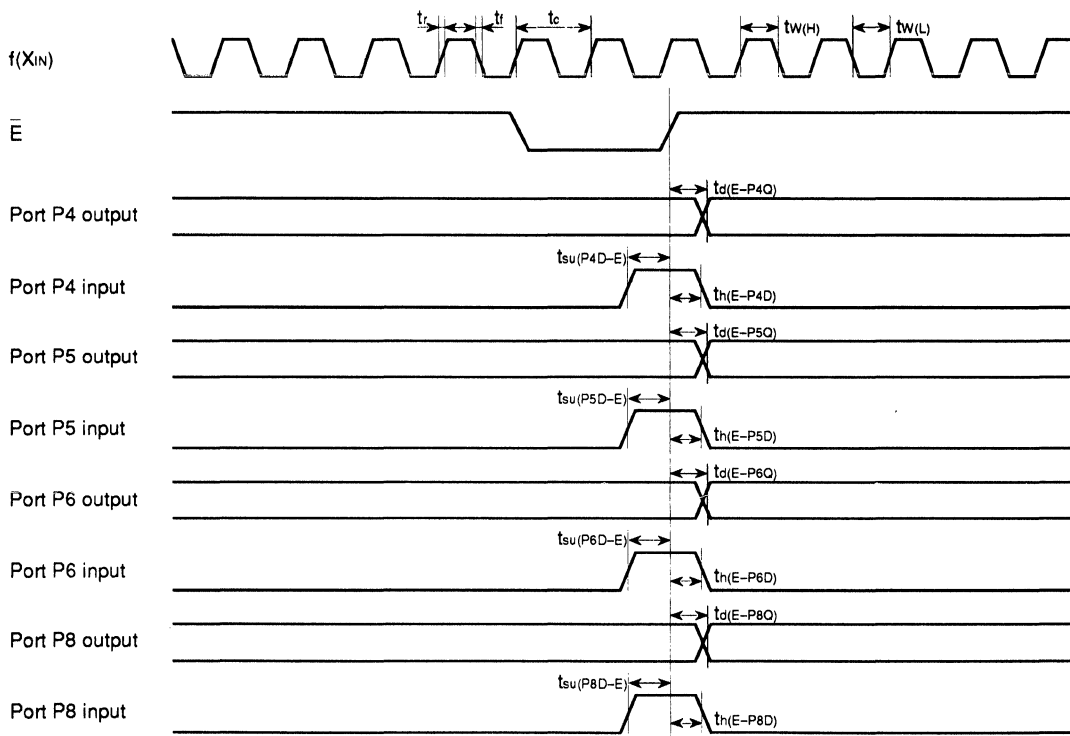
Table 8.3.1 Equations for calculating parameters

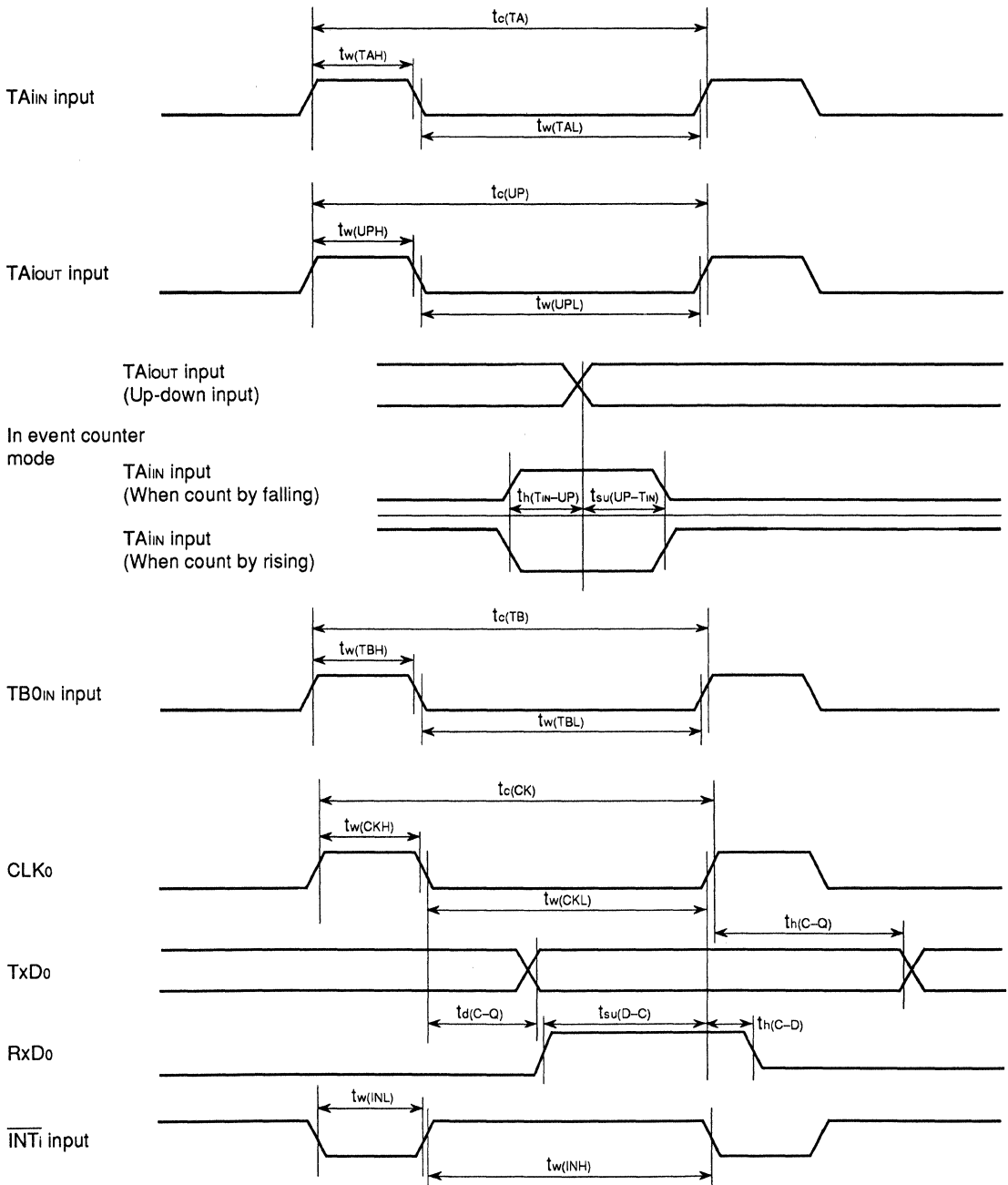
	M37730S2FP/SP	M37730S2AFP/SP	M37730S2BFP/SP
$f(\text{XIN})$	$f(\text{XIN}) \leq 8\text{MHz}$	$8\text{MHz} < f(\text{XIN}) \leq 16\text{MHz}$	$16\text{MHz} < f(\text{XIN}) \leq 25\text{MHz}$
$t_w(\text{ALE})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 35$	$\frac{1 \times 10^9}{f(\text{XIN})} - 27.5$	$\frac{1 \times 10^9}{f(\text{XIN})} - 18$
$t_w(\text{EL})$ Wait bit = "1"	$\frac{2 \times 10^9}{f(\text{XIN})} - 30$		
$t_w(\text{EL})$ Wait bit = "0"	$\frac{4 \times 10^9}{f(\text{XIN})} - 30$		
$t_d(\text{AL/AM/AH-E})$ $t_d(\text{BHE-E})$ $t_d(\text{R/W-E})$	$100 + \frac{1 \times 10^9}{f(\text{XIN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(\text{XIN})} - 75$	$12 + \frac{1 \times 10^9}{f(\text{XIN})} - 40$
$t_{pzx}(\text{E-DLZ/DHZ})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 30$	$\frac{1 \times 10^9}{f(\text{XIN})} - 26$	$\frac{1 \times 10^9}{f(\text{XIN})} - 22$
$t_h(\text{E-AL/AM})$ $t_h(\text{E-DLQ/DHQ})$	$\frac{1 \times 10^9}{2 \times f(\text{XIN})} - 12.5$	$\frac{1 \times 10^9}{2 \times f(\text{XIN})} - 6.25$	$\frac{1 \times 10^9}{2 \times f(\text{XIN})} - 2$
$t_d(\text{AM/AH-ALE})$	$\frac{1 \times 10^9}{f(\text{XIN})} - 45$	$\frac{1 \times 10^9}{f(\text{XIN})} - 38.5$	$\frac{1 \times 10^9}{f(\text{XIN})} - 35$

Unit : ns

8.3.7 Timing diagram

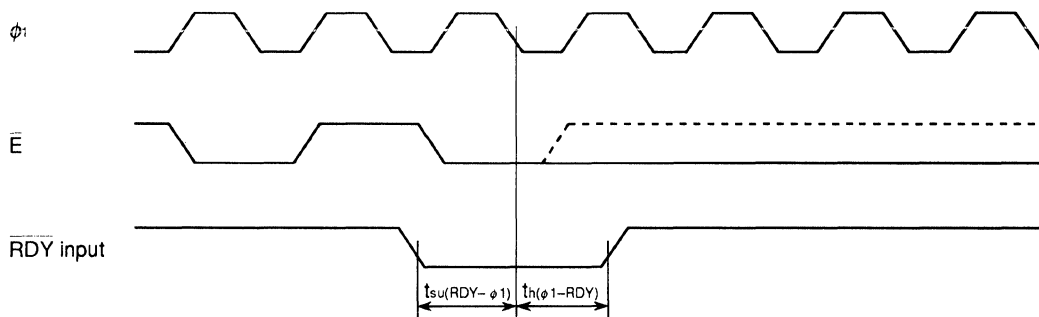
Timing diagram



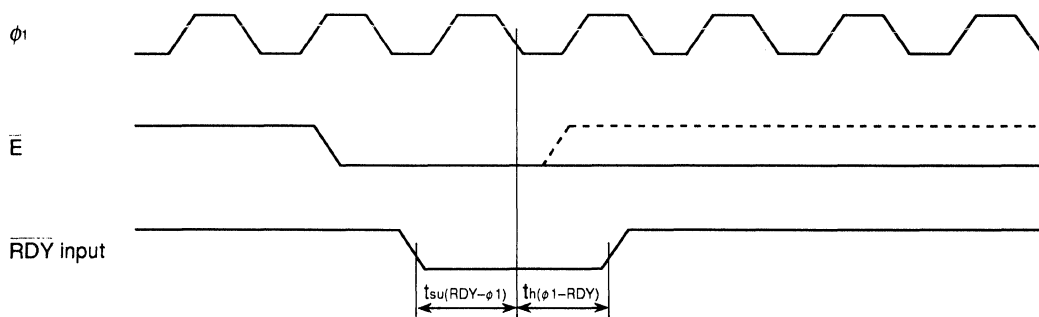


Microprocessor mode

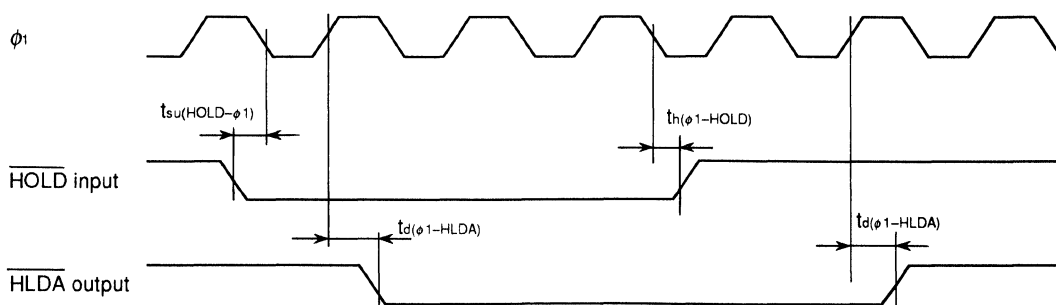
(When wait bit="1")



(When wait bit="0")



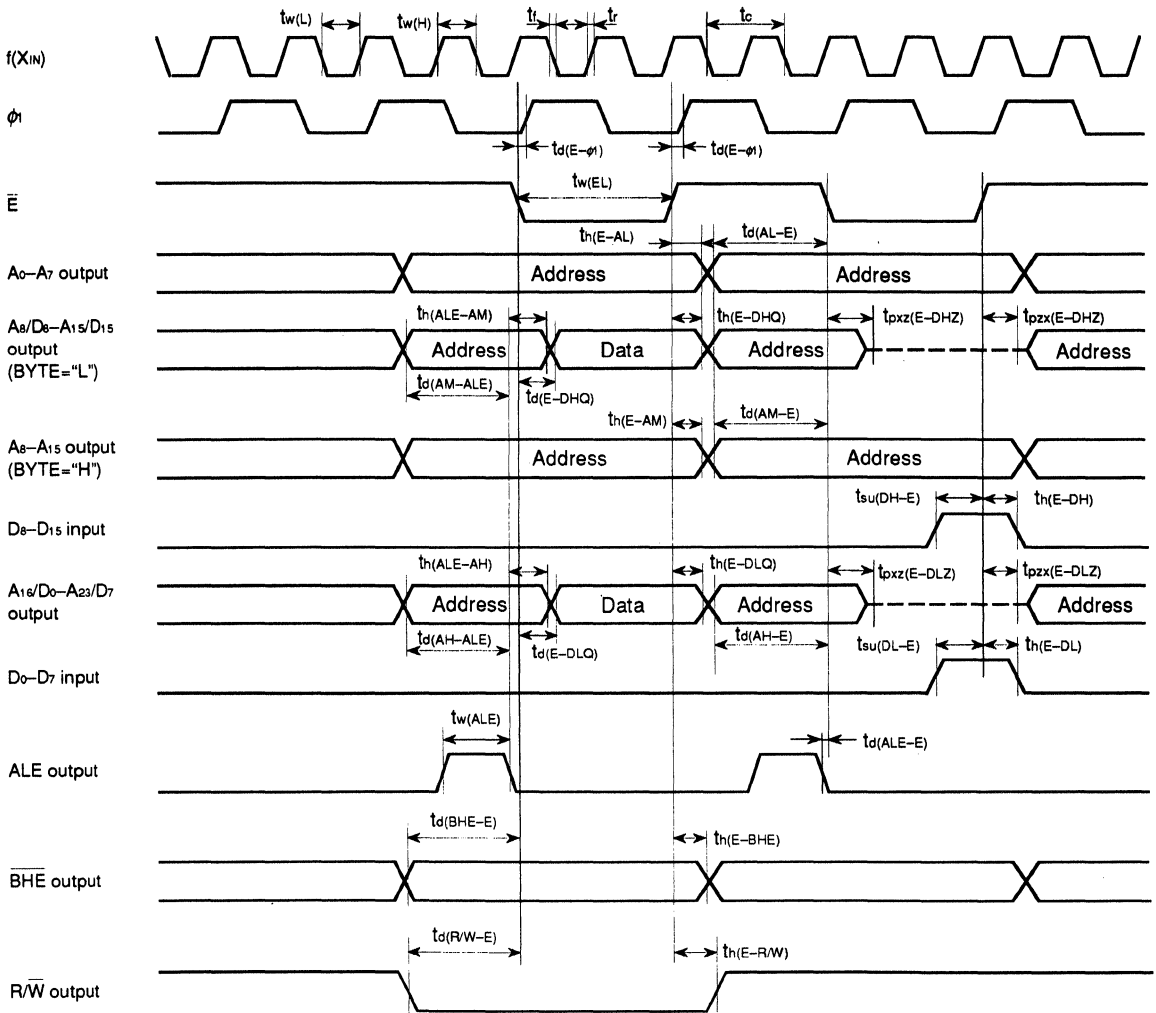
(When wait bit="1" or "0" in common)



Test conditions

- Vcc=5V±10%
- Input timing voltage : V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V

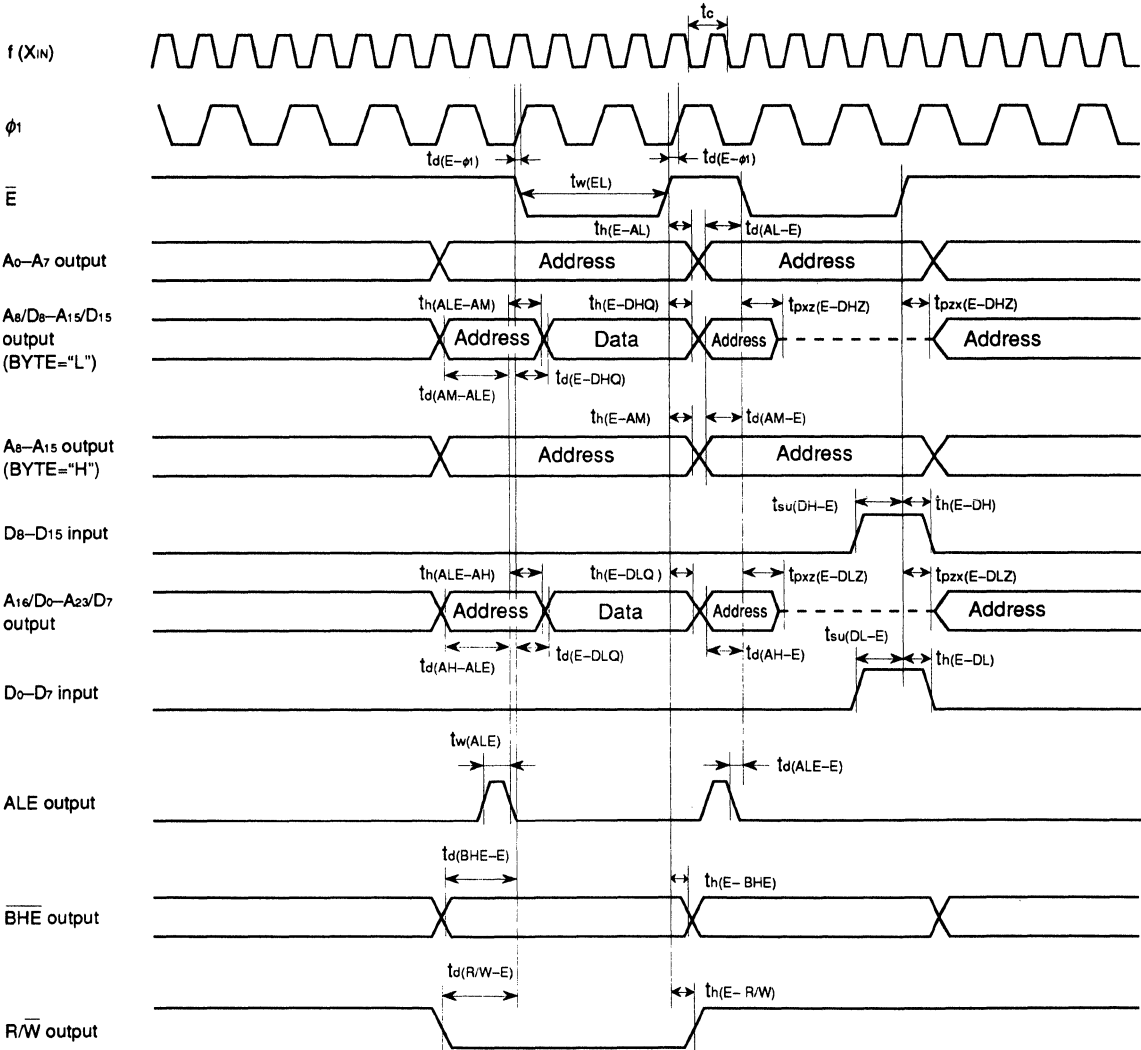
Microprocessor mode (When wait bit="1")



Test conditions

- $V_{CC}=5V \pm 10\%$
- Output timing voltage : $V_{OL}=0.8V$, $V_{OH}=2.0V$
- D0-D15 input : $V_{IL}=0.8V$, $V_{IH}=2.5V$

Microprocessor mode (When wait bit="0", and external memory area is accessed)



Test conditions

- $V_{CC}=5V\pm 10\%$
- Output timing voltage : $V_{OL}=0.8V, V_{OH}=2.0V$
- D_0-D_{15} input : $V_{IL}=0.8V, V_{IH}=2.5V$

MEMORANDUM



CHAPTER 9

APPLICATION

- 9.1 Memory connection
- 9.2 I/O expansion
- 9.3 Low supply voltage version applications
- 9.4 Power saving
- 9.5 Pulse motor control application
- 9.6 Application system examples
- 9.7 Program examples

9.1 Memory connection

The M37732 group's external memory connections are described below. The low supply voltage version (M37732S4LGP/HP) is described in section "9.3 Low supply voltage version applications". This section describes the M37732 group. However, the same memory connections are available with the M37730 group. The application examples described in this chapter may not be applicable to all types in the M37732 group depending on the settings. Therefore the applicable type will simply be referred to as "M37732" in this chapter.

9.1.1 Memory connection model

The memory can be connected by using the external bus width selection pin (BYTE pin). Four external memory connection models shown in Table 9.1.1 are available.

(1) Minimum model

External memory area within the 64K-byte memory space is accessed using an 8-bit external data bus width. No external address latch is necessary. This is the most cost effective connection model for externally adding an 8-bit element.

(2) Medium model A

Memory space beyond 64K bytes is accessed using an 8-bit external data bus width as same as the minimum model. An n-bit ($n \leq 8$) address latch is required to latch the address from the high-order 8 bits ($A_{23}-A_{16}$) of the address bus multiplexed with the data bus, but the accessible memory space is expanded up to 16M bytes.

(3) Medium model B

This connection model limits memory space to within 64K bytes, but provides optimum speed. The external data bus width is 16 bits, and connected memory area can be accessed as fast as internal area if no wait (one-wait mode nor Ready function due to the \overline{RDY} input) is used. This connection model requires an 8-bit address latch to latch the address because the middle-order 8 bits ($A_{15}-A_8$) of the address bus are multiplexed with the data bus.

(4) Maximum model

This connection model uses 16-bit external data bus width to access up to 16M-byte memory space. An 8-bit latch to latch the middle-order 8 bits ($A_{15}-A_8$) of the address bus and an n-bit ($n \leq 8$) latch to latch n bits of the high-order 8 bits ($A_{23}-A_{16}$) are required.

APPLICATION

9.1 Memory connection

Table 9.1.1 Memory connection model

Access space External bus width	Within 64K bytes	Beyond 64K bytes
8-bit BYTE="H"	<p>Memory connection model Minimum model</p>	<p>Memory connection model Medium model A</p>
16-bit BYTE="L"	<p>Memory connection model Medium model B</p>	<p>Memory connection model Maximum model</p>

M37732 group has 37 I/O ports. However, port expansion must be performed together with memory connection for applications that use many ports. Port expansion is described in section "9.2 I/O expansion". Be sure to low the Vss line impedance of the M37732 group during memory connection and I/O expansion because the address bus is used at 24 bits (refer to section "Appendix 7. System development precautions").

APPLICATION

9.1 Memory connection

9.1.2 Memory access time calculation

This section describes how to calculate the memory access time necessary to satisfy the memory connection timing requirements.

Figure 9.1.1 shows the bus timing.

The memory access time $t_{a(AD)}$ is obtained from the following equation ①. However, in the minimum model, there is no need to consider the address latch delay time in equation ① because the ALE signal is not used.

$$t_{a(AD)} = t_{d(AL/AM/AH-E)} + t_{w(EL)} - t_{su(DL/DH-E)} - \{ \text{address decode time} + \text{address latch delay time}^* \} \dots \text{①}$$

* Address latch delay time is not necessary for minimum model.

$t_{d(AL/AM/AH-E)}$ in equation ① represents either $t_{d(AL-E)}$, $t_{d(AM-E)}$ or $t_{d(AH-E)}$ and $t_{su(DL/DH-E)}$ represents either $t_{su(DL-E)}$ or $t_{su(DH-E)}$.

$t_{w(EL)}$ and $t_{su(D)}$ in equation ① are parameters that depend on the external clock input frequency and are calculated by the equations shown in Table 9.1.2. $t_{su(DL/DH-E)}$ is a constant that depends on the external clock input frequency (8, 16, or 25MHz). Address decode time is the time required to decode the address and make the chip select signal valid.

The data setup time $t_{su(D)}$ for write is obtained from the following equation ②.

$$t_{su(D)} = t_{w(EL)} - t_{d(E-DLQ/DHQ)} \dots \text{②}$$

$t_{d(E-DLQ/DHQ)}$ in equation ② represents either $t_{d(E-DLQ)}$ or $t_{d(E-DHQ)}$.

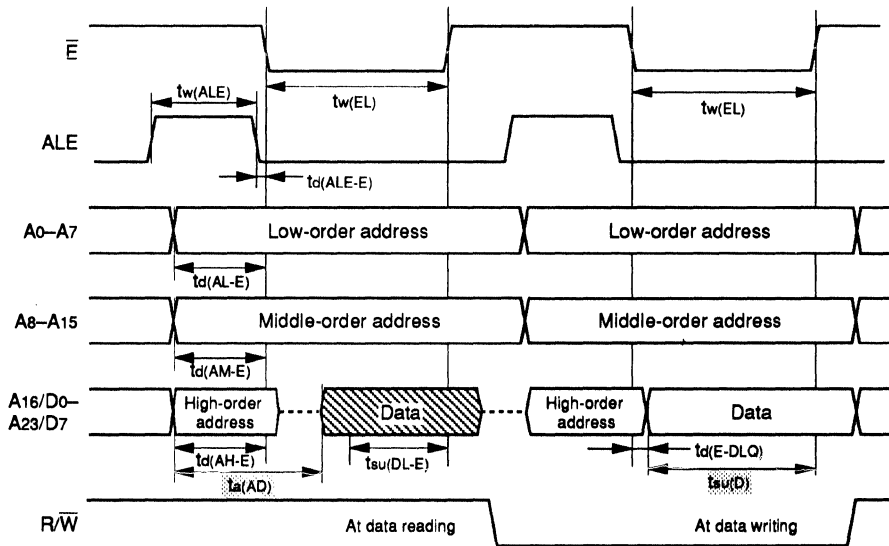
If the setup time requested by the device do not satisfy these values, waits must be inserted in the bus cycle with the wait bit or \overline{RDY} input.

Table 9.1.2 Parameter equation and constants depending on the external clock input frequency

Parameter	Type	8MHz version M37732S4FP	16MHz version M37732S4AFP	25MHz version M37732S4BFP
$f(X_{IN})$		$f(X_{IN}) \leq 8\text{MHz}$	$8\text{MHz} < f(X_{IN}) \leq 16\text{MHz}$	$16\text{MHz} < f(X_{IN}) \leq 25\text{MHz}$
$t_{d(AL-E)}$		$100 + \frac{1 \times 10^9}{f(X_{IN})} - 125$	$30 + \frac{1.2 \times 10^9}{f(X_{IN})} - 75$	$12 + \frac{1 \times 10^9}{f(X_{IN})} - 40$
$t_{d(AM-E)}$				
$t_{d(AH-E)}$				
$t_{w(EL)}$		$\frac{2 \times 10^9}{f(X_{IN})} - 30$	←	←
Under the one-wait mode (the wait bit = "0")		$\frac{4 \times 10^9}{f(X_{IN})} - 30$	←	←
$t_{su(DL-E)}$		60	45	30
$t_{su(DH-E)}$				
$t_{d(E-DLQ)}$		110	70	45
$t_{d(E-DHQ)}$				

(Unit : ns)

BYTE="H" (when external bus width is 8 bits)



BYTE="L" (when external bus width is 16 bits)

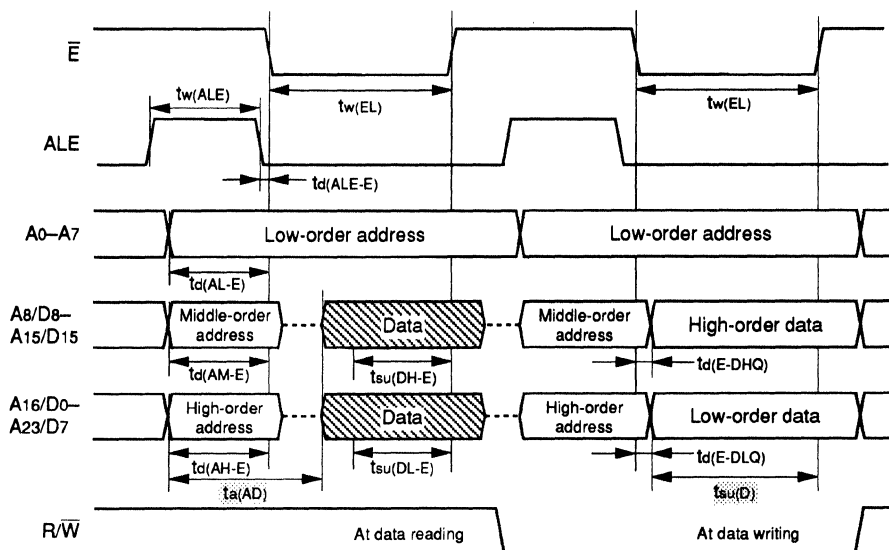


Fig. 9.1.1 Bus timing diagram

APPLICATION

9.1 Memory connection

Figure 9.1.2 shows the relationship between the memory access time $t_{a(AD)}$ and external clock input frequency $f(X_{IN})$. The graph in Figure 9.1.2 shows the memory access time ignoring the address decode time and address latch delay time in equation ①. Therefore, the actual memory access time is the value of the graph minus the address decode time and address latch delay time.

Figure 9.1.3 shows the relationship between the data setup time $t_{su(D)}$ for write and the external clock input frequency.

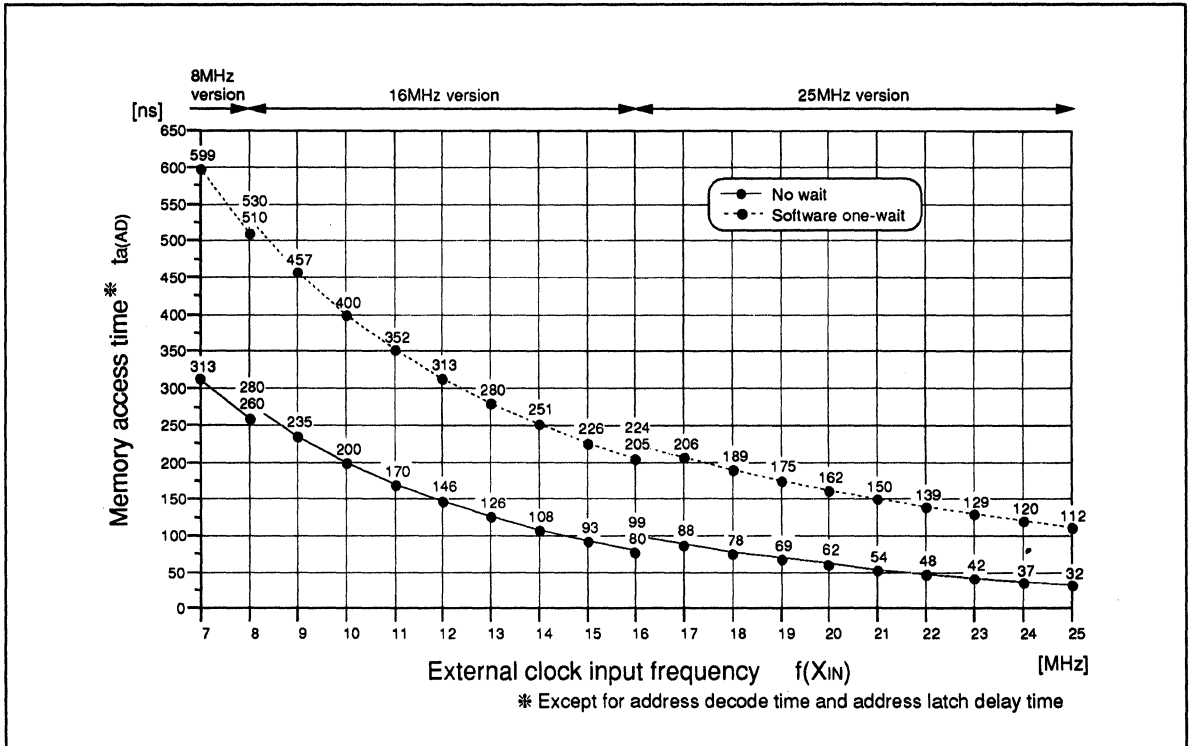


Fig. 9.1.2 Relationship between memory access time and external clock input frequency

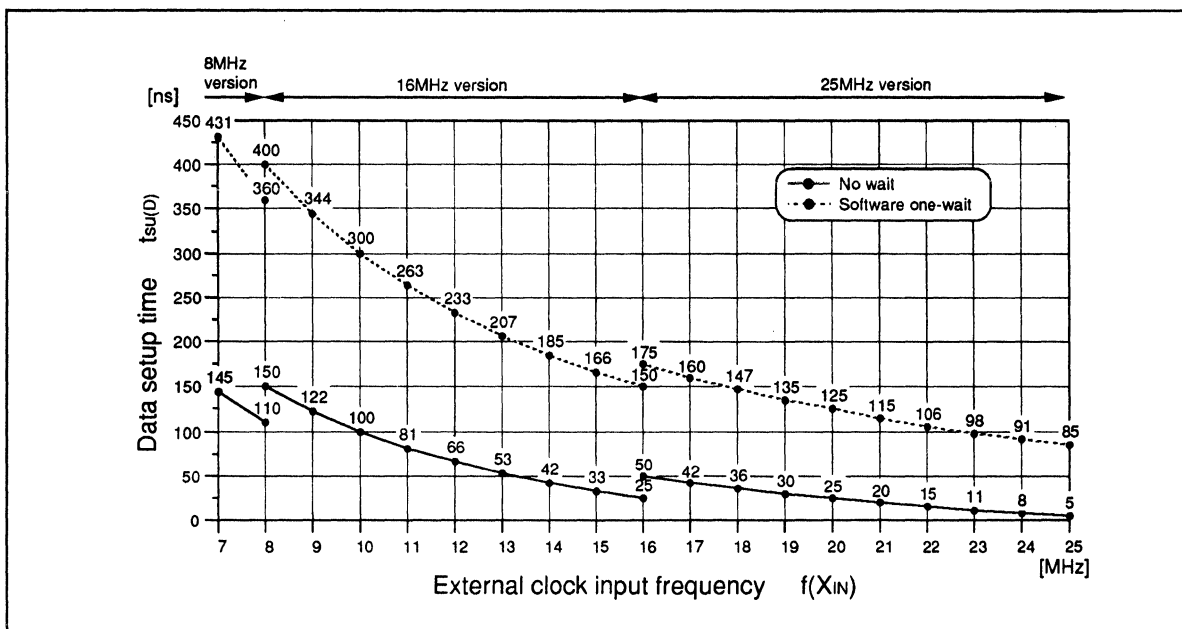


Fig. 9.1.3 Relationship between data setup time for write and external clock input frequency

9.1.3 Memory connection precautions

Figure 9.1.4 shows the memory data read timing. When setting \bar{E} signal to "L" level and reading the data on the bus, the "M37732" has an address hold time ($t_{pxz(E-DHZ/DLZ)}$) of up to 5ns. When the \bar{E} signal becomes "H" level and data on the bus is read, at minimum the floating time ($t_{pxz(E-DHZ/DLZ)}$) shown in Table 9.1.3 is reserved before the next address is output. Therefore, when using devices that output data on the data bus within 5ns from the fall of the \bar{E} signal ($t_{en(OE)} \leq 5ns$) or output data on the data bus for more than $t_{pxz(E-DHZ/DLZ)}$ at the rise of the \bar{E} signal, considerations must be made to prevent bus contention between the address output by the "M37732" and the data output by the device.

When using devices with $t_{en(OE)}$ that does not satisfy $t_{pxz(E-DHZ/DLZ)}$, generate the device read signal \overline{OE} with only the leading edge of the fall of the \bar{E} signal delayed for few nanoseconds.

When using devices with t_{DF} and $t_{dis(OE)}$ that do not satisfy $t_{pxz(E-DHZ/DLZ)}$, delete the data output by the device using a bus buffer for example. Figures 9.1.5 and 9.1.6 show examples of using a bus buffer. Table 9.1.4 shows Mitsubishi memories that can be connected to the "M37732" without bus buffer. When requesting memory with specifications shown in Table 9.1.4, specify "t_{DF} 15ns version, microcomputer' kit".

Table 9.1.3 Address hold time and data floating time

Parameter \ Type	8MHz version M37732S4FP	16MHz version M37732S4AFP	25MHz version M37732S4BFP
$f(X_{IN})$	$f(X_{IN}) \leq 8MHz$	$8MHz < f(X_{IN}) \leq 16MHz$	$16MHz < f(X_{IN}) \leq 25MHz$
$t_{pxz(E-DHZ)}$	5	5	5
$t_{pxz(E-DLZ)}$	$\frac{1 \times 10^9}{f(X_{IN})} - 30$	$\frac{1 \times 10^9}{f(X_{IN})} - 26$	$\frac{1 \times 10^9}{f(X_{IN})} - 22$

(Unit : ns)

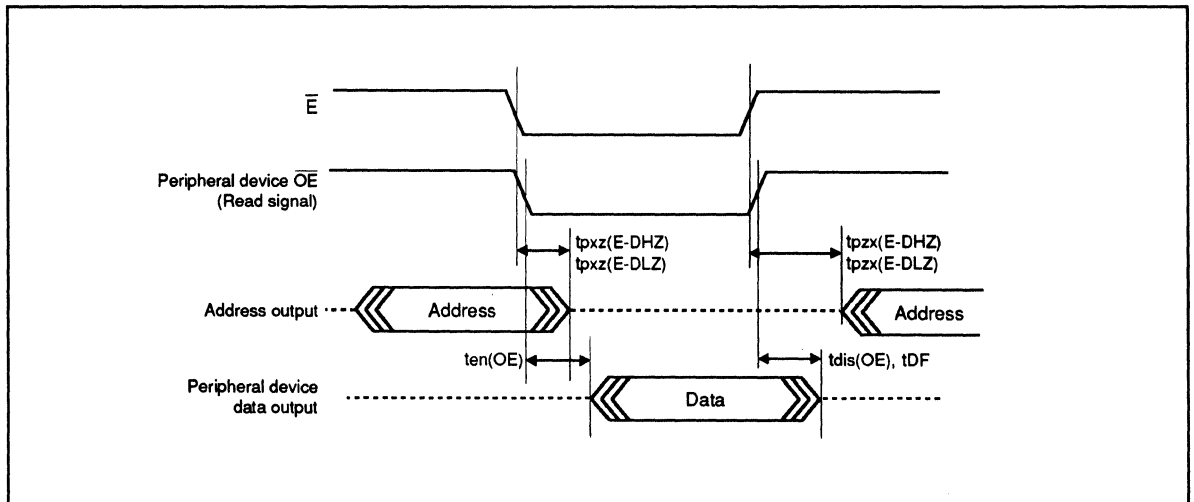


Fig. 9.1.4 Memory data read timing

APPLICATION

9.1 Memory connection

Table 9.1.4 Usable memory without bus buffer

Memory type	Type name	t_{DF}/t_{dis} (OE)	Conditions
EPROM	M5M27C256AK-85/-10/-12/-15 or equivalent OTP component M5M27C512AK-10/-12/-15 or equivalent OTP component M5M27C100K-12/-15 M5M27C101K-12/-15 M5M27C102K-12/-15	15ns	$f(X_{IN}) \leq 20\text{MHz}$ (Note)
SRAM	M5M5256BP-70/-85/-10/-12/-15 or equivalent L, LL types M5M5178P-35/-45/-55	15ns	$f(X_{IN}) \leq 20\text{MHz}$ (Note)

Note: M74F32 or equivalent component is required for read signal generation when using at 16MHz frequency or greater.

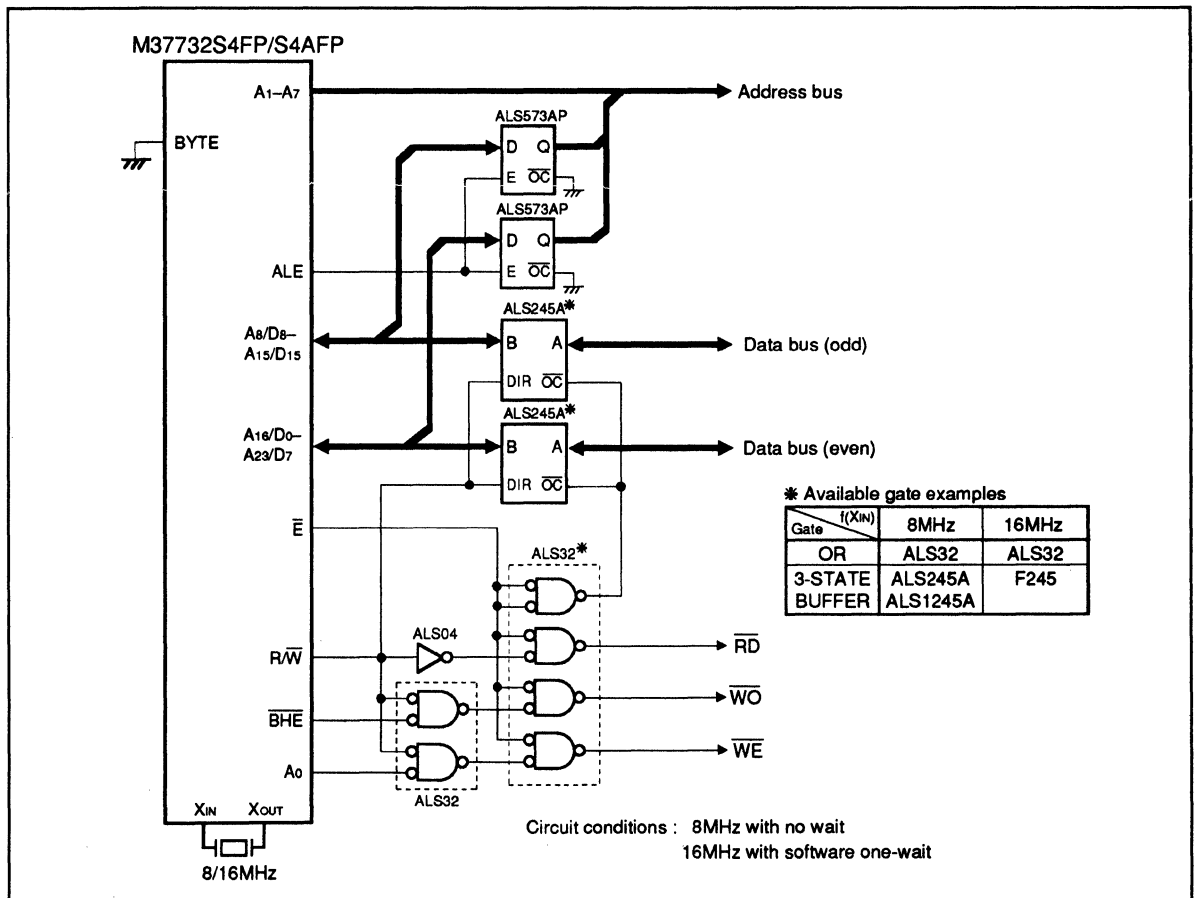


Fig. 9.1.5 Bus buffer usage example (1)

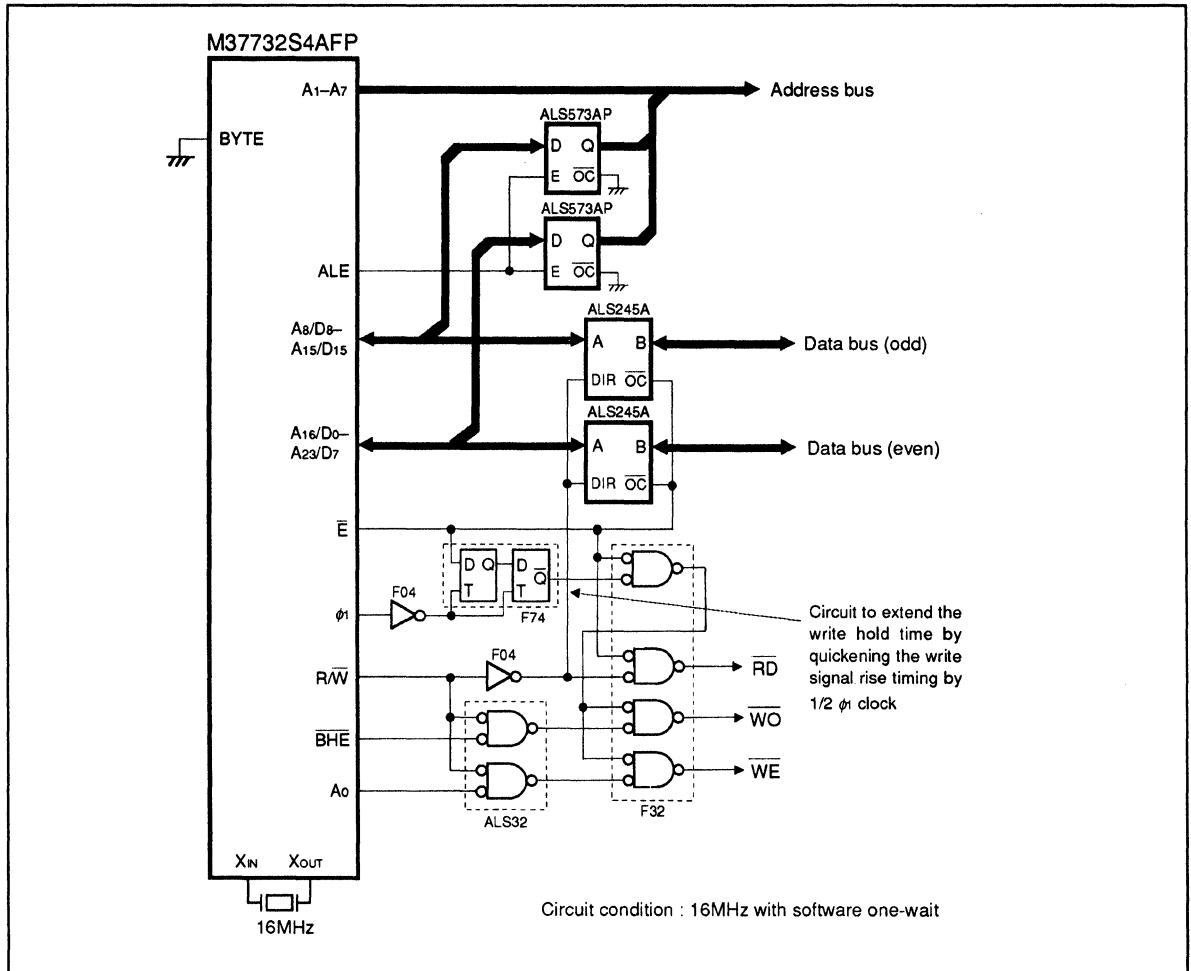


Fig. 9.1.6 Bus buffer usage example (2) (connection to device that requests long hold time during write)

9.1.4 Connection to devices that request long access time

Waits can be inserted in the bus to extend the access time when connecting to devices that request long access time. Wait can be inserted by software one-wait method which sets the wait bit of the processor mode register bit 2 to "0" and inserts waits for 1 cycle of the clock ϕ_1 during the "L" level cycle of the \bar{E} signal, or by hardware using \overline{RDY} input to insert any number of wait cycles. Figure 9.1.7 shows an Ready generating circuit example for one-wait insertion by hardware. Hardware wait by \overline{RDY} input can generate wait for internal area access as well. Therefore, this circuit example uses a chip select signal to specify the area for inserting a wait. If the external clock input frequency is 14.9MHz or greater, this circuit cannot be used because the setup time of the \overline{RDY} input at the fall of the clock ϕ_1 is insufficient. Refer to section "9.1.5 (5) Memory connection example at 25MHz using software one-wait and Ready function" for the use of Ready function when \overline{RDY} input setup time is insufficient.

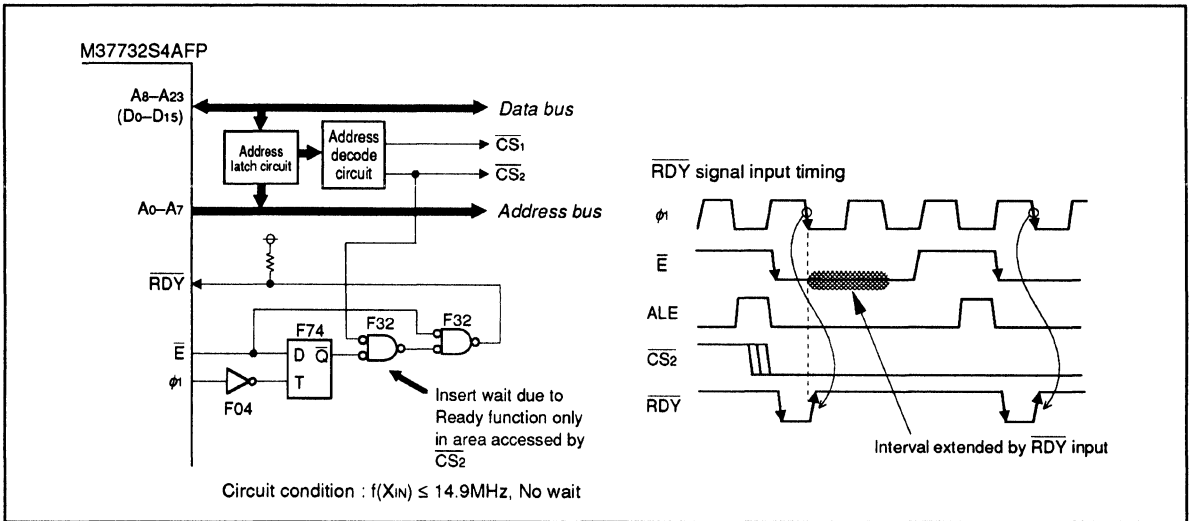


Fig. 9.1.7 Ready generating circuit example

9.1.5 Memory connection examples

Memory connection examples are described below.

(1) Minimum model memory connection example

Figure 9.1.8 shows a minimum model memory connection example with only ROM externally connected and Figure 9.1.9 shows a minimum model memory connection example with ROM and RAM externally connected.

With the minimum model, $\overline{R}/\overline{W}$ signal and \overline{E} signal are normally used to generate the read signal \overline{RD} and write signal \overline{WR} for memory access. The $\overline{BH\overline{E}}$ pin is left open because the $\overline{BH\overline{E}}$ signal is not used. In Figure 9.1.8, A_{15} and $\overline{R}/\overline{W}$ signals are used to generate the \overline{CE} signal in order to select the external ROM when addresses 8000_{16} to $FFFF_{16}$ are read because only ROM is externally connected. In this case, \overline{OE} control can be performed with the \overline{E} signal. This enables an EPROM with address access time of 100ns to be used at external clock input frequency 25MHz with software one-wait.

APPLICATION

9.1 Memory connection

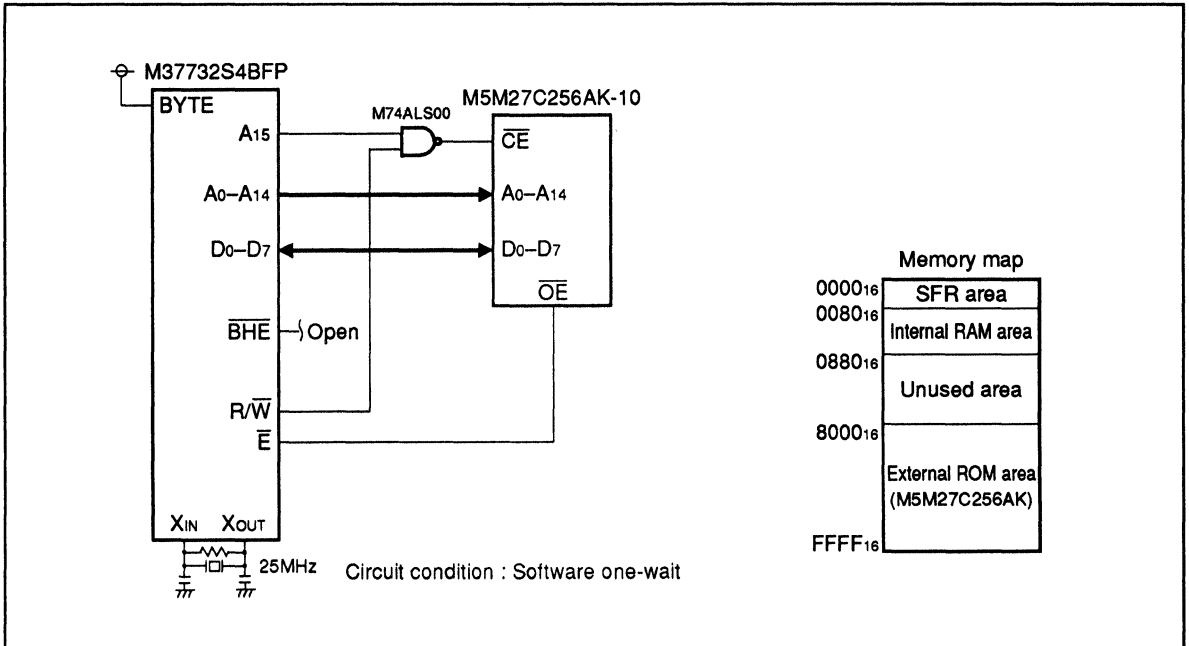


Fig. 9.1.8 Minimum model memory connection example (1)

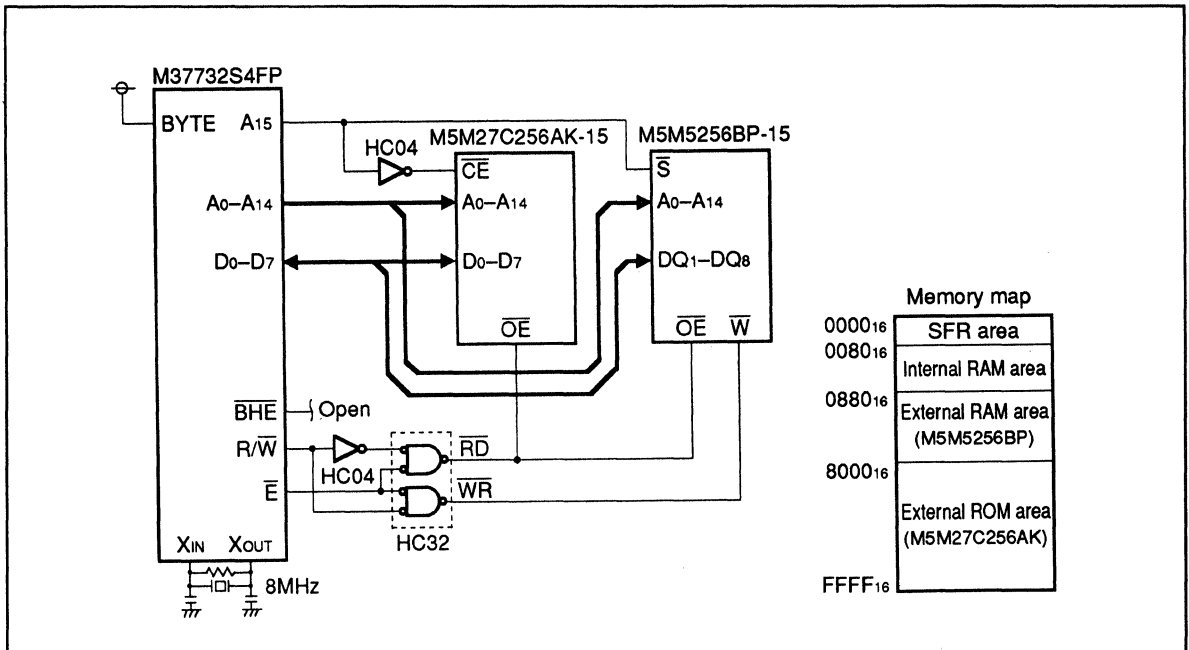


Fig. 9.1.9 Minimum model memory connection example (2)

(2) Maximum model memory connection example

Figure 9.1.10 shows a maximum model memory connection example. In maximum model, the memory write signal must be separated the write signal into \overline{WE} for even address and \overline{WO} for odd address because the external data bus width is used as 16 bits. The write signals for even address and odd address are separated by using A_0 of the address and the \overline{BHE} signal. The read signal need not be separated because the microcomputer selectively inputs the data during read.

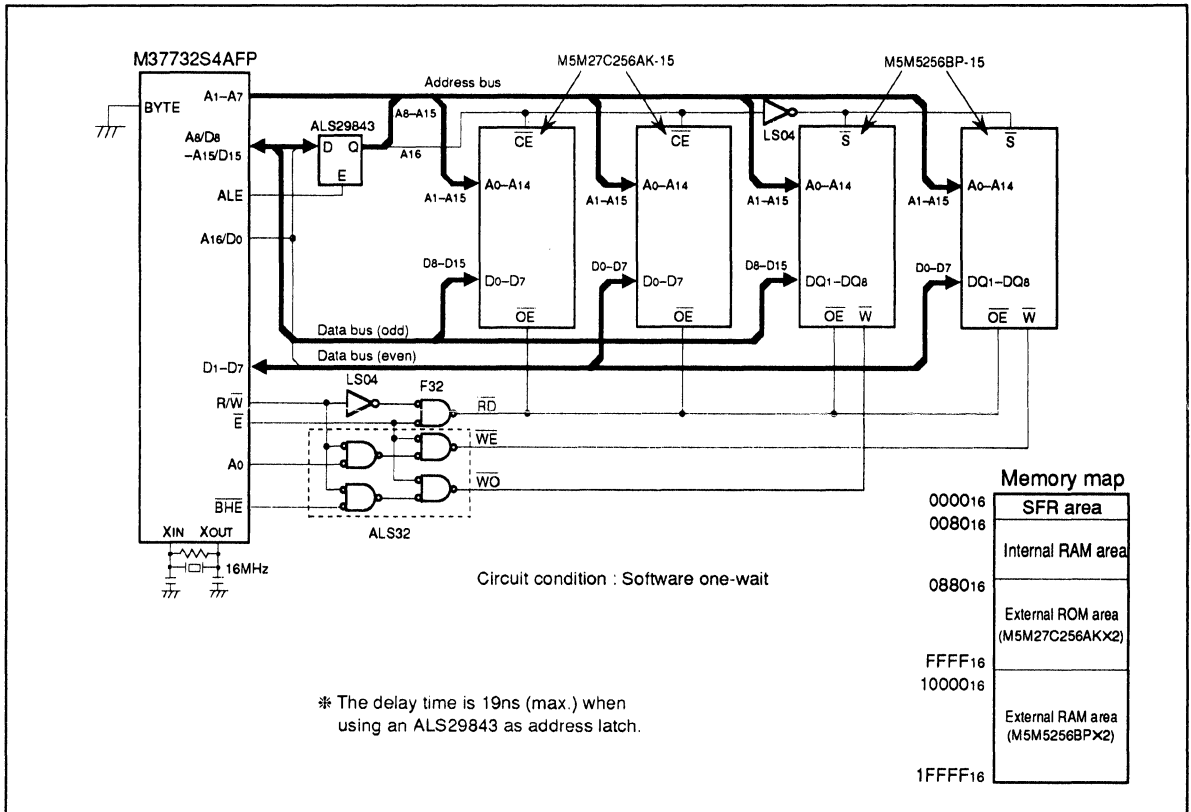


Fig. 9.1.10 Maximum model memory connection example

(3) Memory connection example at 20MHz using software one-wait

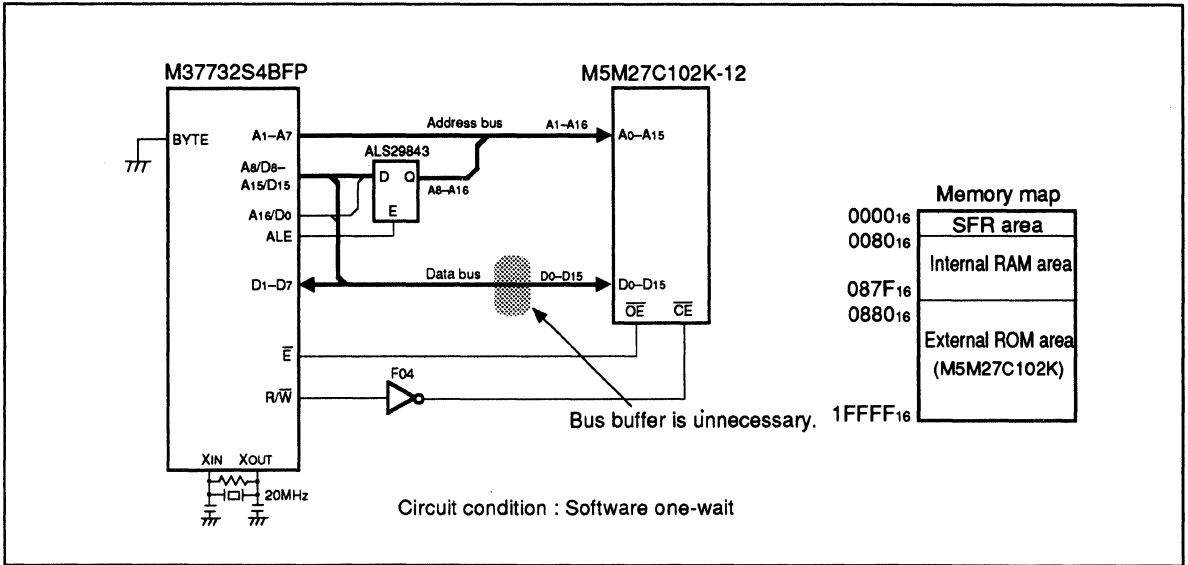
Figure 9.1.11 shows a memory connection example at 20MHz using software one-wait (maximum model). No bus buffer is required in this example because M5M27C102K-12 is used as an external ROM (refer to Table 9.1.4).

(4) Memory connection example at 25MHz using software one-wait

Figure 9.1.12 shows a memory connection example at 25MHz using software one-wait (medium model B). This example requires a bus buffer to prevent bus contention on the data bus.

APPLICATION

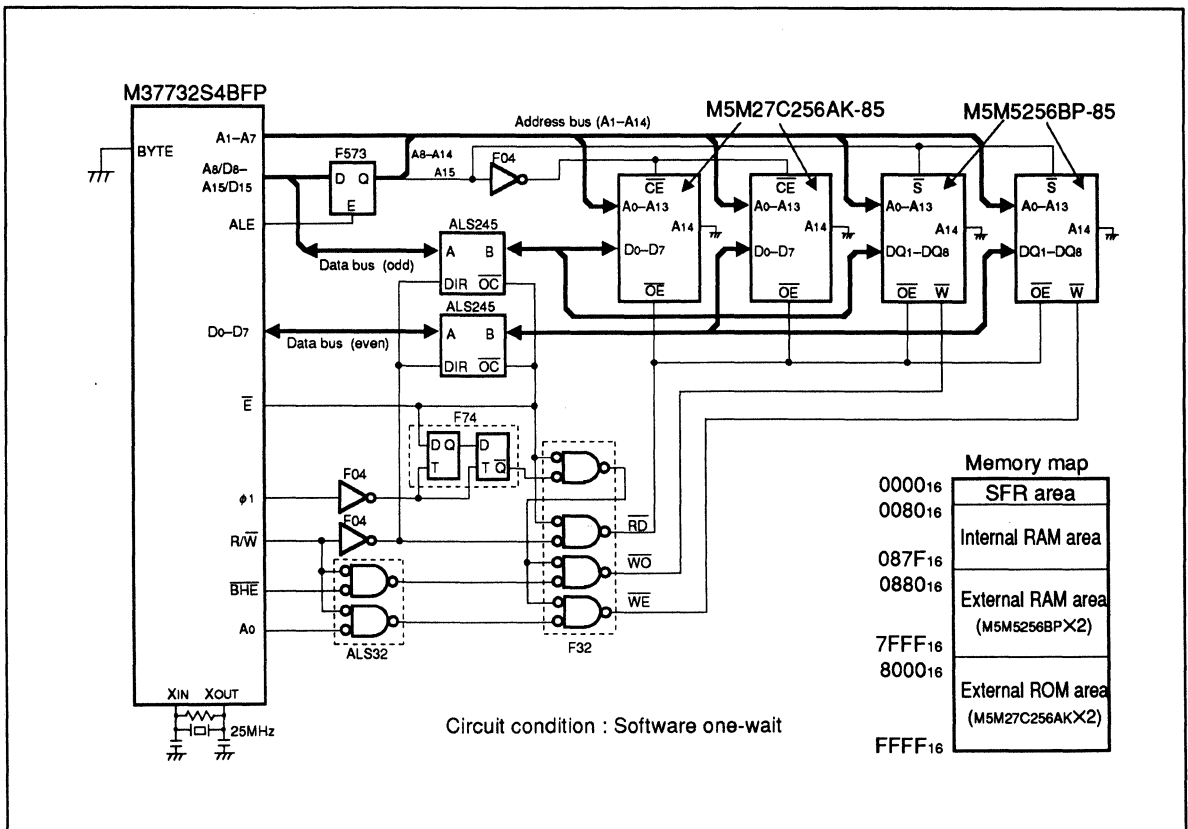
9.1 Memory connection



Memory map

0000 ₁₆	SFR area
0080 ₁₆	
087F ₁₆	Internal RAM area
0880 ₁₆	
0880 ₁₆	External ROM area (M5M27C102K)
1FFF ₁₆	

Fig. 9.1.11 Memory connection example at 20MHz using software one-wait



Memory map

0000 ₁₆	SFR area
0080 ₁₆	
087F ₁₆	Internal RAM area
0880 ₁₆	
0880 ₁₆	External RAM area (M5M5256BP×2)
7FFF ₁₆	
8000 ₁₆	External ROM area (M5M27C256AK×2)
FFFF ₁₆	

Fig. 9.1.12 Memory connection example at 25MHz using software one-wait

APPLICATION

9.1 Memory connection

(5) Memory connection example at 25MHz using software one-wait and Ready function

Figure 9.1.13 shows a memory connection example at 25MHz using software one-wait and Ready function (medium model A). In this example, two waits which consist of a software one-wait and a hardware one-wait by Ready function are inserted during ROM access.

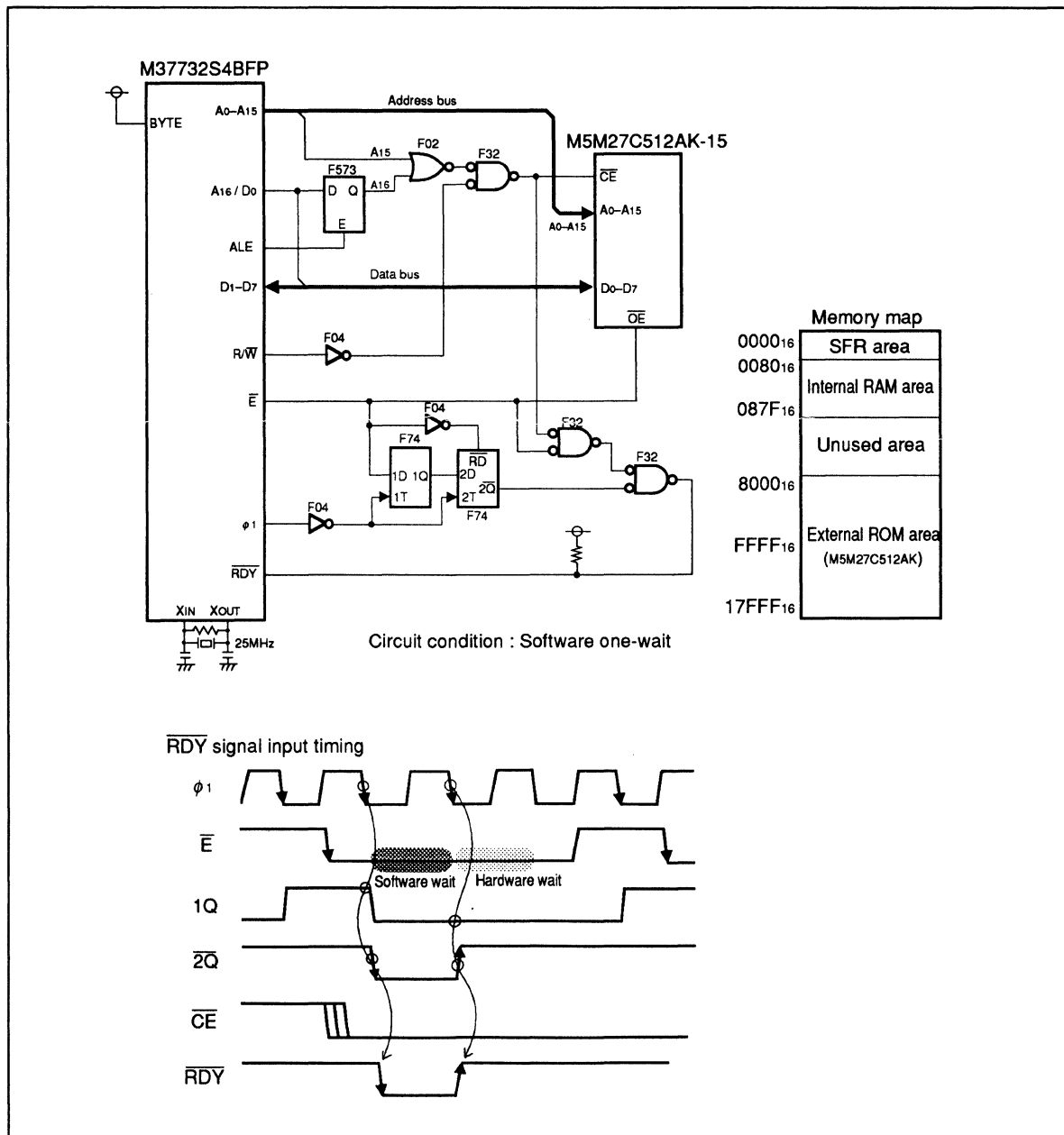


Fig. 9.1.13 Memory connection example at 25MHz using software one-wait and Ready function

(6) Memory connection example using M66800SP/FP

Figures 9.1.14 to 9.1.16 show memory connection examples using the M66800SP/FP (hereafter referred to as "M66800"). The "M66800" is a memory control IC that can be connected to either word bus or byte bus of the "M37732". It is equipped with an internal address decode circuit, read/write signal generating circuit, and an Ready generating circuit.

{Precautions when connecting memory using the "M66800"}

Reinforce the GND lines of the "M37732" and "M66800" in order to prevent errors due to noise. Also add an 80pF capacitor between the ALE signal line and GND line for safety.

Figure 9.1.14 shows an example of adding a 64KB EPROM (M5M27C512AK-10) and a 32KB SRAM (M5M5256BP-10) with no wait to the "M37732" that has an 8-bit external bus width and operates at 12.288MHz. The RDY output pin of the "M66800" is open because it operates at no wait.

Figure 9.1.15 shows an example of adding a 64KB EPROM (M5M27C512AK-15) and a 8KB high-speed SRAM (M5M5178P-45) to the "M37732" that has an 8-bit external bus width and operates at 15MHz. The 8KB high-speed SRAM is no wait and one-wait is inserted with Ready function for the EPROM. CS5 (EPROM chip select signal) is input as the wait request input (WRQ) to the "M66800" in order to limit the area in which the Ready function is valid.

Figure 9.1.16 shows an example of adding a 128KB EPROM (M5M27C102K-15) and two 32KB SRAMs (M5M5256BP-10) to the "M37732" that has a 16-bit external bus width and operates at 12.288MHz. The SRAM is no wait and one-wait is inserted with Ready function for the EPROM. Similar to Figure 9.1.15, an EPROM chip select signal is used to limit the area in which the Ready function is valid.

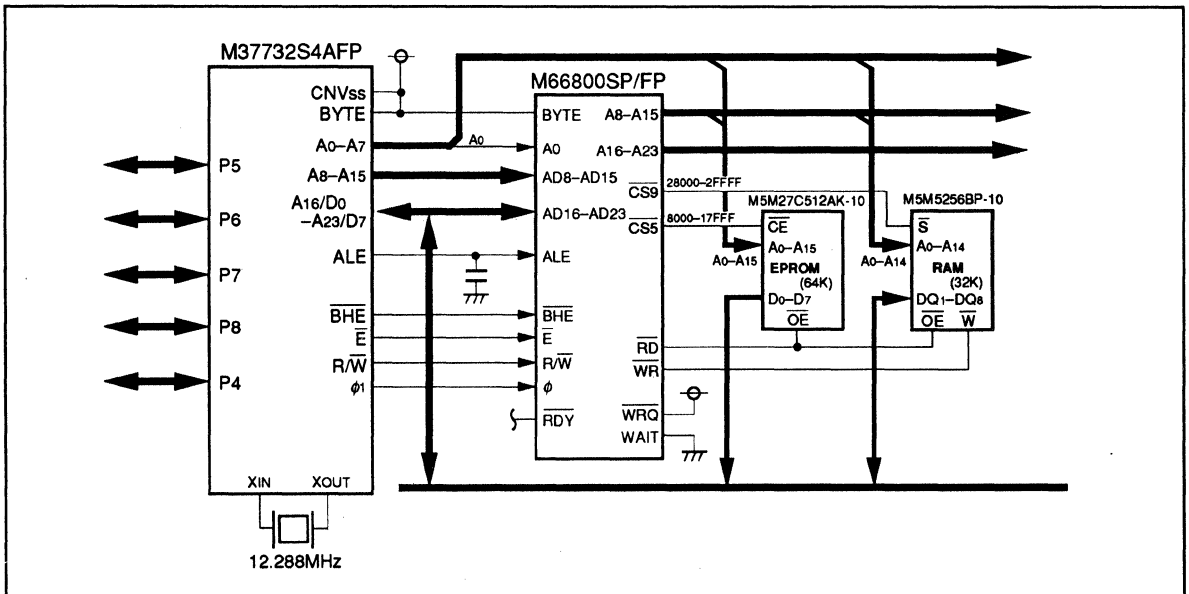


Fig. 9.1.14 Memory connection example using "M66800" (1)

APPLICATION

9.1 Memory connection

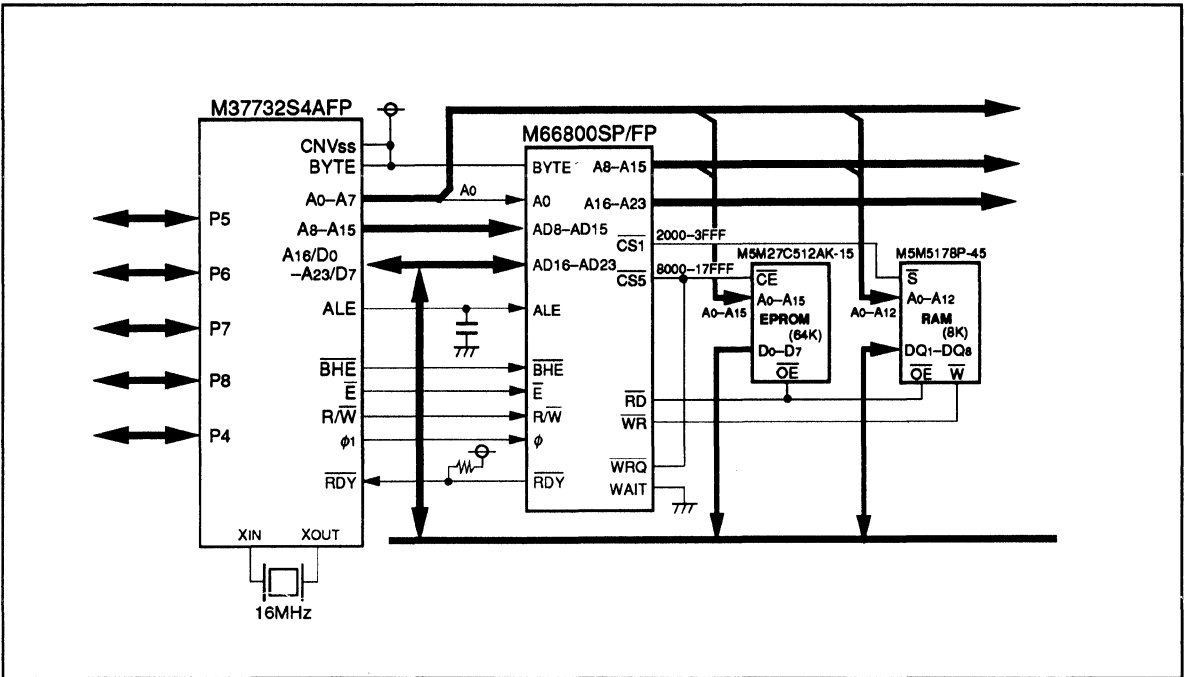


Fig. 9.1.15 Memory connection example using "M66800" (2)

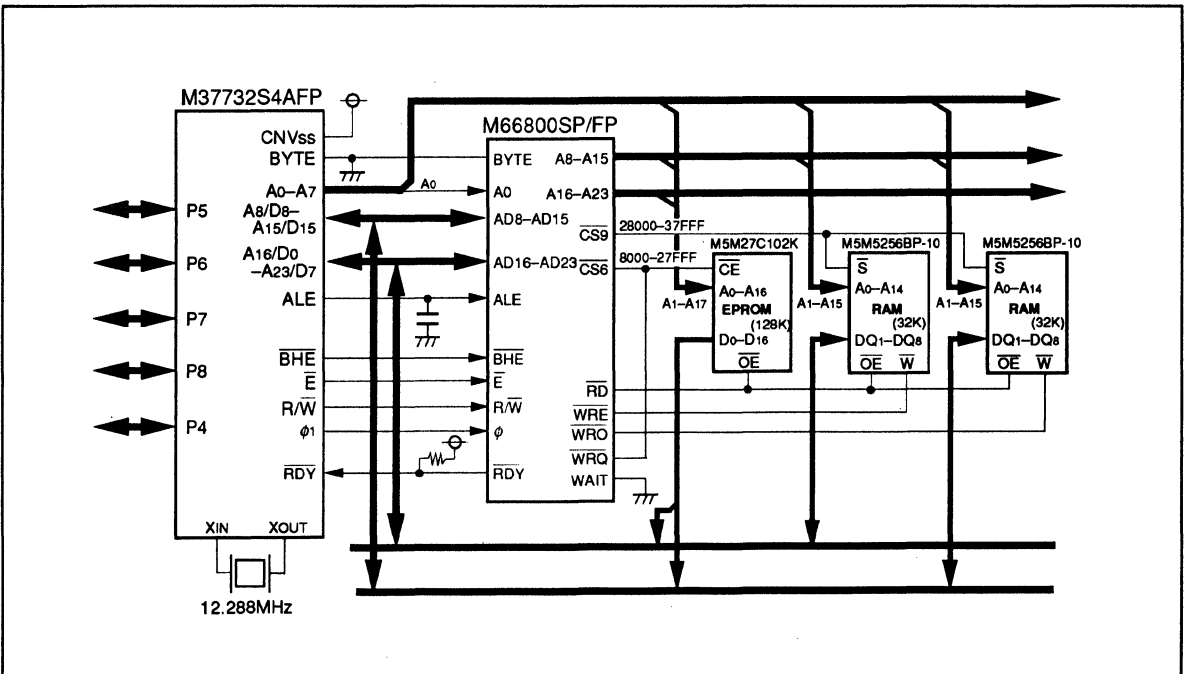


Fig. 9.1.16 Memory connection example using "M66800" (3)

9.2 I/O expansion

I/O expansion is described below.

9.2.1 I/O expansion model

Similar to memory connection, I/O expansion for the "M37732" can be performed for the four models shown in Table 9.1.1. Memory mapped I/O is used for I/O expansion. The expansion methods and precautions are the same as for memory connection.

9.2.2 I/O expansion examples

I/O expansion examples are described below.

(1) Port expansion example using M5M82C55AP-2/FP-2

Figure 9.2.1 shows a port expansion example using two M5M82C55AP-2/FP-2s. This is an expansion example for a medium model B, and one M5M82C55AP-2/FP-2 is connected to the even number port side and the other is connected to the odd number port side of the "M37732" data bus to expand the I/O port to 48. The device reset signal is supplied from port P4₃.

When using "M37732" at 8MHz or less, bus buffers such as ALS245A become unnecessary and direct bus connection is possible. In this case, F32 or TC74ACT32P can be used for RD separation gate, and the circuit condition is no wait.

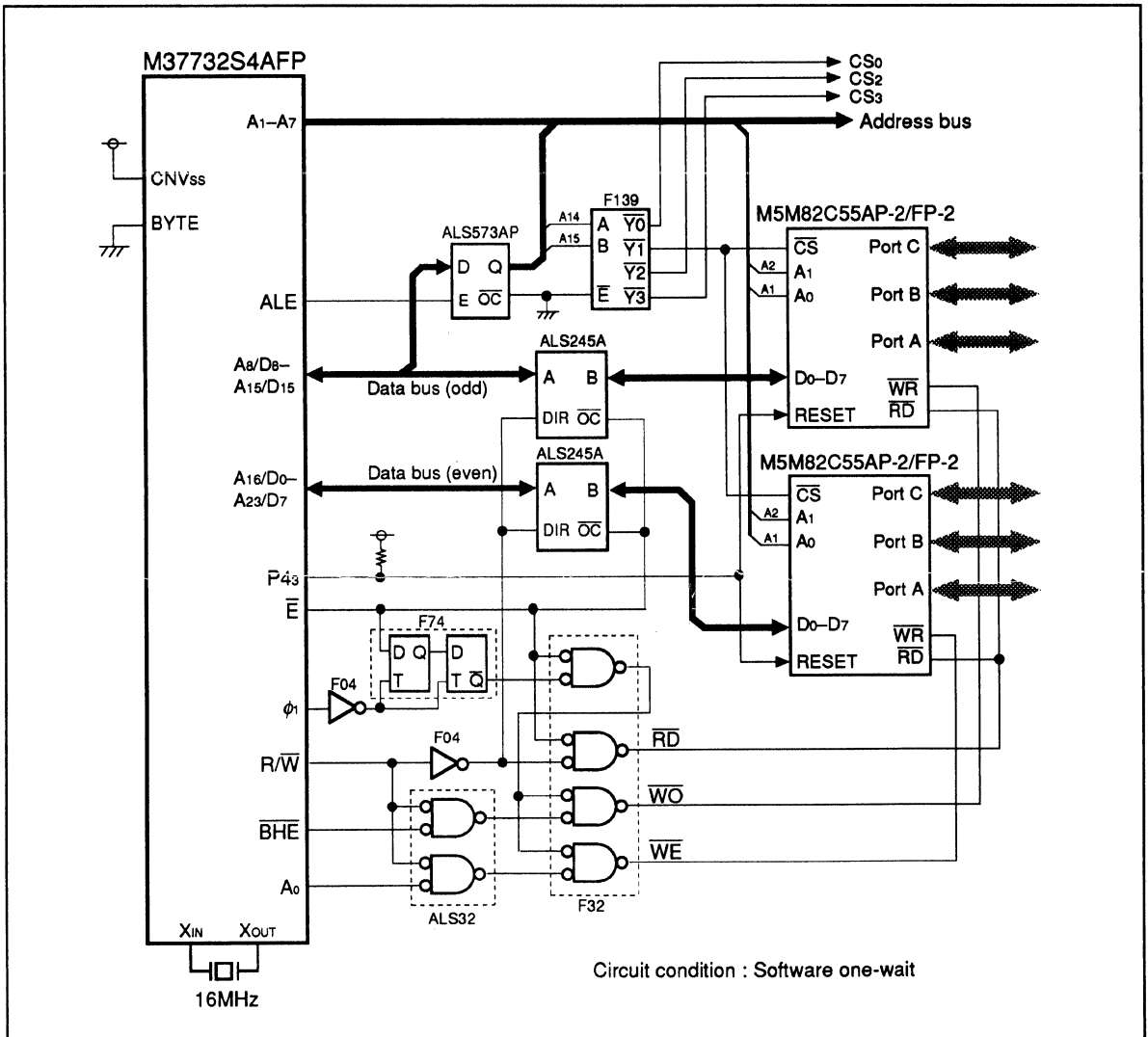


Fig. 9.2.1 Port expansion example using M5M82C55AP-2/FP-2

(2) Port expansion example using M66500SP/FP

Figure 9.2.2 shows a port expansion example using M66500SP/FP. This is an expansion example for a minimum model which adds 24 I/O ports, 16 high voltage output ports, and 4-bit input ports. When using "M37732" at 8MHz or less, bus buffers such as ALS245A become unnecessary and direct bus connection is possible. In this case, F32 or TC74ACT32P can be used for \overline{RD} separation gate, and the circuit condition is no wait.

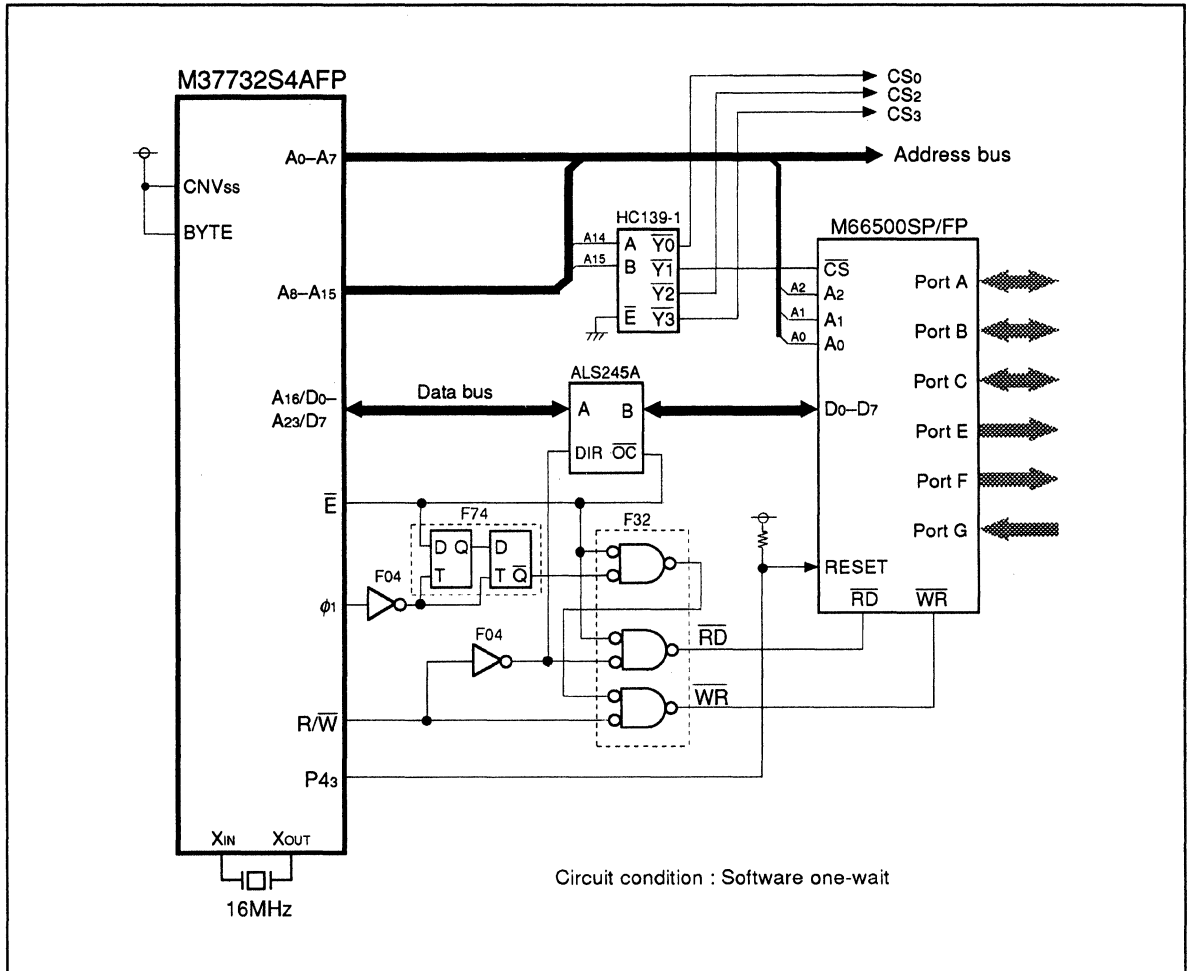


Fig. 9.2.2 Port expansion example using M66500SP/FP

(3) Port expansion example using M37451M4-XXXFP

Figure 9.2.3 shows an expansion example using an 8-bit single-chip microcomputer M37451M4-XXXFP. This is an expansion example for a minimum model and the host bus interface function of the M37451M4-XXXFP is used.

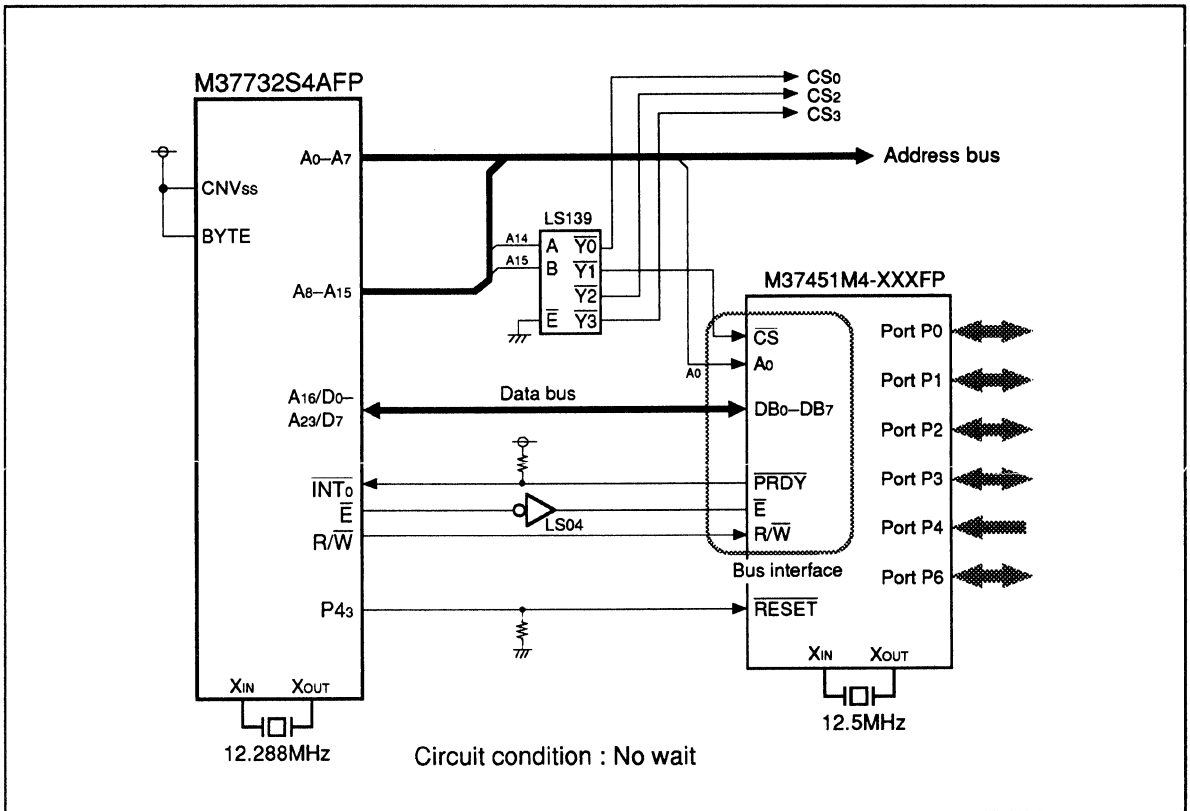


Fig. 9.2.3 Port expansion example using M37451M4-XXXFP

APPLICATION

9.3 Low supply voltage version applications

9.3 Low supply voltage version applications

The external memory connections of the M37732S4LGP/HP (hereafter referred to as "M37732L") are described below.

9.3.1 Memory connection

The memory connection model, memory access time calculation method, bus timing, and memory connection method of the "M37732L" are the same as for "M37732". However, some calculation equations and constants for parameters that depend on the external clock input frequency are different. Equation ① shows how to calculate the memory access time $t_{a(AD)}$ and equation ② shows how to calculate the data setup time $t_{su(D)}$.

$$t_{a(AD)} = t_{d(AL/AM/AH-E)} + t_{w(EL)} - t_{su(DL/DH-E)} - \{\text{address decode time} + \text{address latch delay time}^*\} \dots \textcircled{1}$$

* Address latch delay time is not necessary for minimum model.

$$t_{su(D)} = t_{w(EL)} - t_{d(E-DLQ/DHQ)} \dots \textcircled{2}$$

In equation ①, $t_{d(AL/AM/AH-E)}$ represents either $t_{d(AL-E)}$, $t_{d(AM-E)}$, or $t_{d(AH-E)}$ and $t_{su(DL/DH-E)}$ represents either $t_{su(DL-E)}$ or $t_{su(DH-E)}$. In equation ②, $t_{d(E-DLQ/DHQ)}$ represents either $t_{d(E-DLQ)}$ or $t_{d(E-DHQ)}$.

Table 9.3.1 shows the calculation equations and constants for parameters of "M37732L" that depend on the external clock input frequency.

Figure 9.3.1 shows the relationship between memory access time $t_{a(AD)}$ and external clock input frequency $f(X_{IN})$ and Figure 9.3.2 shows the relationship between data setup time $t_{su(D)}$ for write and external clock input frequency. Similar to the graph for "M37732", the graph in Figure 9.3.1 shows the memory access time when address decode time and address latch delay time are not taken into consideration. Therefore, these values must be subtracted from the data in the graph to obtain the actual memory access time.

APPLICATION

9.3 Low supply voltage version applications

Table 9.3.1 Parameter equation and constants depending on the external clock input frequency

Parameter	V _{CC}	2.7 to 5.5V
f(X _{IN})		f(X _{IN}) ≤ 8MHz
t _d (AL-E) t _d (AM-E) t _d (AH-E)		$50 + \frac{1 \times 10^9}{f(X_{IN})} - 125$
t _w (ALE)		$\frac{1 \times 10^9}{f(X_{IN})} - 65$
t _w (EL)		$\frac{2 \times 10^9}{f(X_{IN})} - 40$
Under the one-wait mode (the wait bit = "0")		$\frac{4 \times 10^9}{f(X_{IN})} - 40$
t _{su} (DL-E) t _{su} (DH-E)		80
t _d (E-DLQ) t _d (E-DHQ)		130
t _d (ALE-E)		4
t _{pxz} (E-DLZ) t _{pxz} (E-DHZ)		10
t _{pzx} (E-DLZ) t _{pzx} (E-DHZ)		$\frac{1 \times 10^9}{f(X_{IN})} - 30$

(Unit : ns)

APPLICATION

9.3 Low supply voltage version applications

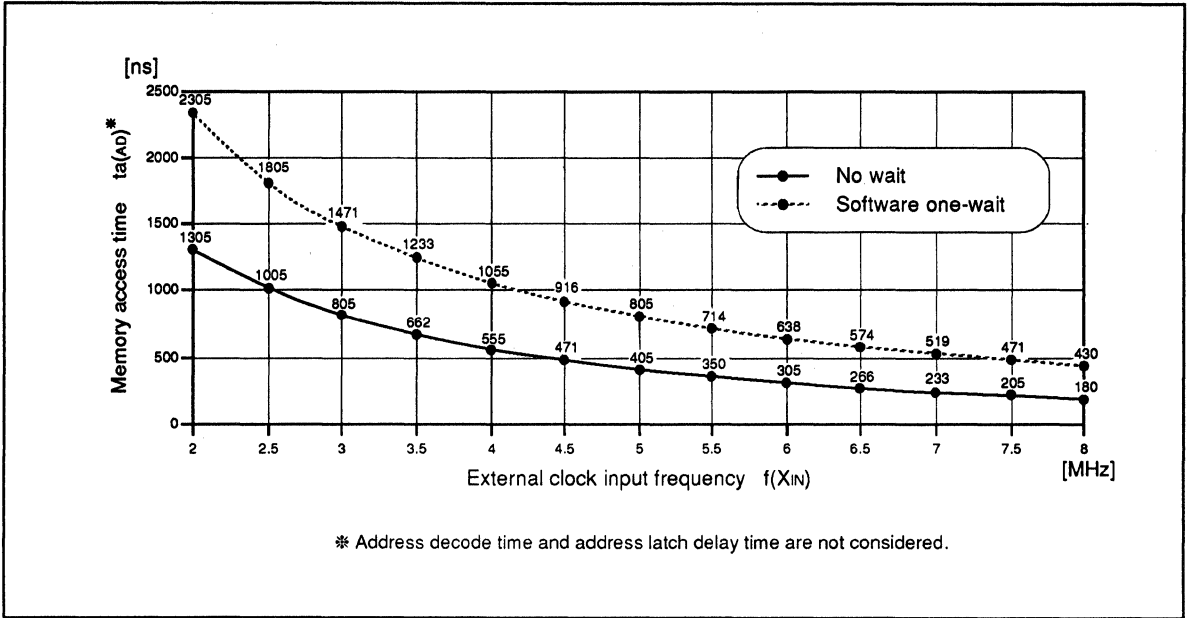


Fig. 9.3.1 Relationship between memory access time and external clock input frequency

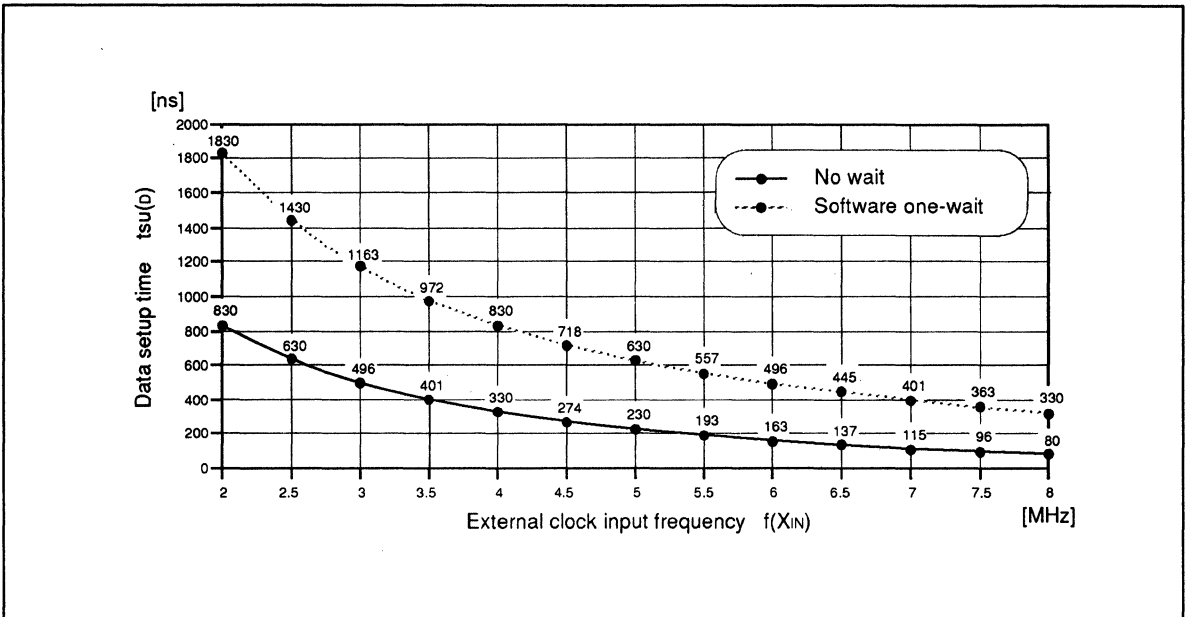


Fig. 9.3.2 Relationship between write data setup time and external clock input frequency

APPLICATION

9.3 Low supply voltage version applications

(1) Minimum model memory connection example

Figure 9.3.3 shows the memory connection example for a minimum model without external RAM and Figure 9.3.4 shows the memory connection example for a minimum model with external RAM. These examples use Atmel Corporation EPROM (AT27LV256R) as external ROM.

The examples in Figure 9.3.3 and 9.3.4 access the external memory with software one-wait. To use these circuits with no wait, lower the external clock input frequency to 5.8MHz or less.

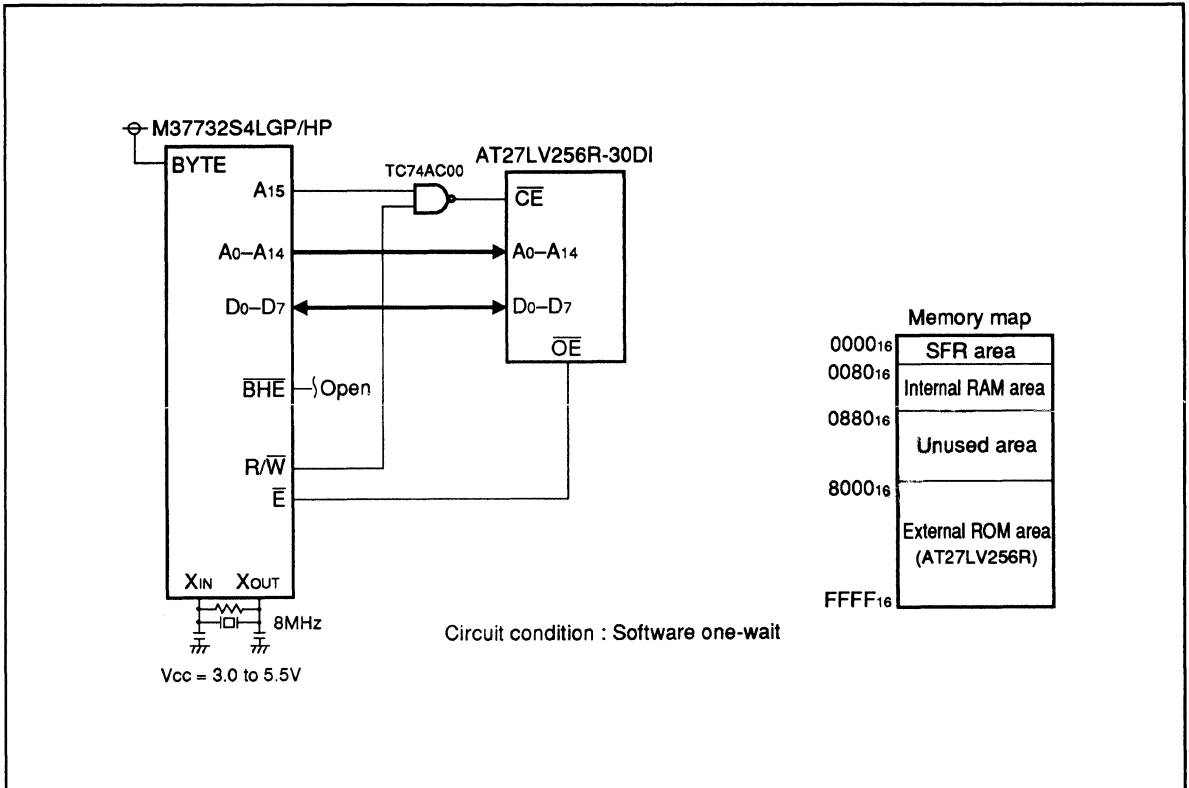


Fig. 9.3.3 Minimum model memory connection example (without external RAM)

APPLICATION

9.3 Low supply voltage version applications

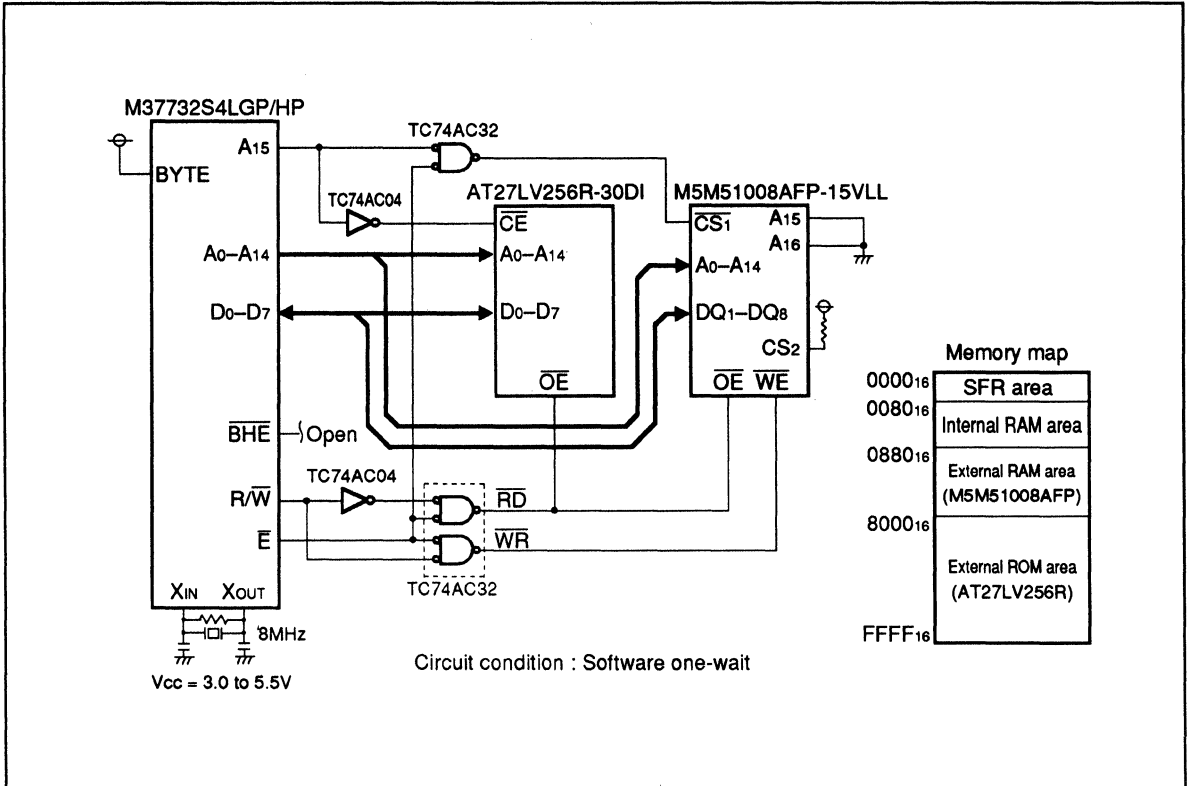


Fig. 9.3.4 Minimum model memory connection example (with external RAM)

APPLICATION

9.3 Low supply voltage version applications

(2) Maximum model memory connection example

Figure 9.3.5 shows the memory connection example for a maximum model. This example uses Atmel Corporation EPROM (AT27LV256R) as external ROM.

This example accesses the external memory with software one-wait. To use this circuit with no wait, lower the external clock input frequency to 5.8MHz or less.

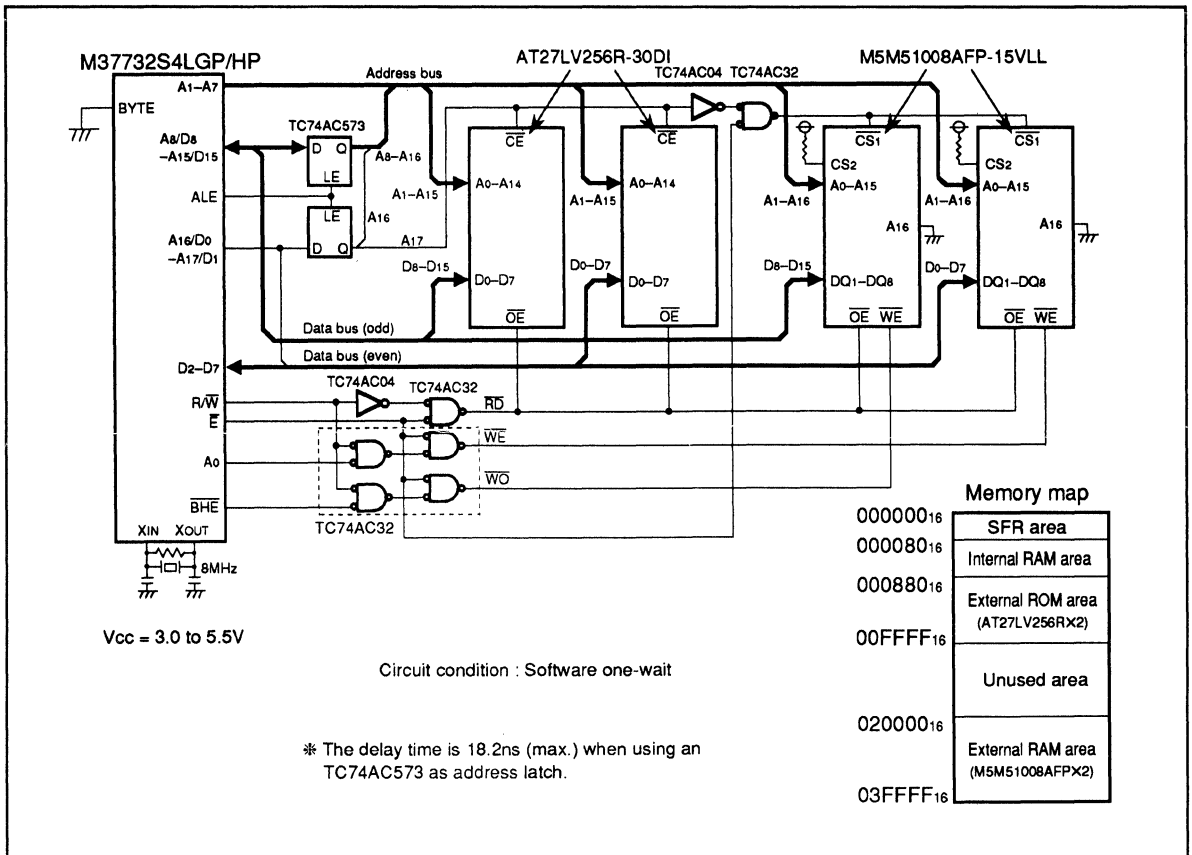


Fig. 9.3.5 Maximum model memory connection example

APPLICATION

9.3 Low supply voltage version applications

(3) Ready generating circuit example

In connection examples shown in Figures 9.3.3 to 9.3.5, use Ready function if wait is to be inserted only for a certain area (ROM area for example). Figure 9.3.6 shows an example of a Ready generating circuit.

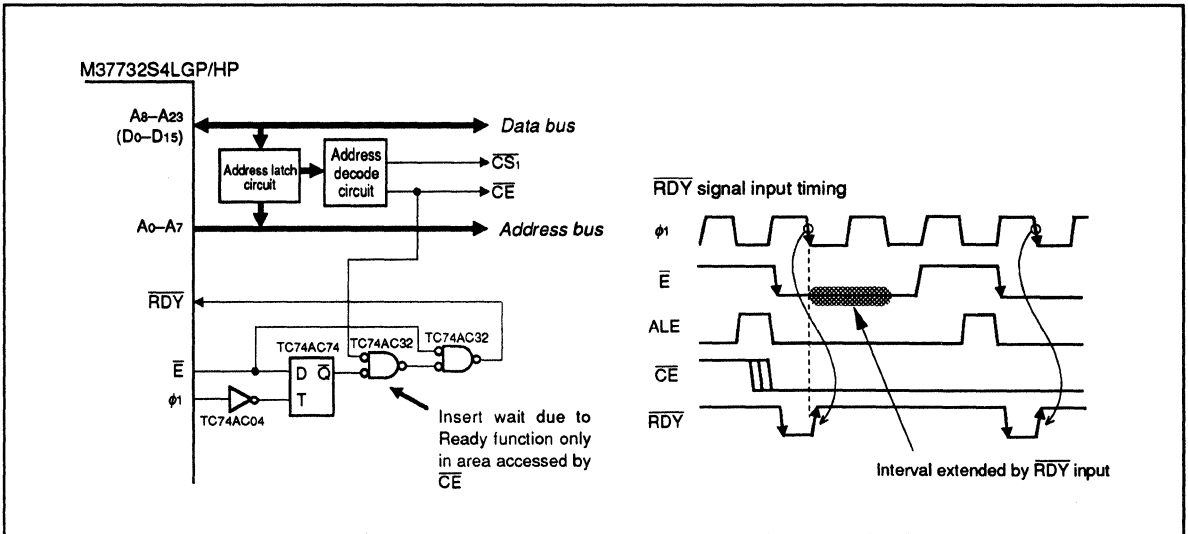


Fig. 9.3.6 Ready generating circuit example

9.4 Power saving

Described in this section are power dissipation conserving (power saving) of system (hereafter refers to the entire system including the microcomputer as system unless specified otherwise) and internal RAM back-up application examples using wait mode (entered by executing a **WIT** instruction) and stop mode (entered by executing an **STP** instruction).

When a **WIT** instruction is executed, the wait mode is entered and the internal clock ϕ stops at "L" level state. The oscillator does not stop however. When an **STP** instruction is executed, the stop mode is entered and oscillation stops with the internal clock ϕ at "L" level state.

In wait mode, the power dissipation is approximately 20% compared to normal operation. The power saving for wait mode is slightly less than that for stop mode. However, wait mode has the advantage in that an instruction can be executed immediately when the mode is removed.

In stop mode, normal operation cannot restore immediately at removing stop mode because some time is necessary for the oscillation to stabilize. However, power is conserved considerably because oscillation is stopped and power dissipation is a minimal $1\mu\text{A}$ ($T_a=25^\circ\text{C}$). Therefore, this mode is suitable for battery driven portable products.

Refer to section "Appendix 3. Stop, wait, one-wait, Ready, and Hold state" for information on wait mode and stop mode state.

9.4.1 Wait mode

In wait mode, the system power is conserved while supplying power to the entire system.

If wait mode is entered with an external memory selected, the entire system power cannot be conserved because the power dissipation of the selected memory does not decrease. Therefore, be sure to deactivate the chip select signal before entering the wait mode (the method differs for each system).

One method to deactivate the chip select signal is to execute a **WIT** instruction in the internal RAM area. Figure 9.4.1 shows an example of a **WIT** instruction execution circuit and Figure 9.4.2 shows an example of a **WIT** instruction execution program. In this program example, a **WIT** instruction is transferred to internal RAM from external ROM with an **MVN** instruction and executed in internal RAM. When a **WIT** instruction is executed in internal area, the chip select signals ($\overline{\text{CE}}$ and $\overline{\text{CS}}_1$ in Figure 9.4.1) are deactivated and no external memory is selected (do not allocate external memory at the same address as the internal RAM area in which the **WIT** instruction is executed).

In addition to the chip select signal, ports must also be externally deactivated.

In wait mode, peripheral devices such as watchdog timer are still operating. When using the watchdog timer, the chip select signal must be deactivated by the above procedure when terminating the watchdog timer interrupt and returning to wait mode with a **WIT** instruction.

In the program example shown in Figure 9.4.2, if a watchdog timer interrupt occurs while in wait mode, writing to watchdog timer is performed in the interrupt routine and a **WIT** instruction is re-executed in the internal RAM area. If a watchdog timer interrupt occurs outside the wait mode, program runaway is assumed and a software reset occurs. The wait mode is removed with timer A0 interrupt.

In addition to the above, measures such as stopping the operation of timers, serial I/O, and A-D converter, or externally separating V_{REF} with cut-off transistor or analog switch can be taken to further conserve power (the method differs from system to system).

[Precaution during wait mode]

When executing a write instruction just before a **WIT** instruction, insert some **NOP** instructions after the write instruction to make sure the write instruction completes before the **WIT** instruction is executed (refer to section "Appendix 3. Stop, wait, one-wait, Ready, and Hold state" for the number of instructions to be inserted).

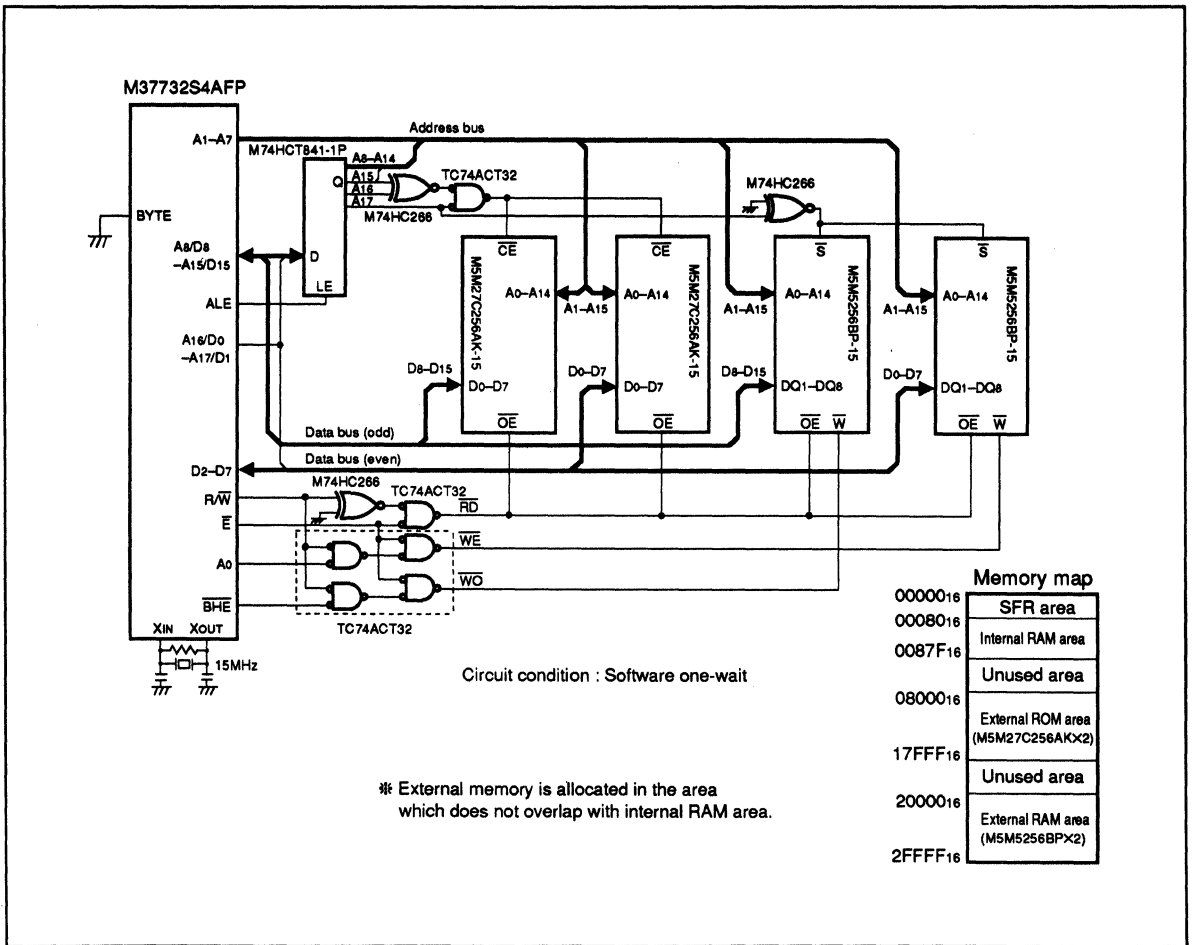


Fig. 9.4.1 Example of WIT instruction execution circuit

```

282 ;-----
283 ; Example of WIT instruction execution program =
284 ;-----
285 ;
286 008021 E224 SEP m,1 ;
287 008023 *848000 L LDM.B #0000000B,FLG ; Bit 0:Wait mode restore flag
288 ; ; Bit 1:WIT instruction execution flag
289 008028 *8458C4 L LDM.B #11000100B,TAOMR ; Set timer A0
290 008029 D8 CLM ;
291 00802A *8448FFFF L LDM.W #OFFFHH,TAO ;
292 00802E F8 SEM ;
293 00802F *14750F L CLB.B #00001111B,TA0IC ; Clear timer A0 interrupt request bit
294 008032 *047501 L SEB.B #00000001B,TA0IC ; Enable timer A0 interrupt (level 1)
295 008035 58 CLI ; Enable interrupt
296 008038 *844001 L LDM.B #00000001B,TABSR ; Start count
297
298 008039 ROM_1:
299 008039 C210 CLP X ;
300 00803B A24B80 L LDX.W #ROM_3 ; Transfer source address (external ROM)
301 00803E A00001 L LDY.W #RAM ; Transfer destination address (internal RAM)
302 008041 A912 LDA.B A,#12H ; Transfer byte count
303 008043 540000 MVN 0,0 ; Transfer data
304 008046 4C0001 L JMP RAM ; Jump to RAM area
305
306 008049 ROM_2:
307 008049 80FE L BRA ROM_2
308
309 00804B ROM_3:
310 00804B F8 SEM ;
311 00804C C210 CLP X ;
312 00804E *048002 L SEB.B #00000010B,FLG ; Set WIT instruction execution flag
313 008051 EA NOP ;
314 008052 ROM_4:
315 008052 CB WIT ; Execute WIT instruction
316 008053 *348001FB L BBC.B #00000001B,FLG,ROM_4 ; Restore from wait mode ?
317 ; ; Not restore --> Execute WIT instruction again
318 008057 *148003 L CLB.B #00000011B,FLG ; Restore --> Clear WIT instruction execution flag
319 ; ; Clear wait mode restore flag
320 00805A 4C4980 L JMP ROM_2 ; Return to ROM area
321 ;
322 ;
323 .ORG 9000H
324 ;-----
325 ; Example of interrupt processing program =
326 ;-----
327 ;
328 009000 INT_TAO: ; *** Interrupt routine for restoring from wait mode
329 009000 F8 SEM ;
330 009001 *048001 L SEB.B #00000001B,FLG ; Restore from wait mode
331 009004 40 RTI
332
333 009005 INT_WDT: ; *** Watchdog timer interrupt routine
334 009005 F8 SEM ;
335 009008 *24800203 L BBS.B #00000010B,FLG,WDT_1 ;
336 00900A *045E08 L SEB.B #00001000B,PMR ; Program runaway
337 ; ; --> Software reset
338
339 00900D WDT_1:
340 00900D *8580 L STA A,WDT ; WDT interrupt
341 ; ; --> Writing to watchdog timer
342 00900F 40 RTI

```

Fig. 9.4.2 Example of WIT instruction execution program

9.4.2 Stop mode

When oscillation is stopped by executing an **STP** instruction, the current dissipation of the "M37732" itself can be conserved considerably to $1\mu\text{A}$ ($T_a=25^\circ\text{C}$). Furthermore, when the external bus width is 8 bits ($\overline{\text{BHE}}$ open) and no wait (or if software one-wait is available for external memory, by transferring the **STP** instruction to internal RAM and executing it in internal RAM just as in a **WIT** instruction), power to the entire system except the "M37732" can be stopped by supplying power to only the "M37732" from a secondary battery such as a Lithium battery. In this case, the internal RAM is preserved (internal RAM back-up).

Figure 9.4.3 shows an example of an **STP** instruction execution program.

When an **STP** instruction is executed, "M37732" stops with "H" level output from some address pins because the oscillation stops after accessing the addresses in ROM near the **STP** instruction is allocated to. When power supply to external CMOS memory and CMOS gate is stopped in this state, the system power dissipation cannot be conserved because current flows to these devices from the "H" level ports of the "M37732" due to the diode effect. Therefore, this program example makes sure the address/data bus, $\overline{\text{ALE}}$, $\overline{\text{R/W}}$, and $\overline{\text{E}}$ output stop with "L" level state when executing the **STP** instruction (**NOP** instructions are not inserted between the **STP** instruction and write instruction because the oscillation is stopped in the middle of a write operation).

Stop mode is removed with a hardware reset or an accepting external interrupt. If a ceramic resonator is used, return the $\overline{\text{RESET}}$ pin to "H" level after applying "L" level to the $\overline{\text{RESET}}$ pin for an interval longer than the time required for the oscillation to stabilize when returning by hardware reset.

Figure 9.4.4 shows an example of power-on reset circuit for internal RAM back-up using a system reset IC M62003.

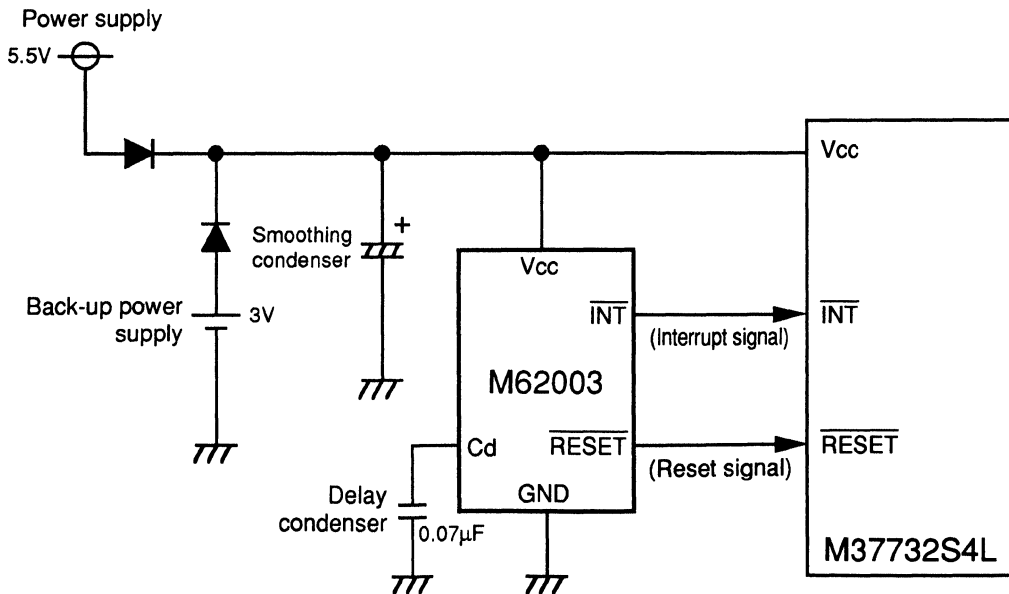
Note that in the back-up circuit example, circuit and circuit constants must be adjusted according to the actual usage condition of the power supply system.

```

295 ; ;
296 ; ;
297 ; ;
298 ; ;
299 00C022 DA PHX ; Store the contents of X register to stack
300 00C023 89C201 LDT #01H ; Set bank other than 0
301 ; ;
302 00C028 C280 CLP X,M ; Data length, register length 18 bits
303 00C028 A20000 LDX.W #0000H ;
304 00C02B 8001 BRA lab ;
305 [ROM+1][RAM+0] L .EVEN ; Adjust to even address
306 00C02E lab: ;
307 00C02E 9EFFFF0000 LDM.W #0000H,DT:OFFFH,X ;
308 00C033 DB STP ; Execute STP instruction
309 ; ;
310 ; ;
311 ; Return data bank register, data length and so on to the former state.
312 ; And then restore the contents which stored to stack.
313 ; ;
314 ; ;
315 ; ;

```

Fig. 9.4.3 Example of STP instruction execution program for memory back-up



* In this circuit example, a 0.07µF delay condenser is used because the rise of the reset signal is delayed approximately 10ms.

Fig. 9.4.4 Example of power-on reset circuit for internal RAM back-up

9.5 Pulse motor control application

This section describes the case of stepping motor 1-2 excitation (without pulse width modulation) control as an application example using the basic functions of the pulse output port mode of the M37732 group.

Figure 9.5.1 shows the procedure for setting the pulse output port mode and Figure 9.5.2 shows an example of initialization for stepping motor control. Figure 9.5.3 shows an example of updating the pulse output data and output timing, Figure 9.5.4 shows the output waveform example when using these programs, and Figure 9.5.5 shows the RTP0₀–RTP1₃ output control block diagram.

In this example, RTP1 is used to update pulse output data registers 1 and timer A2 in the timer A2 interrupt service routine during motor control.

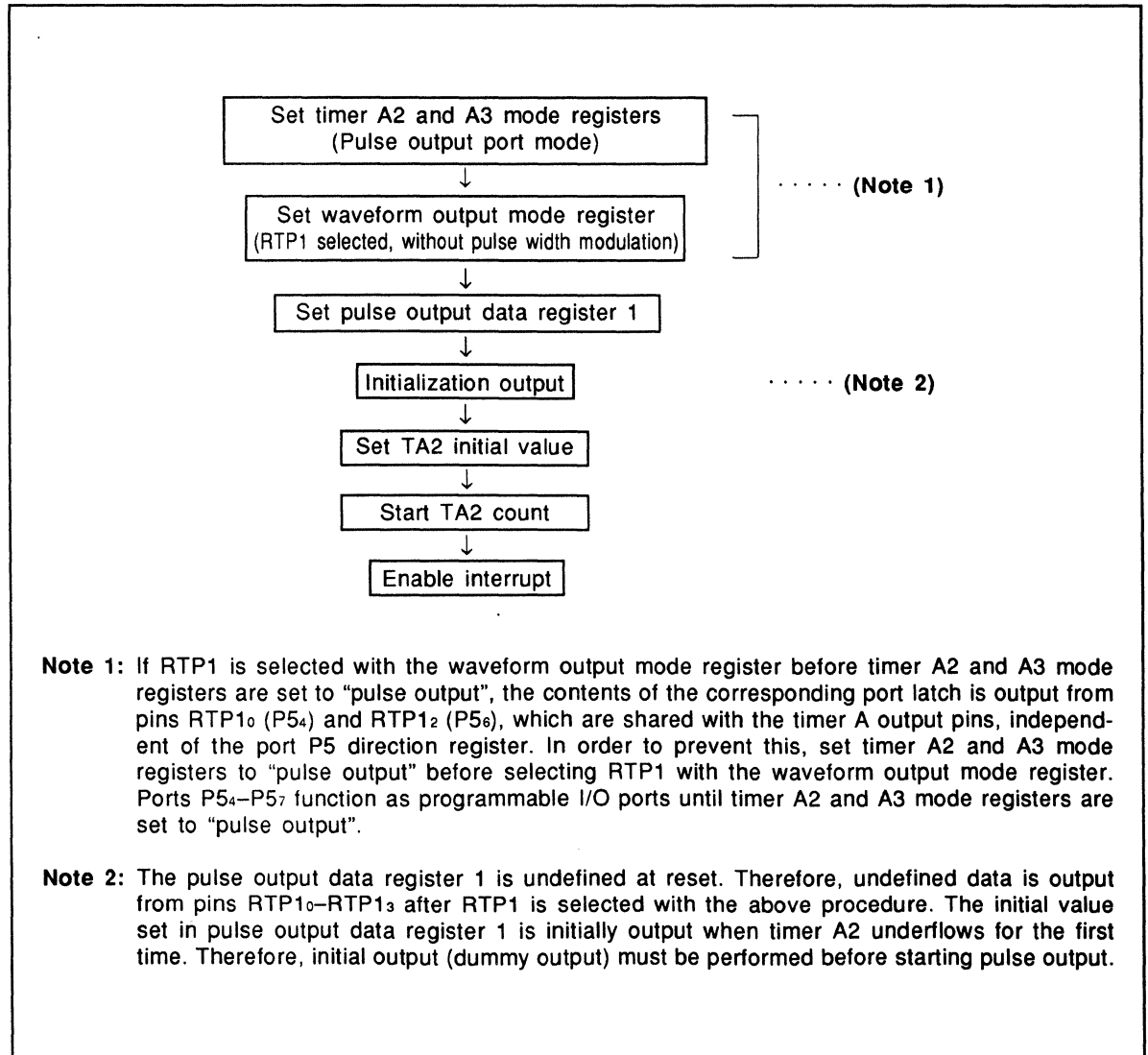


Fig. 9.5.1 Pulse output port mode setting procedure

```

292                                     ;=====
293                                     ;   Initialization for stepping motor control   =
294                                     ;=====
295                                     ;
296 00C027          SUB_PL1:
297 00C027 78          SEI                      ; Disable interrupt
298
299                      .DATA 8
300 00C028 F8          SEM
301 00C029 *144004    L          CLB          #00000100B,TABSR      ; Stop timer A2 count
302 00C02C *845844    L          LDM          #01000100B,TA2MR      ; Timer mode, enable pulse output, disable gate function
303 00C02F *845804    L          LDM          #00000100B,TA3MR      ; Enable pulse output (set P58 to output)
304 00C032 *846201    L          LDM          #00000001B,WOUTMR      ; Disable pulse width modulation, select RTP1
305 00C035 *846404    L          LDM          #00000100B,POUTDR1     ; Set initial value of pulse output data register 1
306 00C038 *14770F    L          CLB          #00001111B,TA2IC      ; Clear timer A2 interrupt request bit
307 00C03B *047701    L          SEB          #00000001B,TA2IC      ; Enable timer A2 interrupt (level 1)
308
309                      .DATA 16
310 00C03E D8          CLM
311 00C03F *844A0200  L          LDM          #0002H,TA2          ; Initial output of timer A2
312
313                      .DATA 8
314 00C043 F8          SEM
315 00C044 *044004    L          SEB          #00000100B,TABSR      ; Start timer A2 count
316 00C047          LOOP_1:
317 00C047 *347708FC  L          BBC          #00001000B,TA2IC,LOOP_1 ; Initial output complete?
318 00C04B *144004    L          CLB          #00000100B,TABSR      ; Stop count
319
320                      .DATA 16
321 00C04E D8          CLM
322 00C04F *844A204E  L          LDM          #4E20H,TA2          ; Set timer A2 initial value (period:20000 µs at 16MHz)
323
324                      .DATA 8
325 00C053 F8          SEM
326 00C054 *147708    L          CLB          #00001000B,TA2IC      ; Clear timer A2 interrupt request bit
327 00C057 *044004    L          SEB          #00000100B,TABSR      ; Start timer A2 count
328 00C05A 58          CLI                      ; Enable interrupt
329
330 00C05B 80          RTS

```

Fig. 9.5.2 Example of initialization for stepping motor control

```

333                                     ;=====
334                                     ;   Timer A2 interrupt           =
335                                     ;   Updating of pulse output data and =
336                                     ;   output timing (timer A2 data)  =
337                                     ;=====
338                                     ;
339 00C05C                               INT_TA2:
340                                     .DATA 16
341 00C05C D8                             CLM
342                                     ;
343 00C05D 48                             PHA                               ; Store registers
344                                     ;                               ;
345                                     ;                               ;
346                                     .DATA 8
347 00C05E F8                             SEM
348                                     ;
349 00C05F *AD00C0                         L   LDA   A,PLS_DATA           ; Change pulse output data register 1
350 00C082 *8584                           L   STA   A,POUTDR1          ;
351                                     ;
352                                     .DATA 16
353 00C084 D8                             CLM
354                                     ;
355 00C085 *AD00C0                         L   LDA   A,TA2_DATA        ; Change timer A2 data
356 00C088 *854A                           L   STA   A,TA2            ;
357                                     ;
358                                     ;                               ; Next PLS_DATA,TA2_DATA calculation
359                                     ;                               ; and so on
360                                     ;                               ;
361                                     ;                               ;
362 00C08A 68                             PLA                               ; Restore registers
363 00C08B 40                             RTI

```

Fig. 9.5.3 Example of updating pulse output data and output timing for stepping motor control

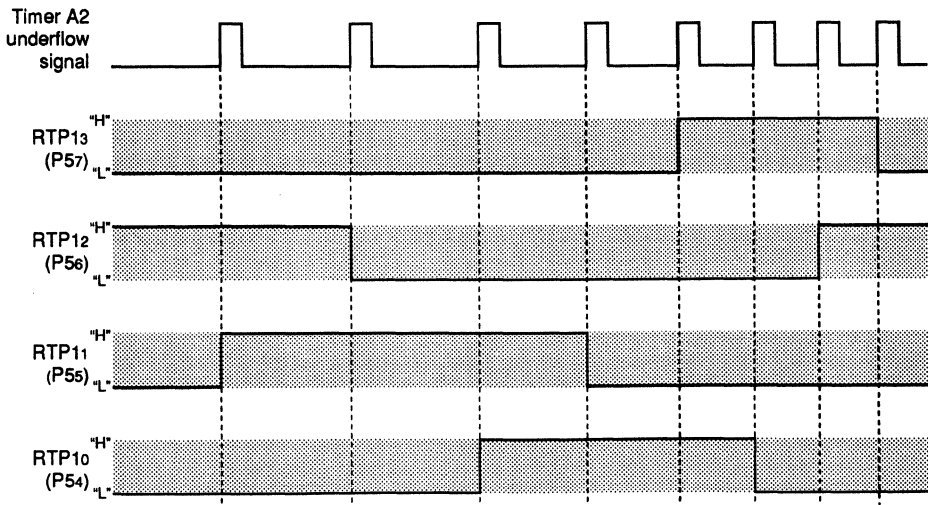


Fig. 9.5.4 Example of output waveform for stepping motor 1-2 excitation (without pulse width modulation)

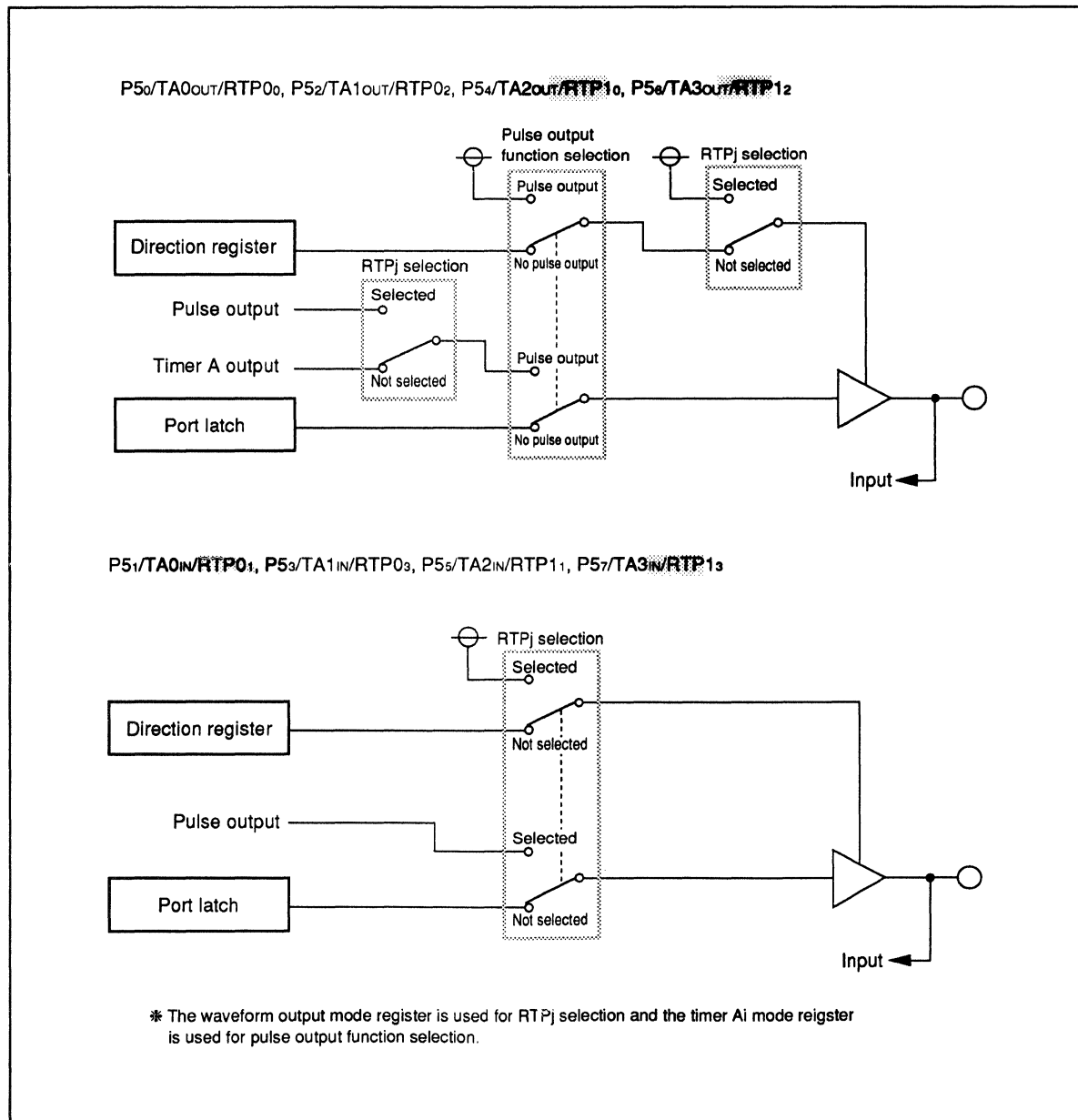


Fig. 9.5.5 RTP0₀–RTP1₃ output control block diagram

[Note]

With the M37730 group, the RTP1₃ pin is shared with P6₀/TA4_{OUT} pin rather than P5₇/TA3_{IN} pin. In addition, with the M37730 group, pulse output can be enabled/disabled with the waveform output control bit (bit 7 at address 62₁₆) and $\overline{\text{INT}}_0$ input (refer to section “CHAPTER 8. M37730 GROUP” for information on M37730 group).

9.6 Application system examples

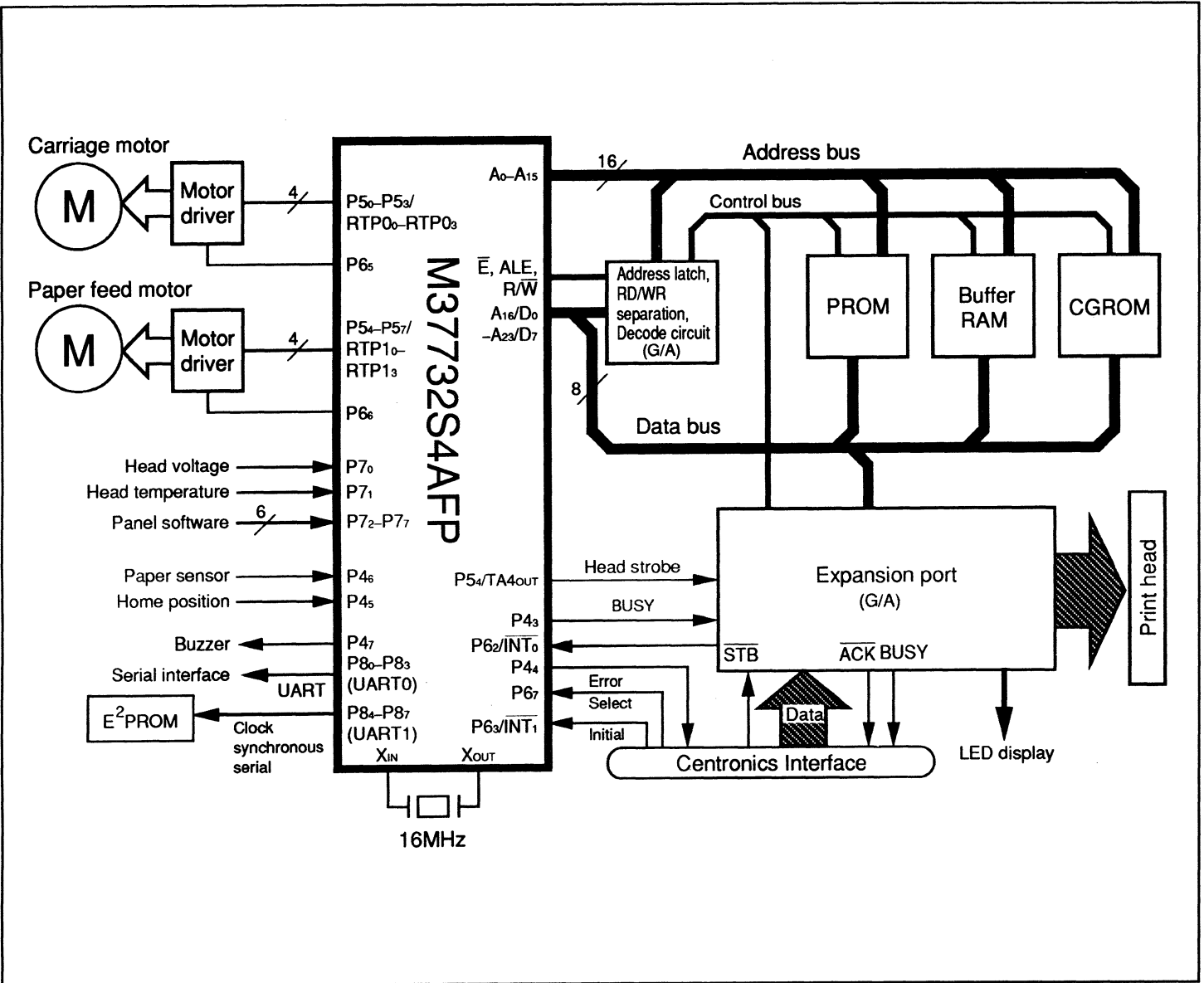


Fig. 9.6.1 Serial printer

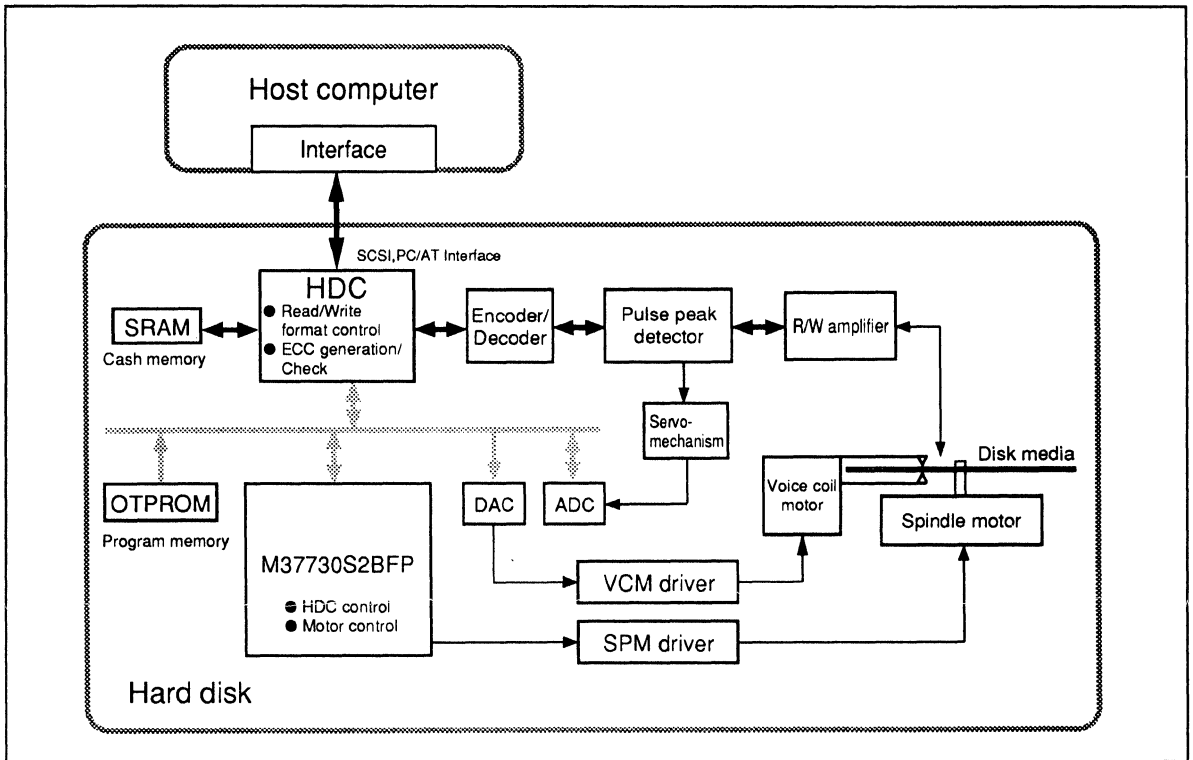


Fig. 9.6.2 Hard disk (HDD)

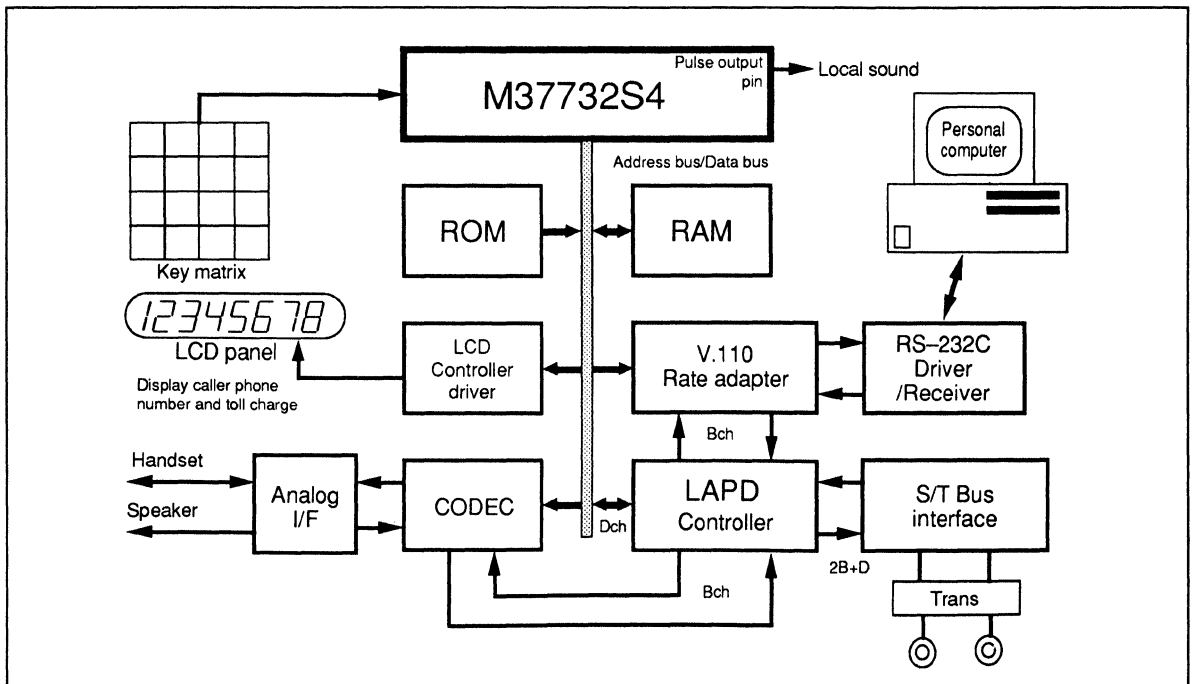


Fig. 9.6.3 ISDN digital telephone

9.7 Program examples

The program examples listed below are shown in the following pages. These program examples show how to set various registers when using each function. When using any of these program examples in actual application, be sure to modify it according to the type of application and evaluate the result carefully.

9.7.1 Hardware definition examples

9.7.2 Initialization examples

9.7.3 Timer setting examples

- (1) Timer A Timer mode (without pulse output)
- (2) Timer A Timer mode (with pulse output)
- (3) Timer A Event counter mode
- (4) Timer A Event counter mode (two-phase pulse signal processing function)
- (5) Timer A One-shot pulse mode (internal trigger)
- (6) Timer A One-shot pulse mode (external trigger)
- (7) Timer A PWM mode (16-bit PWM)
- (8) Timer A PWM mode (8-bit PWM)
- (9) Timer B Pulse period measurement mode
- (10) Timer B Pulse width measurement mode

9.7.4 Pulse output port mode setting examples

- (1) Without pulse width modulation
- (2) With pulse width modulation

9.7.5 Serial I/O modes setting examples

- (1) 8-bit UART (1-byte receive)
- (2) 8-bit UART (1-byte transmit)
- (3) 8-bit UART (n-byte transmit)
- (4) Clock synchronous (1-byte receive)
- (5) Clock synchronous (1-byte transmit)
- (6) Clock synchronous (n-byte transmit)
- (7) Error processing

9.7.6 A-D conversion modes setting examples

- (1) One-shot mode
- (2) Repeat mode
- (3) Single sweep mode
- (4) Repeat sweep mode

9.7.7 Interrupt processing examples

- (1) Interrupt setting examples
- (2) Interrupt processing example (When memory space is 64K bytes or less.)
- (3) Interrupt processing example (When memory space exceeds 64K bytes.)

9.7.8 Watchdog timer setting examples

- (1) Watchdog timer write routine
- (2) Watchdog timer interrupt processing example

9.7.9 Software timer setting examples

9.7.10 Interrupt vector setting example

9.7.1 Hardware definition examples

This is a setting example for a program that defines SFR (refer to page 24 for information on SFR).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...
12          1          .SECTION      SFR_AREA
13          1 ;
14          1 ;=====
15          1 ;      Port registers      =
16          1 ;=====
17          1 ;
18          1          .ORG      00000AH
19 00000A    1 P45:          ; Port P4, P5 registers
20 (00000A)  1H BYTE  1 P4:  .BLKB  1          ; Port P4 register
21 (00000B)  1H BYTE  1 P5:  .BLKB  1          ; Port P5 register
22          1 ;-----
23 00000C    1 P45D:         ; Port P4, P5 direction registers
24 (00000C)  1H BYTE  1 P4D: .BLKB  1          ; Port P4 direction register
25 (00000D)  1H BYTE  1 P5D: .BLKB  1          ; Port P5 direction register
26          1 ;-----
27 00000E    1 P67:          ; Port P6, P7 registers
28 (00000E)  1H BYTE  1 P6:  .BLKB  1          ; Port P6 register
29 (00000F)  1H BYTE  1 P7:  .BLKB  1          ; Port P7 register
30          1 ;-----
31 000010    1 P67D:         ; Port P6, P7 direction registers
32 (000010)  1H BYTE  1 P6D: .BLKB  1          ; Port P6 direction register
33 (000011)  1H BYTE  1 P7D: .BLKB  1          ; Port P7 direction register
34          1 ;-----
35 (000012)  2H BYTE  1 P8:  .BLKB  2          ; Port P8 register
36          1 ;-----
37 (000014)  1H BYTE  1 P8D: .BLKB  1          ; Port P8 direction register
38          1 ;

```

APPLICATION

9.7 Program examples

```
SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
41          1 ;=====
42          1 ;      A-D conversion registers          =
43          1 ;=====
44          1 ;
45          1      .ORG      00001EH
46 (00001E)  1H BYTE  1 ADCON: .BLKB  1          ; A-D control register
47 (00001F)  1H BYTE  1 ADSPS: .BLKB  1          ; A-D sweep pin selection register
48          1 ;-----
49 (000020)  2H BYTE  1 ADO:   .BLKB  2          ; A-D register 0
50          1 ;-----
51 (000022)  2H BYTE  1 AD1:   .BLKB  2          ; A-D register 1
52          1 ;-----
53 (000024)  2H BYTE  1 AD2:   .BLKB  2          ; A-D register 2
54          1 ;-----
55 (000026)  2H BYTE  1 AD3:   .BLKB  2          ; A-D register 3
56          1 ;-----
57 (000028)  2H BYTE  1 AD4:   .BLKB  2          ; A-D register 4
58          1 ;-----
59 (00002A)  2H BYTE  1 AD5:   .BLKB  2          ; A-D register 5
60          1 ;-----
61 (00002C)  2H BYTE  1 AD6:   .BLKB  2          ; A-D register 6
62          1 ;-----
63 (00002E)  2H BYTE  1 AD7:   .BLKB  2          ; A-D register 7
64          1 ;
65          1 ;=====
66          1 ;      Serial I/O 0 registers          =
67          1 ;=====
68          1 ;
69 000030    1 SOMx8:          ;
70 (000030)  1H BYTE  1 SOMR:  .BLKB  1          ; UART0 transmit/receive mode register
71 (000031)  1H BYTE  1 SOBRG: .BLKB  1          ; UART0 baud rate register
72          1 ;-----
73 000032    1 SOTB:          ; UART0 transmission buffer register
74 (000032)  1H BYTE  1 SOTBL: .BLKB  1          ; UART0 transmission buffer register (low-order)
75 (000033)  1H BYTE  1 SOTBH: .BLKB  1          ; UART0 transmission buffer register (high-order)
76          1 ;-----
77 000034    1 SOC:          ; UART0 transmit/receive control register
78 (000034)  1H BYTE  1 SOCL:  .BLKB  1          ; UART0 transmit/receive control register 0
79 (000035)  1H BYTE  1 SOCH:  .BLKB  1          ; UART0 transmit/receive control register 1
80          1 ;-----
81 000036    1 SORB:          ; UART0 receive buffer register
82 (000036)  1H BYTE  1 SORBL: .BLKB  1          ; UART0 receive buffer register (low-order)
83 (000037)  1H BYTE  1 SORBH: .BLKB  1          ; UART0 receive buffer register (high-order)
84          1 ;
85          1 ;=====
86          1 ;      Serial I/O 1 registers          =
87          1 ;=====
88          1 ;
89 000038    1 SIMx8:          ;
90 (000038)  1H BYTE  1 SIMR:  .BLKB  1          ; UART1 transmit/receive mode register
91 (000039)  1H BYTE  1 SIBRG: .BLKB  1          ; UART1 baud rate register
92          1 ;-----
93 00003A    1 SITB:          ; UART1 transmission buffer register
94 (00003A)  1H BYTE  1 SITBL: .BLKB  1          ; UART1 transmission buffer register (low-order)
95 (00003B)  1H BYTE  1 SITBH: .BLKB  1          ; UART1 transmission buffer register (high-order)
96          1 ;-----
97 00003C    1 SIC:          ; UART1 transmit/receive control register
98 (00003C)  1H BYTE  1 SICL:  .BLKB  1          ; UART1 transmit/receive control register 0
99 (00003D)  1H BYTE  1 SICH:  .BLKB  1          ; UART1 transmit/receive control register 1
100         1 ;-----
101 00003E    1 SIRB:          ; UART1 receive buffer register
102 (00003E)  1H BYTE  1 SIRBL: .BLKB  1          ; UART1 receive buffer register (low-order)
103 (00003F)  1H BYTE  1 SIRBH: .BLKB  1          ; UART1 receive buffer register (high-order)
104         1 ;
```

APPLICATION

9.7 Program examples

```

SEQ. LOC.  OBJ.      ....*...1....*...2....*...SOURCE STATEMENT....5....*...6....*...7....*...8....*...8....*...9....*...
107          1 ;=====
108          1 ;      Timer registers          =
109          1 ;=====
110          1 ;
111 (000040)  2H BYTE 1 TABSR: .BLKB 2          ; Count start flag
112          1 ;-----
113 (000042)  2H BYTE 1 ONSF:  .BLKB 2          ; One-shot start flag
114          1 ;-----
115 (000044)  2H BYTE 1 UDF:   .BLKB 2          ; Up-down flag
116          1 ;-----
117 000048    1 TAO:          ; Timer A0 register
118 (000048)  1H BYTE 1 TAOL:  .BLKB 1          ; Timer A0 register (low-order)
119 (000047)  1H BYTE 1 TAOH:  .BLKB 1          ; Timer A0 register (high-order)
120          1 ;-----
121 000048    1 TAI:          ; Timer A1 register
122 (000048)  1H BYTE 1 TAIL:  .BLKB 1          ; Timer A1 register (low-order)
123 (000049)  1H BYTE 1 TAIH:  .BLKB 1          ; Timer A1 register (high-order)
124          1 ;-----
125 000048    1 TA2:          ; Timer A2 register
126 (000048)  1H BYTE 1 TA2L:  .BLKB 1          ; Timer A2 register (low-order)
127 (000048)  1H BYTE 1 TA2H:  .BLKB 1          ; Timer A2 register (high-order)
128          1 ;-----
129 000048    1 TA3:          ; Timer A3 register
130 (000048)  1H BYTE 1 TA3L:  .BLKB 1          ; Timer A3 register (low-order)
131 (000048)  1H BYTE 1 TA3H:  .BLKB 1          ; Timer A3 register (high-order)
132          1 ;-----
133 000048    1 TA4:          ; Timer A4 register
134 (000048)  1H BYTE 1 TA4L:  .BLKB 1          ; Timer A4 register (low-order)
135 (000048)  1H BYTE 1 TA4H:  .BLKB 1          ; Timer A4 register (high-order)
136          1 ;-----
137 000050    1 TBO:          ; Timer B0 register
138 (000050)  1H BYTE 1 TBOL:  .BLKB 1          ; Timer B0 register (low-order)
139 (000051)  1H BYTE 1 TBOH:  .BLKB 1          ; Timer B0 register (high-order)
140          1 ;-----
141 000052    1 TB1:          ; Timer B1 register
142 (000052)  1H BYTE 1 TB1L:  .BLKB 1          ; Timer B1 register (low-order)
143 (000053)  1H BYTE 1 TB1H:  .BLKB 1          ; Timer B1 register (high-order)
144          1 ;-----
145 000054    1 TB2:          ; Timer B2 register
146 (000054)  1H BYTE 1 TB2L:  .BLKB 1          ; Timer B2 register (low-order)
147 (000055)  1H BYTE 1 TB2H:  .BLKB 1          ; Timer B2 register (high-order)
148          1 ;-----
149 000058    1 TAO1MR:       ; Timer A0, A1 mode registers
150 (000058)  1H BYTE 1 TAO1MR: .BLKB 1          ; Timer A0 mode register
151 (000057)  1H BYTE 1 TAI1MR: .BLKB 1          ; Timer A1 mode register
152          1 ;-----
153 000058    1 TA23MR:       ; Timer A2, A3 mode registers
154 (000058)  1H BYTE 1 TA2MR:  .BLKB 1          ; Timer A2 mode register
155 (000058)  1H BYTE 1 TA3MR:  .BLKB 1          ; Timer A3 mode register
156          1 ;-----
157 00005A    1 TA4B0MR:       ; Timer A4, B0 mode registers
158 (00005A)  1H BYTE 1 TA4MR:  .BLKB 1          ; Timer A4 mode register
159 (00005B)  1H BYTE 1 TBO1MR: .BLKB 1          ; Timer B0 mode register
160          1 ;-----
161 00005C    1 TB12MR:       ; Timer B1, B2 mode registers
162 (00005C)  1H BYTE 1 TB1MR:  .BLKB 1          ; Timer B1 mode register
163 (00005D)  1H BYTE 1 TB2MR:  .BLKB 1          ; Timer B2 mode register
164          1 ;

```


APPLICATION

9.7 Program examples

```

SEQ. LOC.  OBJ.      ....*...1....*...2....*...SOURCE STATEMENT...5....*...6....*...7....*...8....*...9....*...
187          1 ;=====
188          1 ; Processor mode register =
189          1 ;=====
190          1 ;
191          1 ;
171 (00005E) 2H BYTE 1 PMR: .BLKB 2 ; Processor mode register
172          1 ;
173          1 ;=====
174          1 ; Watchdog timer registers =
175          1 ;=====
176          1 ;
177          1 ;
177 (000080) 1H BYTE 1 WDT: .BLKB 1 ; Watchdog timer
178 (000081) 1H BYTE 1 WDC: .BLKB 1 ; Watchdog timer frequency selection flag
179          1 ;
180          1 ;=====
181          1 ; Pulse output port mode registers =
182          1 ;=====
183          1 ;
184 (000082) 2H BYTE 1 WOUTMR: .BLKB 2 ; Waveform output mode register
185 (000084) 1H BYTE 1 POUTDRI: .BLKB 1 ; Pulse output data register 1
186 (000085) 1H BYTE 1 POUTDRO: .BLKB 1 ; Pulse output data register 2
187          1 ;
188          1 ;=====
189          1 ; Interrupt control registers =
190          1 ;=====
191          1 ;
192          1 ; .ORG 000070H
193 000070    1 ADxSOTIC:
194 (000070) 1H BYTE 1 ADIC: .BLKB 1 ; A-D conversion interrupt control register
195 (000071) 1H BYTE 1 SOTIC: .BLKB 1 ; UART0 transmission interrupt control register
196          1 ;-----
197 000072    1 SORxSITIC:
198 (000072) 1H BYTE 1 SORIC: .BLKB 1 ; UART0 receive interrupt control register
199 (000073) 1H BYTE 1 SITIC: .BLKB 1 ; UART1 transmission interrupt control register
200          1 ;-----
201 000074    1 SIRxTAOIC:
202 (000074) 1H BYTE 1 SIRIC: .BLKB 1 ; UART1 receive interrupt control register
203 (000075) 1H BYTE 1 TAOIC: .BLKB 1 ; Timer A0 interrupt control register
204          1 ;-----
205 000076    1 TAlxTA2IC:
206 (000076) 1H BYTE 1 TAl1IC: .BLKB 1 ; Timer A1 interrupt control register
207 (000077) 1H BYTE 1 TA2IC: .BLKB 1 ; Timer A2 interrupt control register
208          1 ;-----
209 000078    1 TAsxTA4IC:
210 (000078) 1H BYTE 1 TAs1IC: .BLKB 1 ; Timer A3 interrupt control register
211 (000079) 1H BYTE 1 TA4IC: .BLKB 1 ; Timer A4 interrupt control register
212          1 ;-----
213 00007A    1 TBoxBT1IC:
214 (00007A) 1H BYTE 1 TBo1IC: .BLKB 1 ; Timer B0 interrupt control register
215 (00007B) 1H BYTE 1 TB1IC: .BLKB 1 ; Timer B1 interrupt control register
216          1 ;-----
217 00007C    1 TB2xINTOIC:
218 (00007C) 1H BYTE 1 TB2IC: .BLKB 1 ; Timer B2 interrupt control register
219 (00007D) 1H BYTE 1 INTOIC: .BLKB 1 ; INTO interrupt control register
220          1 ;-----
221 00007E    1 INT1xINT2IC:
222 (00007E) 1H BYTE 1 INT1IC: .BLKB 1 ; INT1 interrupt control register
223 (00007F) 1H BYTE 1 INT2IC: .BLKB 1 ; INT2 interrupt control register
224          1 ;-----
225          1 ;

```

9.7.2 Initialization examples

This is a setting example for a program that clears RAM and initializes stack pointer, direct page register, and data bank register.

```

SEQ. LOC.  OBJ.      ....*...1...*...2....*...SOURCE STATEMENT...5...*...8...*...7...*...8...*...8...*...
242          ;=====
243          ;      Initialization routine      =
244          ;=====
245          ;
246          ;-----
247          ;      Assembler declarations    =
248          ;-----
249          ;
250          ; These are assembler declarations.
251          ; Flags and registers used
252          ; in the program must match
253          ; these declarations.
254          ;
255          .SECTION      PROGRAM      ; Section name
256          .ORG          0C000H      ; Start address
257          .DATA        16          ; Data length
258          .INDEX       16          ; Index register length
259          .DP           0000H      ; Direct page
260          .DT           00H        ; Data bank
261          ;
262          ;-----
263          ;      Initialize registers and flags =
264          ;-----
265          ;
266          00C000      INITIAL:
267          00C000  78          SEI          ; Disable interrupt
268          00C001  C288       CLP          m,x,D ; Set data and index register length to 16 bits
269          ;                                     ; Binary operation mode
270          00C003  A27F08     LDX          #087FH ;
271          00C008  9A          TXS          ; Stack pointer=the highest address of RAM
272          00C007  A90000     LDA          A,#0 ;
273          00C00A  5B          TAD          ; Direct page=0H to FFH
274          00C00B  89C200     LDT          #0 ; Data bank=bank 0
275          00C00E  *845E2200  L          LDM          #0022H,PMR ; Interrupt priority detection time=2φ cycles
276          ;                                     ; Microprocessor mode
277          ;
278          ;-----
279          ;      Clear RAM                    =
280          ;-----
281          ;
282          00C012  A27E08     LDX          #087EH ;
283          00C015  A90000     LDA          A,#0 ; Set initial value (0H) in accumulator A
284          00C018      RAM_CLR:
285          00C018  *9500      STA          A,0,X ; Write contents of accumulator A to specified
286          00C01A  CA          DEX          ; RAM area in 16-bit units
287          00C01B  CA          DEX          ;
288          00C01C  E07E00     CPX          #07EH ;
289          00C01F  D0F7      L          BNE          RAM_CLR ;
290          ;
291          ;-----
292          ;      Change register model        =
293          ;      (if necessary)              =
294          ;-----
295          ;
296          .DATA        8          ; Data length
297          00C021  F8          SEM          ; Data length 8 bits
298          ;
299          ;
300          00C022  2000D0     L          JSR          SUB_T1_1 ; Initialization of timer and so on
301          00C025      E_LOOP:
302          00C025  80FE      L          BRA          E_LOOP ; Main routine
303          ;

```

9.7.3 Timer setting examples

(1) Timer A Timer mode (without pulse output)

This is a setting example for a program that sets the interrupt request bit to "1" every 500 μ s using timer A0 (refer to page 66 for information on timer mode).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....8....*....7....*....8....*....9....*....
328                1 ;=====
327                1 ;      「Timer A: Timer mode」          =
328                1 ;      Without pulse output          =
329                1 ;=====
330                1 ;      Timer      : Timer A0
331                1 ;      Mode      : Timer mode
332                1 ;      Pulse output : Disabled
333                1 ;      Gate function: Disabled
334                1 ;      Count source : f(XIN)/2
335                1 ;
336 00D000          1 SUB_T1_1:
337                1      .DATA 8
338 00D000 E224      1      SEP  #,I          ; Data length 8 bits, disable interrupt
339 00D002 #144001   LI      CLB  #00000001B,TABSR ; Stop timer A0 count
340 00D005 #645800  LI      LDM  #00000000B,TAOMR ; Timer mode, disable pulse output and gate function
341                1
342                1      .DATA 16
343 00D008 DE        1      CLM          ; Data length 16 bits
344 00D008 #6448CF07 LI      LDM  #2000-1,TA0 ; Set counter value (period:500  $\mu$ s at 8MHz)
345                1      .DATA 8
346 00D00D F8        1      SEM          ; Data length 8 bits
347 00D00E #14750F  LI      CLB  #00001111B,TAOIC ; Clear timer A0 interrupt request bit
348 00D011 #047507  LI      SEB  #00000111B,TAOIC ; Enable timer A0 interrupt (level 7)
349 00D014 #044001  LI      SEB  #00000001B,TABSR ; Start timer A0 count
350 00D017 58        1      CLI          ; Enable interrupt
351                1 ;
352                1 ;      ※ Hereafter, interrupt request bit is set to "1" every time timer A0 underflows.
353                1 ;
354 00D018 60        1      RTS

```

(2) Timer A Timer mode (with pulse output)

This is a setting example for a program that outputs the pulse of inverse phase every 500 μ s from the TA0_{OUT} pin using timer A0 (refer to page 66 for information on timer mode).

```

SEQ. LOC.  OBJ.      ....*....1.....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...
356          1 ;=====
357          1 ;      [Timer A: Timer mode]          =
358          1 ;      With pulse output          =
359          1 ;=====
360          1 ;      Timer      : Timer A0
361          1 ;      Mode      : Timer mode
362          1 ;      Pulse output : Enabled
363          1 ;      Gate function: Disabled
364          1 ;      Count source : f(XIN)/2
365          1 ;
366 00D018      1 SUB_T1_2:
367          1      .DATA 8
368 00D018 F8      1      SEM                      ; Data length 8 bits
369 00D01A *144001 L1      CLB #0000001B,TABSR      ; Stop timer A0 count
370 00D01D *845804 L1      LDM #0000100B,TAOMR      ; Timer mode, enable pulse output, disable gate function
371          1                      ; Count source=f(XIN)/2
372          1      .DATA 16
373 00D020 D8      1      CLM                      ; Data length 16 bits
374 00D021 *8448CF07 L1      LDM #2000-1,TAO          ; Set counter value (period:500  $\mu$ s at 8MHz)
375          1      .DATA 8
376 00D025 F8      1      SEM                      ; Data length 8 bits
377 00D028 *14750F L1      CLB #00001111B,TAOIC      ; Disable timer A0 interrupt
378 00D029 *044001 L1      SEB #00000001B,TABSR      ; Start timer A0 count
379          1 ;
380          1 ;      ※ Hereafter, phase of output pulse inverts every time timer A0 underflows.
381          1 ;
382 00D02C 80      1      RTS

```

(3) Timer A Event counter mode

This is a setting example for a program that sets the interrupt request bit to "1" for every 100 count of external clock input to the TA_{1IN} pin (refer to page 72 for information on event counter mode).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....8....*....7....*....8....*....9....*....
387          1 ;=====
388          1 ;      [Timer A: Event counter mode] =
389          1 ;=====
390          1 ;
391          1 ;      Timer      : Timer A1
392          1 ;      Mode      : Event counter mode
393          1 ;      Pulse output : Disabled
394          1 ;      Count polarity: Fall
395          1 ;      Up/down switch: Contents of up/down flag
396          1 ;
397          1 SUB_T2:
398          1      .DATA 8
399          1      SEP  #,I ; Data length 8 bits, disable interrupt
400          1      CLB  #00001000B,P5D ; Set P53/TA1IN pin to input mode
401          1      CLB  #00000010B,TABSR ; Stop timer A1 count
402          1      LDM  #00000001B,TA1MR ; Event counter mode, disable pulse output
403          1 ; ; Select falling edge count, up/down flag
404          1      CLB  #00000010B,UDF ; Set to decrement count
405          1      .DATA 16
406          1      CLM ; Data length 16 bits
407          1      LDM  #100-1,TA1 ; Set counter value
408          1      .DATA 8
409          1      SEM ; Data length 8 bits
410          1      CLB  #00001111B,TA1IC ; Clear timer A1 interrupt request bit
411          1      SEB  #00000011B,TA1IC ; Enable timer A1 interrupt (level 3)
412          1      SEB  #00000010B,TABSR ; Start timer A1 count
413          1      CLI ; Enable interrupt
414          1 ;
415          1 ;      * Hereafter, interrupt request bit is set to "1" at every 100 count.
416          1 ;
417          1      RTS

```

(4) Timer A Event counter mode (two-phase pulse signal processing function)

This is a setting example for a program that increments the count of the rising edge input to the TA2_{IN} pin and decrements the count of the falling edge input to the TA2_{IN} pin while the TA2_{OUT} pin level is "H" (refer to page 72 for information on event counter mode).

```

SEQ. LOC.  OBJ.          .....*....1....*....2....*....SOURCE STATEMENT....5....*....8....*....7....*....8....*....8....*....
422                1 ;=====
423                1 ;      [Timer A: Two-phase pulse signal processing function] =
424                1 ;=====
428                1 ;      Timer: Timer A2
427                1 ;      Mode : Event counter mode
428                1 ;      (two-phase pulse signal processing function)
429                1 ;
430 00D04C        1 SUB_T3:
431                1      .DATA 8
432 00D04C F8      1      SEM                ; Data length 8 bits
433 00D04D *140D80 LI      CLB #0011000B,P5D    ; Set P54/TA2OUT, P55/TA2IN pins to input mode
434 00D050 *144004 LI      CLB #00000100B,TABSR   ; Stop timer A2 count
435 00D053 *845811 LI      LDM #00010001B,TA2MR    ; Event counter mode
436 00D056 *844420 LI      LDM #00100000B,UDF     ; Select two-phase pulse signal processing function
437                1      .DATA 16
438 00D059 D8      1      CLM                ; Data length 16 bits
439 00D05A *844A0080 LI     LDM #8000H,TA2      ; Set counter value
440                1      .DATA 8
441 00D05E F8      1      SEM                ; Data length 8 bits
442 00D05F *14770F LI      CLB #00001111B,TA21C   ; Disable timer A2 interrupt
443 00D062 *044004 LI      SEB #00000100B,TABSR   ; Start timer A2 count
444                1 ;
445                1 ;      ※ Hereafter, the counter is incremented when rising edge is input to TA2IN pin and
446                1 ;      decremented when falling edge is input to TA2IN pin while TA2OUT pin level is "H".
447                1 ;
448 00D065 60      1      RTS

```

(5) Timer A One-shot pulse mode (Internal trigger)

This is a setting example for a program that outputs a 500 μ s width "H" level pulse for 1 pulse period from the TA3_{OUT} pin when a trigger (internal trigger) is generated with a one-shot start flag (refer to page 79 for information on one-shot pulse mode).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
452          1 ;=====
453          1 ;      [Timer A: One-shot pulse mode]      =
454          1 ;      When internal trigger is selected  =
455          1 ;=====
456          1 ;      Timer      : Timer A3
457          1 ;      Mode      : One-shot pulse mode
458          1 ;      Trigger   : One-shot start flag
459          1 ;      (Internal trigger)
460          1 ;      Count source: f(XIN)/2
461          1 ;
462 00D068      1 SUB_T4_1:
463          1      .DATA 8
464 00D068 F8      1      SEM                      ; Data length 8 bits
465 00D067 *144008  L1      CLB #0000100B,TA3SR      ; Stop timer A3 count
466 00D06A *845906  L1      LDM #0000110B,TA3MR      ; One-shot pulse mode, internal trigger
467          1 ;      ; Count source=f(XIN)/2
468          1      .DATA 16
469 00D06D D8      1      CLM                      ; Data length 16 bits
470 00D06E *844CD007 L1      LDM #2000,TA3      ; Set counter value (500  $\mu$ s at 8MHz)
471          1      .DATA 8
472 00D072 F8      1      SEM                      ; Data length 8 bits
473 00D073 *14780F  L1      CLB #00001111B,TA3IC      ; Disable timer A3 interrupt
474 00D076 *044008  L1      SEB #00001000B,TA3SR      ; Start timer A3 count
475 00D079 *844208  L1      LDM #00001000B,ONSF      ; Generate timer A3 one-shot trigger
476          1 ;
477          1 ;      ※ Hereafter, "H" level pulse with the specified width (500  $\mu$ s at 8MHz) is output.
478          1 ;
479 00D07C 80      1      RTS

```

(6) Timer A One-shot pulse mode (external trigger)

This is a setting example for a program that outputs a 500 μ s width "H" level pulse for 1 pulse period from the TA3_{OUT} pin when a falling edge (external trigger) is input to the TA3_{IN} pin (refer to page 79 for information on one-shot pulse mode).

```

SEQ. LOC.  OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
482                1 ;=====
483                1 ;      [Timer A: One-shot pulse mode]      =
484                1 ;      When external trigger is selected  =
485                1 ;=====
486                1 ;      Timer      : Timer A3
487                1 ;      Mode      : One-shot pulse mode
488                1 ;      Trigger   : Falling edge of TA3IN
489                1 ;      (External trigger)
490                1 ;      Count source: f(XIN)/2
491                1 ;
492 00D07D        1 SUB_T4_2:
493                1      .DATA  8
494 00D07D  F8    1      SEM                ; Data length 8 bits
495 00D07E *140D80  L1     CLB #10000000B,P5D ; Set P57/TA3IN pin to input mode
496 00D081 *144008  L1     CLB #00001000B,TABSR ; Stop timer A3 count
497 00D084 *845818  L1     LDM #00010110B,TA3MR ; One-shot pulse mode, external trigger
498                1 ;      ; Count source=f(XIN)/2
499                1 ;      .DATA 16
500 00D087  D8    1      CLM                ; Data length 16 bits
501 00D088 *844CD007 L1     LDM #2000,TA3 ; Set counter value (500  $\mu$ s at 8MHz)
502                1 ;      .DATA  8
503 00D08C  F8    1      SEM                ; Data length 8 bits
504 00D08D *14780F  L1     CLB #00001111B,TA3IC ; Disable timer A3 interrupt
505 00D090 *044008  L1     SEB #00001000B,TABSR ; Start timer A3 count
506                1 ;
507                1 ;      ※ Hereafter, "H" level pulse with the specified width (500  $\mu$ s at 8MHz) is output
508                1 ;      when falling edge is input to TA3IN pin.
509                1 ;
510 00D093  80    1      RTS

```


(7) Timer A PWM mode (16-bit PWM)

This is a setting example for a program that outputs a 16.3ms period 500 μ s width "H" level pulse from TA4_{OUT} pin using timer A4 (refer to page 84 for information on PWM mode).

```

SEQ. LOC.  OBJ.          .....1.....2.....SOURCE STATEMENT.....5.....6.....7.....8.....9.....
515          1 ;=====
516          1 ;      「Timer A: PWM mode」          =
517          1 ;      16-bit PWM                    =
518          1 ;=====
519          1 ;      Timer      : Timer A4
520          1 ;      Mode      : PWM mode
521          1 ;      Trigger   : Count start flag
522          1 ;      (Internal trigger)
523          1 ;      PWM mode  : 16-bit PWM mode
524          1 ;      Count source: f(XIN)/2
525          1 ;
526 00D094    1 SUB_T5_1:
527          1      .DATA 8
528 00D094 F8  1      SEM                                ; Data length 8 bits
529 00D095 *144010 L1    CLB #00010000B,TABSR          ; Stop timer A4 count
530 00D098 *645A07 L1    LDM #00000111B,TA4MR          ; 16-bit PWM mode, internal trigger
531          1 ;      ; Count source=f(XIN)/2 (period:16.3ms at 8MHz)
532          1      .DATA 16
533 00D09B D8  1      CLM                                ; Data length 16 bits
534 00D09C *644ED007 L1    LDM #2000,TA4                ; Set counter value ( "H" pulse width:500  $\mu$ s at 8MHz)
535          1      .DATA 8
536 00D0A0 F8  1      SEM                                ; Data length 8 bits
537 00D0A1 *14790F L1    CLB #00001111B,TA4IC          ; Disable timer A4 interrupt
538 00D0A4 *044010 L1    SEB #00010000B,TABSR          ; Start PWM output
539          1 ;
540          1 ;      ※ Hereafter, the set pulse is output. The pulse width of "H" level can be changed by
541          1 ;      modifying the counter value of timer A4.
542          1 ;
543 00D0A7 80  1      RTS

```

(8) Timer A PWM mode (8-bit PWM)

This is a setting example for a program that outputs a 12.8ms period 500μs width "H" level from the TA4_{OUT} pin using timer A4 (refer to page 84 for information on PWM mode).

```

SEQ. LOC.  OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
546                1 ; =====
547                1 ;      Timer A: PWM mode J           =
548                1 ;      8-bit PWM                   =
549                1 ; =====
550                1 ;      Timer      : Timer A4
551                1 ;      Mode       : PWM mode
552                1 ;      Trigger    : Count start flag
553                1 ;                      (Internal trigger)
554                1 ;      PWM mode   : 8-bit PWM mode
555                1 ;      Count source: f(XIN)/2
556                1 ;
557 00D0A8          1 SUB_T5_2:
558                1      .DATA 8
559 00D0A8 F8        1      SEM                      ; Data length 8 bits
560 00D0A9 *144010   L1     CLB #00010000B,TABSR      ; Stop timer A4 count
561 00D0AC *845A27   L1     LDM #00100111B,TA4MR     ; 8-bit PWM mode, internal trigger
562                1 ;                      ; Count source=f(XIN)/2
563 00D0AF *844EC7   L1     LDM #200-1,TA4L          ; Set prescaler value (period:12.8ms at 8MHz)
564 00D0B2 *844FOA   L1     LDM #10,TA4H            ; Set counter value < "H" pulse width:500 μs at 8MHz)
565 00D0B5 *14790F   L1     CLB #00001111B,TA4IC     ; Disable timer A4 interrupt
566 00D0B8 *044010   L1     SEB #00010000B,TABSR      ; Start PWM output
567                1 ;
568                1 ;      ※ Hereafter, the set pulse is output. The pulse width of "H" level can be changed by
569                1 ;      modifying the counter value (high-order 8 bits) of timer A4.
570                1 ;
571 00D0BB 80        1      RTS

```

(9) Timer B Pulse period measurement mode

This is a setting example for a program that measures the input signal pulse period by counting the interval from the falling edge to the falling edge of the signal input to the TB1_{IN} pin using timer B1 (refer to page 100 for information on pulse period measurement mode).

```

SEQ. LOC.  OBJ.          ....*....1.....*....2....*....SOURCE STATEMENT....5....*....8.....*....7....*....8.....*....8.....*....9....*...

576          1 ;=====
577          1 ;      [Timer B: Pulse period measurement mode] =
578          1 ;=====
579          1 ;      Timer      : Timer B1
580          1 ;      Mode      : Pulse period measurement mode
581          1 ;                      (between falling edge and falling edge)
582          1 ;      Count source: f(XIN)/84
583          1 ;
584          1 ;      ※ Timer B1 overflow period is assumed to be sufficiently
585          1 ;      longer than measured pulse period.
586          1 ;
587 00D0BC    1 SUB_T6:
588          1      .DATA 8
589 00D0BC F8  1      SEM                      ; Data length 8 bits
590 00D0BD *141040 L1     CLB #01000000B,P6D      ; Set P6B/TB1IN pin to input mode
591 00D0C0 *144040 L1     CLB #01000000B,TABSR     ; Stop timer B1 count
592 00D0C3 *645C82 L1     LDM #10000010B,TB1MR    ; Pulse period measurement mode, falling edge is effective
593          1 ;                      ; Count source=f(XIN)/84
594 00D0C6 *147B0F L1     CLB #00001111B,TB1IC    ; Clear timer B1 interrupt request bit
595          1 ;                      ; Disable timer B1 interrupt
596 00D0C9 *044040 L1     SEB #01000000B,TABSR     ; Start timer B1 count
597          1 ;
598          1 ;      ※ Hereafter, the counter is incremented. And the contents of the counter is transferred
599          1 ;      to the reload register and the counter is cleared when falling edge is input to TB1IN pin.
600 00D0CC    1 L1_T6:
601 00D0CC *347B08FC L1     BBC #00001000B,TB1IC,L1_T6 ; Timer B1 interrupt request bit= "1"?
602          1      .DATA 16
603 00D0D0 D8  1      CLM                      ; Data length 16 bits
604 00D0D1 *A552  L1     LDA A,TB1              ; Read measurement result in accumulator A
605          1      .DATA 8
606 00D0D3 F8  1      SEM                      ; Data length 8 bits
607          1 ;
608 00D0D4 80  1      RTS

```

(10) Timer B Pulse width measurement mode

This is a setting example for a program that measures the input signal pulse width by counting the interval from the falling edge to the rising edge and rising edge to the falling edge of the signal input to the TB2IN pin using timer B2 (refer to page 100 for information on pulse width measurement mode).

```

SEQ. LOC.  OBJ.      .....1.....2.....SOURCE STATEMENT.....5.....8.....7.....8.....9.....
812          1 ;=====
813          1 ;      「Timer B: Pulse width measurement mode」 =
814          1 ;=====
815          1 ;      Timer      : Timer B2
816          1 ;      Mode      : Pulse width measurement mode
817          1 ;      Count source: f(XIN)/64
818          1 ;
819          1 ;      ※ This program is a measurement example of the signal "H" pulse width
820          1 ;      input to the TB2IN. The timer B2 overflow period is assumed to be sufficiently
821          1 ;      longer than measured pulse width.
822          1 ;
823 OOD0D5    1 SUB_T7:
824          1      .DATA 8
825 OOD0D5 F8  1      SEM                               ; Data length 8 bits
826 OOD0D6 *141080 L1     CLB #1000000B,P6D           ; Set P67/TB2IN pin to input mode
827 OOD0D9 *144080 L1     CLB #1000000B,TABSR        ; Stop timer B2 count
828 OOD0DC *645D8A L1     LDM #10001010B,TB2MR       ; Pulse width measurement mode
829          1 ;      Count source=f(XIN)/64
830 OOD0DF *147C0F L1     CLB #00001111B,TB2IC       ; Clear timer B2 interrupt request bit
831          1 ;      Disable timer B2 interrupt
832 OOD0E2 *044080 L1     SEB #1000000B,TABSR        ; Enable timer B2 count
833 OOD0E5          1 L1_T7:
834 OOD0E5 *240E80FC L1     BBS #1000000B,P6,L1_T7     ; TB2IN= "L" ?
835 OOD0E9          1 L2_T7:
836 OOD0E9 *340E80FC L1     BBC #1000000B,P6,L2_T7     ; TB2IN= "H" ?
837 OOD0ED *147C08 L1     CLB #00001000B,TB2IC       ; Clear interrupt request bit
838 OOD0F0          1 L3_T7:
839 OOD0F0 *347C08FC L1     BBC #00001000B,TB2IC,L3_T7    ; Interrupt request bit= "1" ?
840 OOD0F4 *147C08 L1     CLB #00001000B,TB2IC       ; Clear interrupt request bit
841          1      .DATA 18
842 OOD0F7 D8  1      CLM                               ; Data length 18 bits
843 OOD0F8 *A554 L1     LDA A,TB2                       ; Read measurement result
844          1      .DATA 8
845 OOD0FA F8  1      SEM                               ; Data length 8 bits
846          1 ;
847 OOD0FB 80  1      RTS
    
```

9.7.4 Pulse output port mode setting examples

(1) Without pulse width modulation

This is a setting example for a program that outputs the contents set in pulse output data register 1 from RTP1 (P5₄, P5₅, P5₆, P5₇) each time timer A2 underflows (refer to page 104 and section "9.5 Pulse motor control application" for information on pulse output port mode).

```

SEQ. LOC.  OBJ.      ....*....1...*....2....*....SOURCE STATEMENT...5...*....8....*....7....*....8....*....9....*...
852          1 ;=====
853          1 ;      [Pulse output port mode]      =
854          1 ;      Without pulse width modulation  =
855          1 ;=====
856          1 ;      Pulse output port: RTP1 (ports P54,P55,P56,P57)
857          1 ;      Timer          : Timer A2
858          1 ;      Mode          : Timer mode
859          1 ;      Count source   : f(XIN)/2
860          1 ;
861 00D0FC      1 SUB_P1_1:
862 00D0FC 78      1      SEI          ; Disable interrupt
863          1
864          1      .DATA 8
865 00D0FD F8      1      SEM
866 00D0FE *144004 L1      CLB #0000100B,TABSR ; Stop timer A2 count
867 00D101 *845804 L1      LDM #0000100B,TA2MR ; Timer mode, enable pulse output, disable gate function
868 00D104 *845804 L1      LDW #0000100B,TA3MR ; Enable pulse output (set P56 to output)
869 00D107 *846201 L1      LDM #0000001B,WOUTMR ; Disable pulse width modulation, select RTP1
870 00D10A *84840F L1      LDM #00001111B,POUTDR1 ; Set initial value of pulse output data register 1 (P54,P55,P56,P57)
871 00D10D *14770F L1      CLB #00001111B,TA2IC ; Clear timer A2 interrupt request bit
872 00D110 *047701 L1      SEB #0000001B,TA2IC ; Enable timer A2 interrupt (level 1)
873          1
874          1      .DATA 16
875 00D113 D8      1      CLM
876 00D114 *844A0200 L1      LDM #0002H,TA2 ; Initial output of timer A2
877          1
878          1      .DATA 8
879 00D118 F8      1      SEM
880 00D119 *044004 L1      SEB #0000100B,TABSR ; Start timer A2 count
881 00D11C      1 LOOP_1:
882 00D11C *347708FC L1      BBC #0000100B,TA2IC,LOOP_1 ; Initial output complete?
883 00D120 *144004 L1      CLB #0000100B,TABSR ; Stop count
884          1
885          1      .DATA 16
886 00D123 D8      1      CLM
887 00D124 *844AFF00 L1      LDM #00FFH,TA2 ; Set timer A2 initial value (period:32 μs at 16MHz)
888          1
889          1      .DATA 8
890 00D128 F8      1      SEM
891 00D129 *147708 L1      CLB #0000100B,TA2IC ; Clear timer A2 interrupt request bit
892 00D12C *044004 L1      SEB #0000100B,TABSR ; Start timer A2 count
893 00D12F 58      1      CLI          ; Enable interrupt
894          1
895          1 ;      ※ Hereafter, the contents of the pulse output data register is output to port each time
896          1 ;      timer A2 underflows. The contents of pulse output data register and timer A2 register are
897          1 ;      updated while processing the timer A2 interrupt.
898          1
899 00D130 60      1      RTS

```

(2) With pulse width modulation

This is a setting example for a program that applies pulse width modulation by timer A3 to the contents in pulse output data register 1 and outputs the result to RTP1 (P54, P55, P56, P57) each time timer A2 underflows. Pulse width modulation is performed for pulse output when the contents of the pulse output data register 1 is "1" (refer to page 104 and section "9.5 Pulse motor control application" for information on pulse output port mode).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
703          1 ;=====
704          1 ;      [Pulse output port mode]      =
705          1 ;      With pulse width modulation  =
706          1 ;=====
707          1 ;      Pulse output port: RTP1 (ports P54,P55,P56,P57)
708          1 ;      Timer          : Timer A2, A3
709          1 ;      Mode          : Timer mode (timer A2)
710          1 ;      : Pulse width modulation mode (timer A3)
711          1 ;      Count source   : f(XIN)/2
712          1 ;
713 00D131    1 SUB_P1_2:
714 00D131 78 1      SEI          ; Disable interrupt
715          1
716          1      .DATA 8
717 00D132 F8 1      SEM
718          1
719 00D138 *14400C L1      CLB #00001100B,TABSR ; Stop timer A2 and timer A3 count
720 00D138 *845804 L1      LDM #0000100B,TA2MR ; Timer mode, enable pulse output, disable gate function
721 00D139 *845827 L1      LDM #00100111B,TA3MR ; Pulse width modulation mode (8-bit length)
722 00D13C *848221 L1      LDM #00100001B,WOUTMR ; Enable pulse width modulation, select RTP1
723 00D13F *84840F L1      LDM #00001111B,POUTDR1 ; Set initial value of pulse output data register 1
724          1 ; (P54,P55,P56,P57)
725 00D142 *14770F L1      CLB #00001111B,TA2IC ; Clear timer A2 interrupt request bit
726 00D145 *14780F L1      CLB #00001111B,TA3IC ; Disable timer A3 interrupt
727 00D148 *047701 L1      SEB #00000001B,TA2IC ; Enable timer A2 interrupt (level 1)
728          1
729          1      .DATA 16 ;
730 00D14B D8 1      CLM
731 00D14C *844A0200 L1      LDM #0002H,TA2 ; Initial output of timer A2
732          1
733          1      .DATA 8
734 00D150 F8 1      SEM
735 00D151 *044004 L1      SEB #00000100B,TABSR ; Start timer A2 count
736 00D154          1 LOOP_2:
737 00D154 *347708FC L1      BBC #00001000B,TA2IC,LOOP_2 ; Initial output complete?
738 00D158 *144004 L1      CLB #00000100B,TABSR ; Stop count
739          1
740          1      .DATA 16
741 00D15B D8 1      CLM
742 00D15C *844AFF07 L1      LDM #07FFH,TA2 ; Set timer A2 initial value (period:255 μs at 16MHz)
743 00D160 *844C007F L1      LDM #7FO0H,TA3 ; Set timer A3 initial value (period:31.8 μs at 16MHz,
744          1 ; "H" pulse width: 15.8 μs at 16MHz)
745          1
746          1      .DATA 8
747 00D184 F8 1      SEM
748 00D185 *147708 L1      CLB #00001000B,TA2IC ; Clear timer A2 interrupt request bit
749 00D188 *04400C L1      SEB #00001100B,TABSR ; Start timer A2 and timer A3 count
750 00D18B 58 1      CLI ; Enable interrupt
751          1
752          1 ; * Hereafter, the results of pulse width modulation by timer A3 is output to port each time
753          1 ; timer A2 underflows. The contents of pulse output data register and each timer register are
754          1 ; updated while processing the timer A2 interrupt.
755          1
756 00D18C 80 1      RTS

```

9.7.5 Serial I/O modes setting examples

(1) 8-bit UART (1-byte receive)

This is a setting example for a program that receives 1 byte at 8-bit UART of UART1 (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.      .....1.....2.....SOURCE STATEMENT.....5.....6.....7.....8.....9.....
776          1 ; =====
777          1 ;      8-bit UART (1-byte receive)J      =
778          1 ; =====
779          1 ;      Channel      : UART1
780          1 ;      Mode        : 8-bit UART
781          1 ;      Stop bit length: 1 bit
782          1 ;      Parity       : None
783          1 ;      Clock       : Internal (9800bps)
784          1 ;      Sleep function : Disabled
785          1 ;
786          1 ;
787          1 ;      8-bit UART (1-byte receive) programming model
788          1 ;
789          1 ;
790          1 ;      DATA ----->| RxD1 |
791          1 ;
792          1 ;      <-----| RTS |
793          1 ;
794          1 ;
795          1 ;      9800bps
796          1 ;
797 OOD800      1 SUB_S1:
798          1 ;      .DATA 8
799 OOD800 F8      1 SEM ; Data length 8 bits
800 OOD801 *141440 LI CLB #0100000B,P8D ; Set P86/RXD1 pin to input mode
801 OOD804 *843805 LI LDM #00000101B,S1MR ; 8-bit UART, internal clock, 1 stop bit
802          1 ; ; No parity, disable sleep mode
803 OOD807 *843C04 LI LDM #00000100B,S1CL ; BRG count source=f(XIN)/2
804          1 ; ; Select RTS function
805 OOD80A *843818 LI LDM #25,S1BRG ; Set value in baud rate register (9815bps at 8MHz)
806 OOD80D *843D04 LI LDM #00000100B,S1CH ; Enable receive
807          1 ;
808          1 ; * Receive starts when RTS output becomes "L" (receive enabled) and
809          1 ; a start bit is detected.
810          1 ;
811 OOD810      1 L1_S1:
812 OOD810 *343D08FC LI BBC #00001000B,S1CH,L1_S1 ; Receive complete (receive completion flag= "1")?
813 OOD814 *343D8003 LI BBC #1000000B,S1CH,L2_S1 ; Check error (error sum flag)
814 OOD818 20BAD8 LI JSR SUB_ERR0 ; Jump to UART receive error processing routine
815 OOD81B      1 L2_S1:
816 OOD81B *A53E LI LDA A,S1RB ; Read receive buffer
817 OOD81D *143D04 LI CLB #00000100B,S1CH ; Disable receive
818 OOD820 80      1 RTS
819          1 ;

```

(2) 8-bit UART (1-byte transmit)

This is a setting example for a program that transmits 1 byte at 8-bit UART of UART0 (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
822          1 ;=====
823          1 ;      8-bit UART (1-byte transmit)      =
824          1 ;=====
825          1 ;      Channel      : UART0
826          1 ;      Mode        : 8-bit UART
827          1 ;      Stop bit length: 1 bit
828          1 ;      Parity      : None
829          1 ;      Clock      : Internal (9600bps)
830          1 ;      Sleep function : Disabled
831          1 ;
832          1 ;      8-bit UART (1-byte transmit) programming model
833          1 ;      M37732
834          1 ;      +-----+
835          1 ;      |         |
836          1 ;      | TxDO |-----> DATA
837          1 ;      |     |
838          1 ;      | CTSO |<-----+
839          1 ;      |         |
840          1 ;      |         | --+ GND
841          1 ;      +-----+
842          1 ;      9600bps
843          1 ;
844 00D821    1 SUB_S2:
845          1      .DATA 8
846 00D821 F8      1      SEM                                ; Data length 8 bits
847 00D822 *141401 L1     CLB #00000001B,P8D                ; Set P80/CTS0 pin to input mode
848 00D825 *643005 L1     LDM #00000101B,SOMR              ; 8-bit UART, internal clock, 1 stop bit
849          1 ; ; No parity, disable sleep mode
850 00D828 *643400 L1     LDM #00000000B,SOCL              ; BRG count source=f(XIN)/2
851          1 ; ; Select CTS function
852 00D82B *643119 L1     LDM #25,SOBRG                    ; Set value in baud rate register (9615bps at 8MHz)
853 00D82E *643501 L1     LDM #00000001B,SOCH              ; Enable transmit
854 00D831 *A582    1      LDA A,T_DATA                    ; Read transmit data (T_DATA)
855 00D833 *8532    L1     STA A,SOTB                      ; Transmit data->transmission buffer
856 00D835 *143501 L1     CLB #00000001B,SOCH              ; Disable transmit
857 00D838 60      1      RTS

```


APPLICATION

9.7 Program examples

(3) 8-bit UART (n-byte transmit)

This is a setting example for a program that transmits n bytes at 8-bit UART of UART0 (refer to page 122 for information on serial I/O).

```
SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....8....*....7....*....8....*....8....*....

861          1 ;=====
862          1 ;      「8-bit UART (n-byte transmit)」 =
863          1 ;=====
864          1 ;      Channel      : UART0
865          1 ;      Mode        : 8-bit UART
866          1 ;      Stop bit length: 1 bit
867          1 ;      Parity      : None
868          1 ;      Clock      : Internal (9600bps)
869          1 ;      Sleep function : Disabled
870          1 ;
871          1 ;      8-bit UART (n-byte transmit) programming model
872          1 ;      M37732
873          1 ;      +-----+
874          1 ;      |      |
875          1 ;      | TxDO |-----> DATA
876          1 ;      |      |
877          1 ;      | ---- |
878          1 ;      | CTSO |<----- RTS
879          1 ;      |      |
880          1 ;      +-----+
881          1 ;      9600bps
882          1 ;
883          1 ;-----
884          1 ;      Prepare transmit =
885          1 ;-----
886          1 ;
887 00D839    1 SUB_S3:
888          1      .DATA 8
889          1      .INDEX 8
890 00D839 E230 1 SEP      m,x          ; Data length, index register length 8 bits
891 00D83B *141401 L1 CLB      #00000001B,P8D      ; Set P80/CTS0 pin to input mode
892 00D83E *843005 L1 LDM      #00000101B,SOMR      ; 8-bit UART, internal clock, 1 stop bit
893          1          ; No parity, disable sleep mode
894 00D841 *843400 L1 LDM      #00000000B,SOCL      ; BRG count source=f(XIN)/2
895          1          ; Select RTS function
896 00D844 *843119 L1 LDM      #25,SOBRG          ; Set value in baud rate register (9615bps at 8MHz)
897 00D847 *843501 L1 LDM      #00000001B,SOCH      ; Enable transmit
898          1 ;
899          1 ;-----
900          1 ;      Operate transmit =
901          1 ;-----
902          1 ;
903 00D84A A200 1 LDX      #0          ;
904 00D84C          1 L1_S3:
905 00D84C *B582 1 LDA      A,T_DATA,X          ; Read transmit data
906 00D84E *8532 L1 STA      A,SOTB          ; Transmit data->transmission buffer
907 00D850          1 L2_S3:
908 00D850 *343502FC L1 BFC      #00000010B,SOCH,L2_S3 ; Transmission buffer empty flag= "1" ?
909 00D854 E8 1 INX
910 00D855 *E480 1 CPX      DAT_CNT          ; Compare with transfer data count (DAT_CNT)
911 00D857 D0F3 L1 BNE      L1_S3          ; Continue data transmit?
912          1 ;
913          1 ;-----
914          1 ;      Complete transmit =
915          1 ;-----
916          1 ;
917 00D859 *143501 L1 CLB      #00000001B,SOCH      ; Disable transmit
918          1 ;
919 00D85C 80 1 RTS
```

(4) Clock synchronous (1-byte receive)

This is a setting example for a program that receives 1 byte at clock synchronous of UART1 (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
924          1 ;=====
925          1 ;      「Clock synchronous (1-byte receive)」 =
926          1 ;=====
927          1 ;      Channel: UART1
928          1 ;      Mode  : Clock synchronous
929          1 ;      Clock : External
930          1 ;
931          1 ;      Clock synchronous (1-byte receive) programming model
932          1 ;      M37732
933          1 ;      +-----+
934          1 ;      | RxD1 |<-----
935          1 ;      |      |
936          1 ;      | CLK1 |<----- CLOCK
937          1 ;      | ---- |
938          1 ;      | RTS1 |-----> CTS
939          1 ;      |      |
940          1 ;      +-----+
941          1 ;
942 00D85D      1 SUB_S4:
943          1      .DATA 8
944 00D85D F8      1      SEM                      ; Data length 8 bits
945 00D85E *141480  L1      CLB #01100000B,P8D      ; Set P86/RXD1, P85/CLK1 pins to input mode
946 00D861 *643809  L1      LDM #00001001B,S1MR      ; Select clock synchronous, external clock
947 00D864 *643C04  L1      LDM #00000100B,S1CL      ; Select RTS function
948 00D867 *643A55  L1      LDM #55H,S1TB      ; Set transmit dummy data
949          1 ;
950          1 ;      ※ With clock synchronous serial I/O, the transmitter
951          1 ;      must be operated even if only receive is performed.
952          1 ;
953 00D86A *643D05  L1      LDM #00000101B,S1CH      ; Enable receive/enable transmit
954          1 ;
955          1 ;      ※ Hereafter, receive operation starts when synchronous clock
956          1 ;      is input to CLK1 pin.
957          1 ;
958 00D86D          1 L1_S4:
959 00D86D *343D08FC L1      BBC #00001000B,S1CH,L1_S4 ; Receive completion flag= "1" ?
960 00D871 *343D1003 L1      BBC #00010000B,S1CH,L2_S4 ; Overrun error?
961 00D875 20C1D8  L1      JSR SUB_ERR1      ; Jump to clock synchronous error processing routine
962 00D878          1 L2_S4:
963 00D878 *A53E      L1      LDA A,S1RB      ; Read received result
964 00D87A *143D05  L1      CLB #00000101B,S1CH      ; Disable receive/disable transmit
965          1 ;
966 00D87D 80          1      RTS

```

(5) Clock synchronous (1-byte transmit)

This is a setting example for a program that transmits 1 byte at clock synchronous of UART0 (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
971      1 ; =====
972      1 ;      [Clock synchronous (1-byte transmit)] =
973      1 ; =====
974      1 ;      Channel: UART0
975      1 ;      Mode : Clock synchronous
976      1 ;      Clock : Internal (2Mbps)
977      1 ;
978      1 ;      Clock synchronous (1-byte transmit) programming model
979      1 ;      M37782
980      1 ;      +-----+
981      1 ;      |      |
982      1 ;      | TxDO |-----> DATA
983      1 ;      |      |
984      1 ;      | CLKO |-----> CLOCK(2Mbps)
985      1 ;      |      |
986      1 ;      |      |
987      1 ;      | CTSO |<-----+
988      1 ;      |      |
989      1 ;      |      |      +-+ GND
990      1 ;      +-----+
991      00D87E      1 SUB_S5:
992      1          .DATA 8
993      00D87E  F8      1 SEM          ; Data length 8 bits
994      00D87F *141401  L1 CLB #00000001B,P8D      ; Set P80/CTS0 pin to input mode
995      00D882 *843001  L1 LDM #00000001B,S0MR      ; Clock synchronous, internal clock
996      00D885 *843400  L1 LDM #00000000B,S0CL      ; BRG count source=f(XIN)/2
997      1          ; Select CTS function
998      00D888 *843100  L1 LDM #0,S0BRG      ; Set value in baud rate register (2Mbps at 8MHz)
999      00D88B *843501  L1 LDM #00000001B,S0CH      ; Enable transmit
1000     00D88E *A582      1 LDA A,T_DATA      ; Read transmit data (T_DATA)
1001     00D890 *8582      L1 STA A,S0TB      ; Transmit data->transmission buffer
1002     00D892 *143501  L1 CLB #00000001B,S0CH      ; Disable transmit
1003      1 ;
1004     00D895 80      1 RTS

```

(6) Clock synchronous (n-byte transmit)

This is a setting example for a program that transmits n bytes at clock synchronous of UART0 (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
1008      1 ;=====
1009      1 ;      [Clock synchronous (n-byte transmit)] =
1010      1 ;=====
1011      1 ;      Channel: UART0
1012      1 ;      Mode : Clock synchronous
1013      1 ;      Clock : Internal (2Mbps)
1014      1 ;
1015      1 ;      Clock synchronous (n-byte transmit) programming model
1016      1 ;      M37732
1017      1 ;      +-----+
1018      1 ;      | TxDO |-----> DATA
1019      1 ;      |   |
1020      1 ;      | CLKO |-----> CLOCK(2Mbps)
1021      1 ;      |   |
1022      1 ;      | CTSO |<----- RTS
1023      1 ;      |   |
1024      1 ;      +-----+
1025      1 ;
1026      1 ;-----
1027      1 ;      Prepare transmit =
1028      1 ;-----
1029      1 ;
1030 00D898 1 SUB_S8:
1031      1      .DATA 8
1032      1      .INDEX 8
1033 00D898 E230 1 SEP x,m ; Data length, index register length 8 bits
1034 00D898 *141401 LI CLB #0000001B,P8D ; Set P80/CTS0 pin to input mode
1035 00D898 *643001 LI LDM #0000001B,SOMR ; Clock synchronous, internal clock
1036 00D898 *643400 LI LDM #00000008,SOCL ; BRG count source=f(XIN)/2
1037      1 ; ; Select CTS function
1038 00D8A1 *643100 LI LDM #0,SOBRG ; Set value in baud rate register (2Mbps at 8MHz)
1039 00D8A4 *643501 LI LDM #0000001B,SOCH ; Enable transmit
1040      1 ;
1041      1 ;-----
1042      1 ;      Transmit data =
1043      1 ;-----
1044      1 ;
1045 00D8A7 A200 1 LDX #0
1046 00D8A9 1 L1_S8:
1047 00D8A9 *B582 1 LDA A,T_DATA,X ; Read transmit data
1048 00D8AB *8532 LI STA A,SOTB ; Transmit data->transmission buffer
1049 00D8AD 1 L2_S8:
1050 00D8AD *343502FC LI BBC #0000010B,SOCH,L2_S8 ; Transmission buffer empty flag= "1" ?
1051 00D8B1 E8 1 INX
1052 00D8B2 *E480 1 CPX DAT_CNT ; Compare with transfer data count (DAT_CNT)
1053 00D8B4 D0F8 LI BNE L1_S8 ; Continue data transmit?
1054      1 ;
1055      1 ;-----
1056      1 ;      Complete transmit =
1057      1 ;-----
1058      1 ;
1059 00D8B6 *143501 LI CLB #0000001B,SOCH ; Disable transmit
1060      1 ;
1061 00D8B9 80 1 RTS

```

(7) Error processing

This is a setting example for a program that clears the error flag by setting the receive enable bit to "0" (receive disabled) after error processing for UART receive or clock synchronous receive and then enables the reception again (refer to page 122 for information on serial I/O).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...
1085      1 ;=====
1086      1 ;      「Error processing」 =
1087      1 ;=====
1088      1 ;
1089      1 ;-----
1070      1 ;      UART receive error processing =
1071      1 ;-----
1072      1 ;
1073 00D8BA 1 SUB_ERRO:
1074      1 ;      :
1075      1 ;      : ; Check error type
1076      1 ;      : ; (overrun, framing, parity)
1077      1 ;      : ; Each error processing
1078      1 ;      :
1079      1 ;
1080 00D8BA *143D04 L1      CLB      #00000100B,S1CH      ; Disable receive
1081      1 ; ; (Clear error flag)
1082 00D8BD *043D04 L1      SEB      #00000100B,S1CH      ; Enable receive
1083 00D8C0 80      1 ;      RTS
1084      1 ;
1085      1 ;-----
1086      1 ;      Clock synchronous receive error processing =
1087      1 ;-----
1088      1 ;
1089 00D8C1 1 SUB_ERR1:
1090      1 ;      :
1091      1 ;      : ; Overrun error processing
1092      1 ;      :
1093      1 ;      :
1094      1 ;
1095 00D8C1 *143D04 L1      CLB      #00000100B,S1CH      ; Disable receive
1096      1 ; ; (Clear error flag)
1097 00D8C4 *043D04 L1      SEB      #00000100B,S1CH      ; Enable receive
1098      1 ;
1099 00D8C7 80      1 ;      RTS

```

9.7.6 A-D conversion modes setting examples

(1) One-shot mode

This is a setting example for a program that performs an A-D conversion of input voltage to the analog input pin AN₀ with an internal trigger which is generated by setting the A-D conversion start flag to "1" (refer to page 158 for information on A-D converter).

```

SEQ. LOC.  OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
1115      1 ;=====
1116      1 ;      「A-D conversion (one-shot mode)」 =
1117      1 ;=====
1118      1 ;      A-D conversion mode      : One-shot mode
1119      1 ;      Analog input pin        : AN0
1120      1 ;      Trigger                  : Internal trigger
1121      1 ;      A-D conversion frequency: f(XIN)/4
1122      1 ;
1123  00E000  1 SUB_AD1:
1124      1      .DATA 8
1125  00E000  F8      1      SEM                                ; Data length 8 bits
1126  00E001  *141101 L1     CLB #00000001B,P7D      ; Set P70/ANO pin to input mode
1127  00E004  *841E80 L1     LDM #1000000B,ADCON      ; One-shot mode, internal trigger,
1128      1 ;                                          ; A-D conversion frequency=f(XIN)/4
1129  00E007  *14700F L1     CLB #00001111B,ADIC      ; Clear A-D interrupt request bit, disable A-D interrupt
1130  00E00A  *041E40 L1     SEB #01000000B,ADCON      ; Start A-D conversion
1131      1 ;
1132      1 ;      ※ Interrupt request bit is set to "1" when A-D conversion of AN0 pin completes.
1133      1 ;
1134  00E00D  1 L_AD1:
1135  00E00D  *347008FC L1     BBC #00001000B,ADIC,L_AD1 ; A-D conversion complete (interrupt request bit= "1")?
1136  00E011  *147008 L1     CLB #00001000B,ADIC      ; Clear A-D interrupt request bit
1137  00E014  *A520 L1     LDA A,ADO                                ; Read conversion result (A-D register 0)
1138  00E016  60      1      RTS

```

(2) Repeat mode

This is a setting example for a program that repeatedly performs A-D conversion of input voltage to the analog input pin AN₁ with an internal trigger which is generated by setting the A-D conversion start flag to "1" (refer to page 158 for information on A-D converter).

```

SEQ. LOC.  OBJ.          ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
1141          1 ;=====
1142          1 ;      「A-D conversion (repeat mode)」      =
1143          1 ;=====
1144          1 ;      A-D conversion mode      : Repeat mode
1145          1 ;      Analog input pin      : AN1
1146          1 ;      Trigger                : Internal trigger
1147          1 ;      A-D conversion frequency: f(XIN)/4
1148          1 ;
1149 00E017    1 SUB_AD2:
1150          1      .DATA 8
1151 00E017 F8 1      SEM                      ; Data length 8 bits
1152 00E018 *141102 LI CLB #0000010B,P7D      ; Set P71/AN1 pin to input mode
1153 00E01B *841E89 LI LDM #10001001B,ADCON      ; Repeat mode, internal trigger,
1154          1 ;                                ; A-D conversion frequency=f(XIN)/4
1155 00E01E *041E40 LI SEB #0100000B,ADCON      ; Start A-D conversion
1156          1 ;
1157          1 ;      ※ After the time for A-D conversion (28.5 μs at 8MHz), the latest
1158          1 ;      conversion result can be obtained by reading A-D register 1 at any timing.
1159          1 ;
1160 00E021 60 1      RTS

```

(3) Single sweep mode

This is a setting example for a program that sequentially performs A-D conversion of input voltage to the analog input pins AN₀–AN₅, starting from AN₀, with an external trigger which is generated by changing the input level to AD_{TRG} pin from “H” to “L” with the A-D conversion start flag set to “1”. A-D conversion is stopped when all selected pins (AN₀–AN₅) are converted (refer to page 158 for information on A-D converter).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...
1183      1 ;=====
1184      1 ;      「A-D conversion (single sweep mode)」 =
1185      1 ;=====
1186      1 ;      A-D conversion mode      : Single sweep mode
1187      1 ;      Analog input pin       : AN0-AN5
1188      1 ;      Trigger                : External trigger
1189      1 ;      A-D conversion frequency: f(XIN)/4
1190      1 ;
1191      1171 00E022      1 SUB_AD3:
1192      1172      1      .DATA      8
1193      1173 00E022  F8      1      SEM                                ; Data length 8 bits
1194      1174 00E023 *1411BF  L1      CLB      #1011111B,P7D      ; Set AN0-AN5, ADTRG pins to input mode
1195      1175 00E028 *841F02  L1      LDM      #0000010B,ADSPS      ; Select AN0-AN5 pin for sweep
1196      1176 00E028 *841E0B  L1      LDM      #1011000B,ADCON      ; Single sweep mode, external trigger,
1197      1177      1 ;                                ; A-D conversion frequency=f(XIN)/4
1198      1178 00E02C *14700F  L1      CLB      #0000111B,ADIC      ; Clear A-D interrupt request bit, disable A-D interrupt
1199      1179 00E02F *041E40  L1      SEB      #0100000B,ADCON      ; Enable A-D conversion
1200      1180      1 ;
1201      1181      1 ;      ※ A-D conversion starts at falling edge input to ADTRG pin.
1202      1182      1 ;
1203      1183 00E032      1 L_AD3:
1204      1184 00E032 *347008FC L1      BBC      #00001000B,ADIC,L_AD3 ; Sweep complete (interrupt request bit= "1")?
1205      1185 00E036 *147008  L1      CLB      #00001000B,ADIC      ; Clear A-D interrupt request bit
1206      1186      1 ;
1207      1187      1 ;      ※ Conversion result can be obtained by reading A-D registers 0-5.
1208      1188      1 ;      Then A-D conversion is resumed when falling edge is input to ADTRG pin.
1209      1189      1 ;      Clear A-D conversion start flag to "0" if reconversion is not
1210      1190      1 ;      necessary.
1211      1191      1 ;
1212      1192 00E039  80      1      RTS

```


(4) Repeat sweep mode

This is a setting example for a program that sequentially performs A-D conversion of input voltage to the analog input pins AN₀–AN₇, starting from AN₀, with an internal trigger which is generated by setting the A-D conversion start flag to "1" (refer to page 158 for information on A-D converter).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
1195      1 ;=====
1196      1 ;      「A-D conversion (repeat sweep mode)」 =
1197      1 ;=====
1198      1 ;      A-D conversion mode      : Repeat sweep mode
1199      1 ;      Analog input pin        : AN0-AN7
1200      1 ;      Trigger                : Internal trigger
1201      1 ;      A-D conversion frequency: f(XIN)/4
1202      1 ;
1203 00E03A  1 SUB_AD4:
1204      1      .DATA      8
1205 00E03A  F8      1      SEM                      ; Data length 8 bits
1206 00E03B *841100  L1      LDM      #00H,P7D                ; Set AN0-AN7 pins to input mode
1207 00E03E *841F03  L1      LDM      #00000011B,ADSPS           ; Select AN0-AN7 pin for sweep
1208 00E041 *841E98  L1      LDM      #10011000B,ADCON           ; Repeat sweep mode, internal trigger,
1209      1 ;                      ; A-D conversion frequency=f(XIN)/4
1210 00E044 *041E40  L1      SEB      #01000000B,ADCON           ; Start A-D conversion
1211      1 ;
1212      1 ;      ※ After the first A-D sweep (228 μs at 3MHz), the latest
1213      1 ;      conversion result can be obtained by reading A-D registers 0-7
1214      1 ;      at any timing.
1215      1 ;
1216 00E047  80      1      RTS

```

9.7.7 Interrupt processing examples

(1) Interrupt setting example

This is a setting example for a program that generates an $\overline{\text{INT}}_1$ interrupt request when a falling edge is input to $\overline{\text{INT}}_1$ pin (refer to page 40 for information on interrupts).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....8....*....7....*....8....*....9....*...
1229          1 ;=====
1230          1 ;      Interrupt setting example      =
1231          1 ;=====
1232          1 ;
1233          1 ;      In order to execute an interrupt, the I flag
1234          1 ;      must be cleared within the main routine and
1235          1 ;      interrupt priority level must be set to level 1
1236          1 ;      or greater in each interrupt control register.
1237          1 ;
1238 00E800 78      1      SEI                      ; The I flag is initialized to "1" after reset.
1239          1 ;
1240          1 ;
1241          1      .DATA      8
1242 00E801 F8      1      SEM
1243 00E802 *141008  L1      CLB      #00001000B,P8D      ; Set P83/INT1 pin to input mode
1244 00E805 *147E0F  L1      CLB      #00111111B,INT1IC    ; Clear INT1 interrupt request bit, falling edge is effective
1245 00E808 *047E02  L1      SEB      #00000010B,INT1IC    ; Set INT1 interrupt priority level to level 2
1246 00E80B 58      1      CLI                      ; Enable interrupt
1247          1 ;
1248          1 ;      * Hereafter, INT1 interrupt request bit is set to "1" with falling
1249          1 ;      edge of input signal to INT1 pin.

```

APPLICATION

9.7 Program examples

(2) Interrupt processing example (when memory space is 64K bytes or less.)

This is an example of a program that must be executed within an interrupt service routine. The contents of each register is stored to stack in 16-bit length before starting interrupt processing. When returning from the interrupt processing to the main routine, the stored registers are restored in 16-bit length before executing the RTI instruction (refer to page 40 for information on interrupts).

```
SEQ. LOC. OBJ. ....*...1....*...2....*...SOURCE STATEMENT...5....*...6....*...7....*...8....*...9....*...
1253 1 ;=====
1254 1 ; Interrupt routine processing example 1 =
1255 1 ; When memory space is 64K bytes or less =
1256 1 ; (When data bank need not be changed) =
1257 1 ;=====
1258 1 ;
1259 00E80C 1 INTERRUPT_1:
1260 1 ;-----
1261 1 ; Store registers =
1262 1 ;-----
1263 1 ;
1264 1 .DATA 16 ; Register model declaration
1265 1 .INDEX 16
1266 00E80C C230 1 CLP m,x ; Data length, index register length 16 bits
1267 00E80E EB0D 1 PSH X,Y,A ; Store registers
1268 1 ;
1269 1 .DATA 8 ; Change register model
1270 1 .INDEX 8
1271 00E810 E230 1 SEP m,x ; Data length, index register length 8 bits
1272 1 ;
1273 1 ; : ; :
1274 1 ; : ; Interrupt processing
1275 1 ; : ; :
1276 1 ;
1277 1 ;-----
1278 1 ; Restore registers =
1279 1 ;-----
1280 1 ;
1281 1 .DATA 16 ; Change register model
1282 1 .INDEX 16
1283 00E812 C230 1 CLP m,x ; Data length, index register length 16 bits
1284 00E814 FB0D 1 PUL X,Y,A ; Restore registers X, Y, A
1285 1 ;
1286 1 ;-----
1287 1 ; Return from interrupt processing =
1288 1 ;-----
1289 1 ;
1290 00E816 40 1 RTI ; Return to processing before interrupt
1291 1 ;
1292 1 ; ※ The PS, PG, and PC are restored automatically to their values
1293 1 ; before the interrupt with the RTI instruction.
```

(3) Interrupt processing example (when memory space exceeds 64K bytes.)

This is an example of a program that must be executed within an interrupt service routine. When the memory space exceeds 64K bytes, data bank register must also be stored, set, and restored in addition to the processing for "(2) when memory space is 64K bytes or less" (refer to page 40 for information on interrupts).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*...

1296      1 ;=====
1297      1 ;   Interrupt routine processing example 2 =
1298      1 ;   When memory space exceeds 64K bytes   =
1299      1 ;   (When data bank is changed)           =
1300      1 ;=====
1301      1 ;
1302 00E817  1 INTERRUPT_2:                               ;
1303      1 ;-----
1304      1 ;   Store registers                          =
1305      1 ;-----
1306      1 ;
1307      1   .DATA      16           ; Register model declaration
1308      1   .INDEX    16           ;
1309 00E817  C230  1   CLP      m,x           ; Data length, index register length 16 bits
1310 00E819  EB0D  1   PSH     X,V,A        ; Store registers
1311 00E81B  8B   1   PHT     ; Store data bank register
1312      1   .DATA      8           ; Change register model
1313      1   .INDEX    8           ;
1314 00E81C  E230  1   SEP     m,x           ; Data length, index register length 8 bits
1315      1   .DT       012H        ; Declare data banks used for interrupt processing
1316 00E81E  89C212 1   LDT     #012H        ; Set data bank register used for interrupt processing
1317      1 ;
1318      1 ;           :           ;           :
1319      1 ;           :           ; Interrupt processing
1320      1 ;           :           ;           :
1321      1 ;
1322      1 ;-----
1323      1 ;   Restore registers                          =
1324      1 ;-----
1325      1 ;
1326 00E821  AB   1   PLT     ; Restore data bank register
1327      1   .DATA      16           ; Change register model
1328      1   .INDEX    16           ;
1329 00E822  C230  1   CLP     m,x           ; Data length, index register length 16 bits
1330 00E824  FB0D  1   PUL     X,V,A        ; Restore registers X, V, A
1331      1 ;
1332      1 ;-----
1333      1 ;   Return from interrupt processing =
1334      1 ;-----
1335      1 ;
1336 00E828  40   1   RTI     ; Return to processing before interrupt
1337      1 ;
1338      1 ;   ※ The PS, PG, PC are restored automatically to their values
1339      1 ;   before the interrupt with the RTI instruction.

```

9.7.8 Watchdog timer setting examples

(1) Watchdog timer write routine

This is a setting example for a program that sets "FFF₁₆" in watchdog timer when there is a write operation to the watchdog timer. This is used to detect program runaway (refer to page 172 for information on watchdog timer).

```
SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
1354      1 ;=====
1355      1 ;      [Watchdog timer write routine]      =
1358      1 ;=====
1357 00F000      1 WDT_SET:
1358 00F000 08      1      PHP                      ; Store PS
1359 00F001 F8      1      SEM                      ; Data length 8 bits (Note)
1360 00F002 *8560  LI      STA      A,WDT          ; Write to watchdog timer
1361 00F004 28      1      PLP                      ; Restore PS
1362 00F005 60      1      RTS
1363      1 ;
1364      1 ; Note: The address following the watchdog timer
1365      1 ; contains the watchdog timer frequency
1366      1 ; selection flag. Therefore, be careful not
1367      1 ; to change the value when accessing the
1368      1 ; watchdog timer in 18-bit unit.
```

(2) Watchdog timer interrupt processing example

This is a setting example for a program that performs a software reset when a watchdog timer interrupt occurs (refer to page 172 for information on watchdog timer).

```
SEQ. LOC.  OBJ.      ....*...1....*...2....*...SOURCE STATEMENT...5....*...6....*...7....*...8....*...9....*...
1371                1 ;=====
1372                1 ; Watchdog timer interrupt processing example =
1373                1 ;=====
1374 00F006          1 WATCH_DOG:
1375                1      .DATA 8
1376 00F006  F8      1      SEM          ;
1377 00F007 *045E08  L1      SEB    #00001000B,PMR      ; Software reset
1378 00F00A  40      1      RTI
```

APPLICATION

9.7 Program examples

9.7.9 Software timer setting examples

This is a setting example of a program that generates 10 μ s and 50 μ s waits without using a timer. These waits depend on the external clock input frequency and the address/data bus usage. This example assumes 8MHz external clock input frequency, 16-bit external bus width, and no wait.

```
SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....9....*....
1398      1 ;=====
1399      1 ;      10  $\mu$ s wait routine =
1400      1 ;=====
1401      1 ;      f(XIN)=8MHz    1  $\phi$ cyc=250ns
1402      1 ;
1403 OOF800  1 WIT10:
1404      1      .DATA  8
1405 OOF800  F8      1      SEM                ;          2  $\phi$ cyc
1406 OOF801  894920  1      RLA    #32                ;          8+32  $\phi$ cyc
1407      1 ;-----
1408      1 ;          Total 40  $\phi$ cyc
1409      1 ;
1410      1 ;=====
1411      1 ;      50  $\mu$ s wait routine =
1412      1 ;=====
1413      1 ;      f(XIN)=8MHz    1  $\phi$ cyc=250ns
1414      1 ;
1415 OOF804  1 WIT50:
1416      1      .DATA  8
1417 OOF804  F8      1      SEM                ;          2  $\phi$ cyc
1418 OOF805  8949C0  1      RLA    #192               ;          8+192  $\phi$ cyc
1419      1 ;-----
1420      1 ;          Total 200  $\phi$ cyc
```

9.7.10 Interrupt vector table setting example

Interrupt vector setting examples are shown below. Set the branch address of appropriate interrupt at the interrupt vector address (refer to page 40 for information on interrupts).

```

SEQ. LOC.  OBJ.      ....*....1....*....2....*....SOURCE STATEMENT....5....*....6....*....7....*....8....*....8....*....
1454      ;
1455      ;          Interrupt vector table          =
1456      ;
1457      ;
1458      .SECTION          VECTOR_AREA
1459      .ORG      OFFD8H
1480 OOFFDB 0000      L      .WORD      INT_AD          ; A-D conversion interrupt vector
1481 OOFFDB 0000      L      .WORD      INT_SIT        ; UART1 transmission interrupt vector
1482 OOFFDA 0000      L      .WORD      INT_SIR        ; UART1 receive interrupt vector
1483 OOFFDC 0000      L      .WORD      INT_SOT        ; UART0 transmission interrupt vector
1484 OOFFDE 0000      L      .WORD      INT_SOR        ; UART0 receive interrupt vector
1485 OOFFE0 0000      L      .WORD      INT_TB2        ; Timer B2 interrupt vector
1486 OOFFE2 0000      L      .WORD      INT_TB1        ; Timer B1 interrupt vector
1487 OOFFE4 0000      L      .WORD      INT_TB0        ; Timer B0 interrupt vector
1488 OOFFE8 0000      L      .WORD      INT_TA4        ; Timer A4 interrupt vector
1489 OOFFEB 0000      L      .WORD      INT_TA3        ; Timer A3 interrupt vector
1470 OOFFEA 0000      L      .WORD      INT_TA2        ; Timer A2 interrupt vector
1471 OOFFEC 0000      L      .WORD      INT_TA1        ; Timer A1 interrupt vector
1472 OOFFEE 0000      L      .WORD      INT_TAO        ; Timer A0 interrupt vector
1473 OOFFFO 0000      L      .WORD      INT2          ; INT2 interrupt vector
1474 OOFFF2 0000      L      .WORD      INT1          ; INT1 interrupt vector
1475 OOFFF4 0000      L      .WORD      INTO          ; INTO interrupt vector
1478 OOFFFB 0000      L      .WORD      INT_WDT        ; Watchdog timer interrupt vector
1477 OOFFF8 0000      L      .WORD      RESERVED        ; (Reserved area)
1478 OOFFFA 0000      L      .WORD      INT_BRK        ; BRK instruction interrupt vector
1479 OOFFFC 0000      L      .WORD      INT_DIV0        ; Zero divide interrupt vector
1480 OOFFFE C000      L      .WORD      INITIAL        ; Reset vector
1481      ;
1482      .END

```


APPENDIX

- Appendix 1. SFR area memory map
- Appendix 2. Control registers
- Appendix 3. Stop, wait, one-wait, Ready, Hold state
- Appendix 4. Package outlines
- Appendix 5. Setting example of unused pins
- Appendix 6. Print foot pattern examples
- Appendix 7. System development precautions
- Appendix 8. M66800SP/FP
- Appendix 9. Machine instructions
- Appendix 10. Instruction code table

APPENDIX

Appendix 1. SFR area memory map

Appendix 1. SFR area memory map

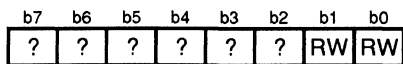
Address
(Hexadecimal notation)

Address (Hexadecimal notation)	Registers	Access
000000		
000001		
000002		
000003		
000004		
000005		
000006		
000007		
000008		
000009		
00000A	Port P4 register	RW
00000B	Port P5 register	RW
00000C	Port P4 direction register	RW
00000D	Port P5 direction register	RW
00000E	Port P6 register	RW
00000F	Port P7 register	RW
000010	Port P6 direction register	RW
000011	Port P7 direction register	RW
000012	Port P8 register	RW
000013		
000014	Port P8 direction register	RW
000015		
00001D		
00001E	A-D control register	RW
00001F	A-D sweep pin selection register	→
000020	A-D register 0	RO
000021		
000022	A-D register 1	RO
000023		
000024	A-D register 2	RO
000025		

RO ... Read only
 WO ... Write only
 RW ... Read/Write
 ? Impossible to write,
 and undefine at reading
 0 Impossible to write,
 and fixed to "0" at reading

} M37730 group dose not have this register.

} M37730 group dose not have this register.



} M37730 group dose not have these registers.

APPENDIX

Appendix 1. SFR area memory map

Address (Hexadecimal notation) Registers Access

000026	A-D register 3	RO
000027		
000028	A-D register 4	RO
000029		
00002A	A-D register 5	RO
00002B		
00002C	A-D register 6	RO
00002D		
00002E	A-D register 7	RO
00002F		
000030	UART0 transmit/receive mode register	RW
000031	UART0 baud rate register	WO
000032	UART0 transmission buffer register	L: WO H: WO
000033		
000034	UART0 transmit/receive control register 0	→
000035	UART0 transmit/receive control register 1	→
000036	UART0 receive buffer register	L: RO H: →
000037		
000038	UART1 transmit/receive mode register	RW
000039	UART1 baud rate register	WO
00003A	UART1 transmission buffer register	L: WO H: WO
00003B		
00003C	UART1 transmit/receive control register 0	→
00003D	UART1 transmit/receive control register 1	→
00003E	UART1 receive buffer register	L: RO H: →
00003F		
000040	Count start flag	RW ※1
000041		
000042	One-shot start flag	→
000043		
000044	Up-down flag	→
000045		
000046	Timer A0 register	L: RW H: RW
000047		
000048	Timer A1 register	L: RW H: RW
000049		

M37730 group dose not have these registers.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	RO

M37730 group dose not have these registers.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	RO	RW	RW	RW
RO	RO	RO	RO	RO	RW	RO	RW

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	RO

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	WO	WO	WO	WO	WO

b7	b6	b5	b4	b3	b2	b1	b0
WO	WO	WO	RW	RW	RW	RW	RW

※1: The contents of M37730 group's count start flag are described on the rihgt.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	RW	RW	RW	RW	RW	RW

APPENDIX

Appendix 1. SFR area memory map

Address (Hexadecimal notation)	Registers	Access
00004A	Timer A2 register	L
00004B		H
00004C	Timer A3 register	L
00004D		H
00004E	Timer A4 register	L
00004F		H
000050	Timer B0 register	L
000051		H
000052	Timer B1 register	L
000053		H
000054	Timer B2 register	L
000055		H
000056	Timer A0 mode register	RW
000057	Timer A1 mode register	RW
000058	Timer A2 mode register	RW
000059	Timer A3 mode register	RW
00005A	Timer A4 mode register	RW
00005B	Timer B0 mode register	→
00005C	Timer B1 mode register	→
00005D	Timer B2 mode register	→
00005E	Processor mode register	→
00005F		
000060	Watchdog timer	WO
000061	Watchdog timer frequency selection flag	→
000062	Waveform output mode register	→
000063		
000064	Pulse output data register 1	→
000065	Pulse output data register 0	→
000066		

M37730 group dose not have these registers.

b7	b6	b5	b4	b3	b2	b1	b0
RW	RW	RO	?	RW	RW	RW	RW
RW	RW	RO	?	RW	RW	RW	RW
RW	RW	RO	?	RW	RW	RW	RW
?	RW	RW	RW	WO	RW	RW	RW

M37730 group dose not have these registers.

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	?	?	?	RW
?	?	RW	RW	RW	?	RW	RW

※2

b7	b6	b5	b4	b3	b2	b1	b0
?	?	?	?	WO	WO	WO	WO
?	?	?	?	WO	WO	WO	WO

※2: The contents of M37730 group's waveform output mode register are described on the rihgt.

b7	b6	b5	b4	b3	b2	b1	b0
RW	?	RW	RW	RW	?	RW	RW

APPENDIX

Appendix 1. SFR area memory map

Address (Hexadecimal notation)	Registers	Access									
000070	A-D conversion interrupt control register	→	b7	b6	b5	b4	b3	b2	b1	b0	} M37730 group dose not have this register.
000071	UART0 transmission interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
000072	UART0 receive interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
000073	UART1 transmission interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
000074	UART1 receive interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
000075	Timer A0 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
000076	Timer A1 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
000077	Timer A2 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
000078	Timer A3 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
000079	Timer A4 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
00007A	Timer B0 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
00007B	Timer B1 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	
00007C	Timer B2 interrupt control register	→	?	?	?	?	RW	RW	RW	RW	} M37730 group dose not have these registers.
00007D	$\overline{\text{INT}}_0$ interrupt control register	→	?	?	RW	RW	RW	RW	RW	RW	
00007E	$\overline{\text{INT}}_1$ interrupt control register	→	?	?	RW	RW	RW	RW	RW	RW	} M37730 group dose not have these registers.
00007F	$\overline{\text{INT}}_2$ interrupt control register	→	?	?	RW	RW	RW	RW	RW	RW	
000080	Internal RAM										

Appendix 2. Control registers

The register structure of each control register allocated in the SFR area of M37732 group are shown on the following pages. Each table shows the bit names, functions, contents when reset is removed, and bit attributes.

* Bit attributes: Each bit in the control register is either read only, write only, or read/write. The following abbreviations are used to indicate the attribute.

R : Read

W : Write

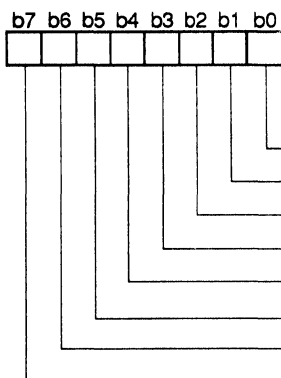
○ : Possible to read or write

× : Impossible to read or write

[M37732 Group]

1. Port PI direction registers (I=4 to 8)

Port PI direction register
(Addresses 0C₁₆, 0D₁₆, 10₁₆, 11₁₆, 14₁₆)

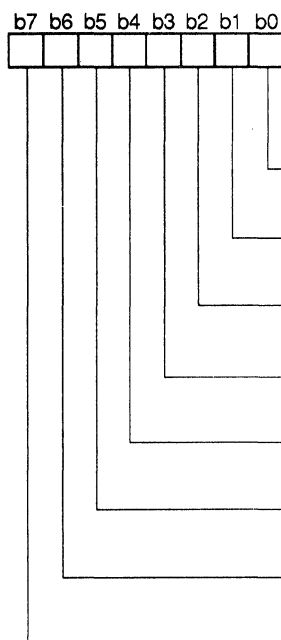


Bit	Bit name	Functions	At reset	R	W
0	Port P ₁₀ direction selection bit	0 : Input mode 1 : Output mode	0	<input type="radio"/>	<input type="radio"/>
1	Port P ₁₁ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
2	Port P ₁₂ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
3	Port P ₁₃ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
4	Port P ₁₄ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
5	Port P ₁₅ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
6	Port P ₁₆ direction selection bit		0	<input type="radio"/>	<input type="radio"/>
7	Port P ₁₇ direction selection bit		0	<input type="radio"/>	<input type="radio"/>

Note: There are no pins corresponding to the low-order 3 bits of port P4 direction register.

2.A-D control register

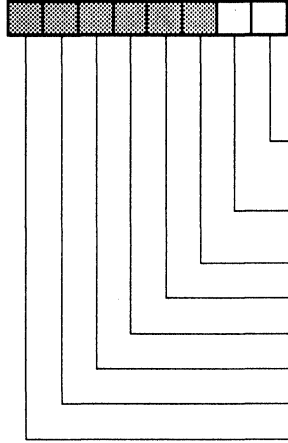
A-D control register (Address 1E₁₆)



Bit	Bit name	Functions	At reset	R	W
0	Analog input selection bits	b2b1b0 0 0 0 : AN ₀ pin selected 0 0 1 : AN ₁ pin selected 0 1 0 : AN ₂ pin selected 0 1 1 : AN ₃ pin selected 1 0 0 : AN ₄ pin selected 1 0 1 : AN ₅ pin selected 1 1 0 : AN ₆ pin selected 1 1 1 : AN ₇ pin selected (Note)	Undefined	<input type="radio"/>	<input type="radio"/>
1			Undefined	<input type="radio"/>	<input type="radio"/>
2			Undefined	<input type="radio"/>	<input type="radio"/>
3			A-D conversion mode selection bits	b4b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode	0
4	0	<input type="radio"/>			<input type="radio"/>
5	Trigger selection bit	0 : Software trigger (internal trigger) 1 : AD _{TRG} input trigger (external trigger)	0	<input type="radio"/>	<input type="radio"/>
6	A-D conversion start flag	0 : Stop A-D conversion 1 : Start A-D conversion	0	<input type="radio"/>	<input type="radio"/>
7	A-D conversion frequency (φ _{AD}) selection flag	0 : f(X _{IN})/8 (f ₈) 1 : f(X _{IN})/4 (f ₄)	0	<input type="radio"/>	<input type="radio"/>

Note: AN₇ pin cannot be used as analog voltage input pin when an external trigger is selected.

3.A-D sweep pin selection register

b7 b6 b5 b4 b3 b2 b1 b0

 A-D sweep pin selection register (Address 1F₁₆)

Bit	Bit name	Functions	At reset	R	W
0	A-D sweep pin selection bits	b1b0 0 0 : AN ₀ , AN ₁ (2 pins) 0 1 : AN ₀ to AN ₃ (4 pins) 1 0 : AN ₀ to AN ₅ (6 pins) 1 1 : AN ₀ to AN ₇ (8 pins)	1	<input type="radio"/>	<input type="radio"/>
1			1	<input type="radio"/>	<input type="radio"/>
2	These bits cannot be written and are undefined at reading.		Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3			Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4			Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5			Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6			Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7			Undefined	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

4.UART_I transmit/receive mode registers (I=0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

 UART₀ transmit/receive mode register (Address 30₁₆)
 UART₁ transmit/receive mode register (Address 38₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Serial I/O mode selection bits	b2b1b0 0 0 0 : Serial I/O prohibited 0 0 1 : Clock synchronous serial I/O 0 1 0 : This is not available. 0 1 1 : This is not available. 1 0 0 : 7-bit UART 1 0 1 : 8-bit UART 1 1 0 : 9-bit UART 1 1 1 : This is not available.	0	<input type="radio"/>	<input type="radio"/>
1			0	<input type="radio"/>	<input type="radio"/>
2			0	<input type="radio"/>	<input type="radio"/>
3	Internal/External clock selection bit	0 : Internal clock 1 : External clock	0	<input type="radio"/>	<input type="radio"/>
4	Stop bit length selection bit (in UART mode)	0 : One stop bit 1 : Two stop bits	0	<input type="radio"/>	<input type="radio"/>
5	Odd/Even parity selection bit (in UART mode)	0 : Odd parity 1 : Even parity	0	<input type="radio"/>	<input type="radio"/>
6	Parity enable bit (in UART mode)	0 : Parity disabled 1 : Parity enabled	0	<input type="radio"/>	<input type="radio"/>
7	Sleep function selection bit (in UART mode)	0 : Sleep function disabled 1 : Sleep function enabled	0	<input type="radio"/>	<input type="radio"/>

Note: Bits 4 to 6 are ignored in clock synchronous mode.
 Bit 7 must be fixed to "0" when using clock synchronous mode.

5. UARTi baud rate register

Bit	Bit name	Functions	At reset	R	W
0	When an internal clock is selected in clock synchronous serial I/O mode, set a value between 00 ₁₆ and FF ₁₆ to set the BRG count source dividing ratio.		Undefined	X	○
1			Undefined	X	○
2	In UART mode, set a value between 00 ₁₆ and FF ₁₆ regardless of an internal clock or external clock to set the BRG count source dividing ratio.		Undefined	X	○
3			Undefined	X	○
4			Undefined	X	○
5			Undefined	X	○
6	The BRG divides an input clock by (n+1). n: a value set in the UARTi baud rate register		Undefined	X	○
7			Undefined	X	○

6. UARTi transmit/receive control register 0 (i=0, 1)

Bit	Bit name	Functions	At reset	R	W
0	BRG count source selection bits	b1b0 0 0 : f(X _{IN})/2 (f ₂) selected 0 1 : f(X _{IN})/16 (f ₁₆) selected 1 0 : f(X _{IN})/64 (f ₆₄) selected 1 1 : f(X _{IN})/512 (f ₅₁₂) selected	0	○	○
1			0	○	○
2	CTS/RTS function selection bit	0 : CTS function selected 1 : RTS function selected	0	○	○
3	Transmission register empty flag	0 : Data in transmission register (transmitting) 1 : No data in transmission register (transmit complete)	1	○	X
4	These bits cannot be written and are undefined at reading.		Undefined	X	X
5			Undefined	X	X
6			Undefined	X	X
7			Undefined	X	X

7. UARTI transmit/receive control register 1 (I=0, 1)

b7 b6 b5 b4 b3 b2 b1 b0



UART0 transmit/receive control register 1 (Address 35₁₆)
 UART1 transmit/receive control register 1 (Address 3D₁₆)

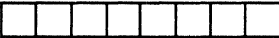
Bit	Bit name	Functions	At reset	R	W
0	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0	○	○
1	Transmission buffer empty flag	0 : Data in transmission buffer register 1 : No data in transmission buffer register	1	○	×
2	Receive enable bit	0 : Receive disabled 1 : Receive enabled	0	○	○
3	Receive completion flag	0 : No data in receive buffer register 1 : Data in receive buffer register	0	○	×
4	Overrun error flag	0 : No overrun error 1 : Overrun error	0	○	×
5	Framing error flag (in UART mode)	0 : No framing error 1 : Framing error	0	○	×
6	Parity error flag (in UART mode)	0 : No parity error 1 : Parity error	0	○	×
7	Error sum flag (in UART mode)	0 : No error 1 : Error	0	○	×

Note: Bits 5 to 7 are ignored in clock synchronous mode.

Receive completion flag and each error flag are cleared to "0" when the low-order byte of receive buffer register is read or the receive enable bit is cleared to "0".

8. Count start flag

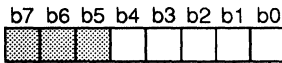
b7 b6 b5 b4 b3 b2 b1 b0



Count start flag (Address 40₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 count start flag	0 : Count stop 1 : Count start	0	○	○
1	Timer A1 count start flag		0	○	○
2	Timer A2 count start flag		0	○	○
3	Timer A3 count start flag		0	○	○
4	Timer A4 count start flag		0	○	○
5	Timer B0 count start flag		0	○	○
6	Timer B1 count start flag		0	○	○
7	Timer B2 count start flag		0	○	○

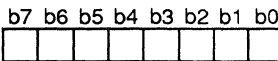
9. One-shot start flag



One-shot start flag (Address 42₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 one-shot start flag	1 : One-shot start	0	X	O
1	Timer A1 one-shot start flag		0	X	O
2	Timer A2 one-shot start flag		0	X	O
3	Timer A3 one-shot start flag		0	X	O
4	Timer A4 one-shot start flag		0	X	O
5	These bits cannot be written and are undefined at reading.		Undefined	X	X
6			Undefined	X	X
7			Undefined	X	X

10. Up-down flag



Up-down flag (Address 44₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Timer A0 up-down flag	0 : Decrement count 1 : Increment count	0	O	O
1	Timer A1 up-down flag		0	O	O
2	Timer A2 up-down flag		0	O	O
3	Timer A3 up-down flag		0	O	O
4	Timer A4 up-down flag		0	O	O
5	Timer A2 two-phase signal processing selection bit	0 : Two-phase pulse signal processing function disabled 1 : Two-phase pulse signal processing function enabled	0	X	O
6	Timer A3 two-phase signal processing selection bit		0	X	O
7	Timer A4 two-phase signal processing selection bit		0	X	O

Note: Data must be written using **LDM** and **STA** instructions for bits 5 to 7.

11. Timer Ai mode registers (I=0 to 4)

b7 b6 b5 b4 b3 b2 b1 b0
 Timer Ai mode register (Addresses 56₁₆ to 5A₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Operating mode selection bits	b1b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot pulse mode 1 1 : PWM mode	0	○	○
1			0	○	○
2	These bits' functions depend on operating mode.		0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})	0	○	○
7			0	○	○

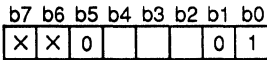
Note: In event counter mode, bits 6 and 7 are ignored.

(1) Timer mode

b7 b6 b5 b4 b3 b2 b1 b0
 Timer Ai mode register <Timer mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 0 : Timer mode
1		
2	Pulse output function selection bit	0 : No pulse output 1 : Pulse output from TAIOUT pin
3	Gate function selection bits	b4b3 0 X : No gate function 1 0 : Count while TAIIN input level is "L" 1 1 : Count while TAIIN input level is "H"
4		
5	This bit must be fixed to "0".	
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})
7		

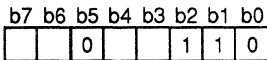
(2)Event counter mode



Timer Ai mode register <Event counter mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 1 : Event counter mode
1		
2	Pulse output function selection bit	0 : No pulse output 1 : Pulse output from TAIOUT pin
3	Count polarity selection bit	0 : Count at the falling edge of the input signal 1 : Count at the rising edge of the input signal
4	Up-down switching factor selection bit	0 : Contents of the up-down flag 1 : Input signal of the TAIOUT pin
5	This bit must be fixed to "0".	
6	These bits are ignored (may be "0" or "1").	
7		

(3)One-shot pulse mode



Timer Ai mode register <One-shot pulse mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 0 : One-shot pulse mode
1		
2	This bit must be fixed to "1".	
3	Trigger selection bits	b4b3 0 × : Writing operation to the one-shot start flag 1 0 : Falling edge of the input signal to TAIIN pin 1 1 : Rising edge of the input signal to TAIIN pin
4		
5	This bit must be fixed to "0".	
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})
7		

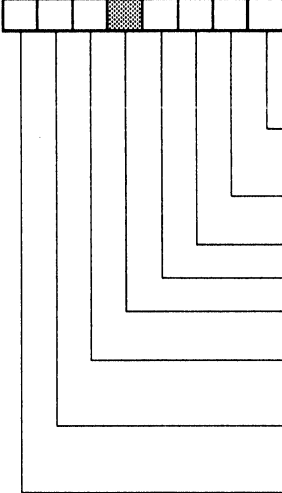
(4)PWM mode

b7 b6 b5 b4 b3 b2 b1 b0
 [] [] [] [] [] [1] [1] [1] Timer Ai mode register <PWM mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 1 : PWM mode
1		
2	This bit must be fixed to "1".	
3	Trigger selection bits	b4b3 0 X : Count start flag 1 0 : Falling edge of the input signal to TAin pin 1 1 : Rising edge of the input signal to TAin pin
4		
5	16/8-bit PWM mode selection bit	0 : 16-bit PWM mode 1 : 8-bit PWM mode
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})
7		

12.Timer BI mode registers (I=0 to 2)

b7 b6 b5 b4 b3 b2 b1 b0



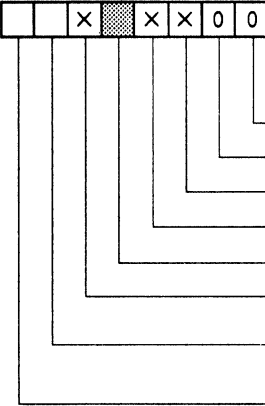
Timer BI mode register (Addresses 5B1_h to 5D1_h)

Bit	Bit name	Functions	At reset	R	W
0	Operating mode selection bits	b1b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/Pulse width measurement mode 1 1 : This is not available.	0	○	○
1			0	○	○
2	These bits' functions depend on operating mode.		0	○	○
3			0	○	○
4	This bit cannot be written and is undefined at reading.		Undefined	×	×
5	This bit's function depends on operating mode.		1	○	×
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})	0	○	○
7			0	○	○

Note: In timer mode and event counter mode, bit 5 is ignored and undefined at reading.

(1)Timer mode

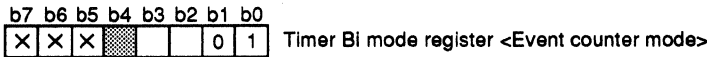
b7 b6 b5 b4 b3 b2 b1 b0



Timer BI mode register <Timer mode>

Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 0 : Timer mode
1		
2	These bits are ignored (may be "0" or "1").	
3		
4	This bit cannot be written and is undefined at reading.	
5	This bit is ignored and undefined at reading.	
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})
7		

(2)Event counter mode



Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 0 1 : Event counter mode
1		
2	Count polarity selection bits	b3b2 0 0 : Count at the falling edge of the input signal 0 1 : Count at the rising edge of the input signal 1 0 : Count at both edges of the input signal 1 1 : This is not available.
3		
4	This bit cannot be written and is undefined at reading.	
5	This bit is ignored and undefined at reading.	
6	These bits are ignored (may be "0" or "1").	
7		

(3)Pulse period/Pulse width measurement mode



Bit	Bit name	Functions
0	Operating mode selection bits	b1b0 1 0 : Pulse period/Pulse width measurement mode
1		
2	Measurement mode selection bits	b3b2 0 0 : Pulse period measurement mode (between the falling edge and the falling edge of the input signal) 0 1 : Pulse period measurement mode (between the rising edge and the rising edge of the input signal) 1 0 : Pulse width measurement mode 1 1 : This is not available.
3		
4	This bit cannot be written and is undefined at reading.	
5	Timer Bi overflow flag	0 : No overflow 1 : Overflow
6	Count source selection bits	b7b6 0 0 : $f(X_{IN})/2$ (f_2) 0 1 : $f(X_{IN})/16$ (f_{16}) 1 0 : $f(X_{IN})/64$ (f_{64}) 1 1 : $f(X_{IN})/512$ (f_{512})
7		

Note: The timer Bi overflow flag which is set to "1" is cleared by writing to this register with the count start flag is "1".

13.Processor mode register

b7 b6 b5 b4 b3 b2 b1 b0
 Processor mode register (Address 5E16)

Bit	Bit name	Functions	At reset	R	W
0	This bit must be fixed to "0".		0	○	○
1	This bit must be fixed to "1".		1	○	○
2	Wait bit	0 : Wait during external access 1 : No wait	0	○	○
3	Software reset bit	Software reset activated by writing "1".	0	✗	○
4	Interrupt priority detection time selection bits	b5b4 0 0 : 7 cycles at internal clock ϕ 0 1 : 4 cycles at internal clock ϕ 1 0 : 2 cycles at internal clock ϕ 1 1 : This is not available.	0	○	○
5			0	○	○
6	This bit must be fixed to "0".		0	○	○
7	This bit cannot be written and is undefined at reading.		Undefined	✗	✗

14.Watchdog timer frequency selection flag

b7 b6 b5 b4 b3 b2 b1 b0
 Watchdog timer frequency selection flag (Address 6116)

Bit	Bit name	Functions	At reset	R	W
0	Watchdog timer frequency selection flag	0 : $f(X_{IN})/512$ (f_{512}) 1 : $f(X_{IN})/32$ (f_{32})	0	○	○
1	These bits cannot be written and are undefined at reading.		Undefined	✗	✗
2			Undefined	✗	✗
3			Undefined	✗	✗
4			Undefined	✗	✗
5			Undefined	✗	✗
6			Undefined	✗	✗
7			Undefined	✗	✗

15. Waveform output mode register

b7 b6 b5 b4 b3 b2 b1 b0
 Waveform output mode register (Address 62₁₆)

Bit	Bit name	Functions	At reset	R	W
0	Waveform output mode selection bits	b1b0 0 0 : Programmable I/O port 0 1 : RTP1 selected 1 0 : RTP0 selected 1 1 : RTP0 and RTP1 selected	0	○	○
1			0	○	○
2	This bit cannot be written and is undefined at reading.		Undefined	×	×
3	RTP0 polarity selection bit	0 : Positive polarity 1 : Negative polarity	0	○	○
4	RTP0 pulse width modulation by timer A1 selection bit	0 : Not modulated 1 : Modulated	0	○	○
5	RTP1 pulse width modulation by timer A3 selection bit	0 : Not modulated 1 : Modulated	0	○	○
6	These bits cannot be written and are undefined at reading.		Undefined	×	×
7			Undefined	×	×

16. Pulse output data register I (I=0, 1)

b7 b6 b5 b4 b3 b2 b1 b0
 Pulse output data register 1 (Address 64₁₆)
 Pulse output data register 0 (Address 65₁₆)

Bit	Bit name	Functions	At reset	R	W
0	RTP ₀ output data	0 : "L" level output 1 : "H" level output	Undefined	×	○
1	RTP ₁ output data		Undefined	×	○
2	RTP ₂ output data		Undefined	×	○
3	RTP ₃ output data		Undefined	×	○
4	These bits cannot be written and are undefined at reading.		Undefined	×	×
5			Undefined	×	×
6			Undefined	×	×
7			Undefined	×	×

Note: Data must be written using **LDM** and **STA** instructions for bits 0 to 3.

APPENDIX

Appendix 2. Control registers

17. A-D conversion, UART 0 and 1 transmission, UART 0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers

A-D conversion, UART0 and 1 transmission, UART0 and 1 receive, timers A0 to A4, timers B0 to B2 interrupt control registers (Addresses 70₁₆ to 7C₁₆)

Bit	Bit name	Functions	At reset	R	W	
0	Interrupt priority level selection bits	b2b1b0	0	○	○	
1		0 0 0 : Level 0 (Interrupt disabled)				
		0 0 1 : Level 1				
		0 1 0 : Level 2				
		0 1 1 : Level 3				
		1 0 0 : Level 4				
		1 0 1 : Level 5				
2		1 1 0 : Level 6				
		1 1 1 : Level 7				
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	○	
4	These bits cannot be written and are undefined at reading.			Undefined	×	×
5				Undefined	×	×
6				Undefined	×	×
7				Undefined	×	×

Note: Use the **SEB** and **CLB** instructions when setting each interrupt control register.

18. $\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers

$\overline{INT_0}$ to $\overline{INT_2}$ interrupt control registers (Addresses 7D₁₆ to 7F₁₆)

Bit	Bit name	Functions	At reset	R	W	
0	Interrupt priority level selection bits	b2b1b0	0	○	○	
1		0 0 0 : Level 0 (Interrupt disabled)				
		0 0 1 : Level 1				
		0 1 0 : Level 2				
		0 1 1 : Level 3				
		1 0 0 : Level 4				
		1 0 1 : Level 5				
2		1 1 0 : Level 6				
		1 1 1 : Level 7				
3	Interrupt request bit	0 : No interrupt request 1 : Interrupt request	0	○	○	
4	Level/Edge selection bit	0 : Set interrupt request bit at "H" level for level sense and the falling edge for edge sense. 1 : Set interrupt request bit at "L" level for level sense and the rising edge for edge sense.	0	○	○	
5	Level sense/Edge sense selection bit	0 : Edge sense 1 : Level sense	0	○	○	
6	These bits cannot be written and are undefined at reading.			Undefined	×	×
7				Undefined	×	×

Note: Use the **SEB** and **CLB** instructions when setting each interrupt control register.

[M37730 Group]

For M37730 group, the following two registers are different from M37732 group.

1.Count start flag

b7	b6	b5	b4	b3	b2	b1	b0	Count start flag (Address 40 ₁₆)
Bit	Bit name	Functions	At reset	R	W			
0	Timer A0 count start flag	0 : Count stop 1 : Count start	0	○	○			
1	Timer A1 count start flag		0	○	○			
2	Timer A2 count start flag		0	○	○			
3	Timer A3 count start flag		0	○	○			
4	Timer A4 count start flag		0	○	○			
5	Timer B0 count start flag		0	○	○			
6	These bits cannot be written and are undefined at reading.		Undefined	×	×			
7			Undefined	×	×			

2.Waveform output mode register

b7	b6	b5	b4	b3	b2	b1	b0	Waveform output mode register (Address 62 ₁₆)
Bit	Bit name	Functions	At reset	R	W			
0	Waveform output mode selection bits	b1b0 0 0 : Programmable I/O port 0 1 : RTP1 selected 1 0 : RTP0 selected 1 1 : RTP0 and RTP1 selected	0	○	○			
1			0	○	○			
2	This bit cannot be written and is undefined at reading.		Undefined	×	×			
3	RTP0 polarity selection bit	0 : Positive polarity 1 : Negative polarity	0	○	○			
4	RTP0 pulse width modulation by timer A1 selection bit	0 : Not modulated 1 : Modulated	0	○	○			
5	RTP1 pulse width modulation by timer A3 selection bit	0 : Not modulated 1 : Modulated	0	○	○			
6	This bit cannot be written and is undefined at reading.		Undefined	×	×			
7	Waveform output control bit	0 : Waveform output disabled 1 : Waveform output enabled	0	○	○			

APPENDIX

Appendix 3. Stop, wait, one-wait, Ready, Hold state

Appendix 3. Stop, wait, one-wait, Ready, Hold state

1. Stop, wait, one-wait, Ready, Hold state

Table 1 shows the stop, wait, one-wait, Ready, and Hold state.

The following are some notes for items in Table 1.

(1) Oscillation

Timer A, timer B, serial I/O, and A-D converter can be used when the oscillator is operating.

(2) STP Instruction

Table 2 shows the external interrupts used to remove the state (stop mode) after executing the **STP** instruction.

If these external interrupts are used to remove the stop mode, enable the interrupts before entering the stop mode.

Table 2 External interrupts used to remove stop mode

External interrupt	Interrupt source
External input signal	INT ₀ , INT ₁ , INT ₂
Serial I/O using external clock (clock synchronous/asynchronous)	UART0 reception, UART0 transmission UART1 reception, UART1 transmission
Timer interrupts in event counter mode	Timer A0, Timer A1, Timer A2, Timer A3, Timer A4, Timer B0, Timer B1, Timer B2

Note: M37730 group does not have UART1, Timer B1 and Timer B2.

(3) STP, WIT instructions

The reset used to remove the state (stop mode) after executing the **STP** instruction or the state (wait mode) after executing the **WIT** instruction is a hardware reset. If a hardware reset is used to remove a stop mode or wait mode, the contents of the internal RAM are the contents before executing these instructions. The state of the other internal registers are the same as described in section "3.1.2 Internal registers state at reset". The contents of the internal RAM is not retained if a hardware reset is performed in cases other than stop mode or wait mode.

If **STP** or **WIT** instruction is to be executed after writing to internal RAM, SFR, external memory, or peripheral IC, insert **NOP** instructions in front of these instructions. Table 3 shows the number of **NOP** instructions to insert.

Table 3 Number of NOP instructions to insert

Condition	NOP instructions
After writing to internal RAM and SFR	1
After writing to external memory and peripheral IC when the wait bit (bit 2 at address 5E ₁₆) is "1"	1
After writing to external memory and peripheral IC when the wait bit is "0" (one-wait mode)	3

APPENDIX

Appendix 3. Stop, wait, one-wait, Ready, Hold state

Table 1 Stop, wait, one-wait, Ready, Hold state

Parameter	Source	STP instruction (stop mode)	WIT instruction (wait mode)	Wait bit (one-wait mode)	RDY input (Ready state)	HOLD input (Hold state)
Oscillator		Stopped	Operating	Operating	Operating	Operating
ϕ_1 output		Stop at "L" level	Operating	Operating	Operating	Operating
\bar{E} output		Stop at "H" or "L" level	Stop at "H" or "L" level	"L" pulse width becomes twice at external access.	Stop at "H" or "L" level	Stop at "H" level
Bus state		Retain bus and port state when STP instruction is executed	Retain bus and port state when WIT instruction is executed	—	Retain bus and port state when "L" level is applied	Bus, R/W pin, BHE pin are floating. ALE pin, HLDA pin stop at "L" level. Ports P43–P47, P5, P6, P7, P8 retain state when "L" level is applied.
Port state						
CPU		Stopped	Stopped	Operating	Stopped	Stopped
Timer		Stopped	Operating	Operating	Operating	Operating
Serial I/O						
A-D converter						
Watchdog timer state		Stopped (set "FFF16" in watchdog timer and select count source f32)	Operating	Operating	Operating	Stopped
Removing of state		Hardware reset or accepting external interrupt	Hardware reset or accepting interrupt	Set processor mode register bit 2 to "1".	Return RDY input to "H" level.	Return HOLD input to "H" level.

Note: M37730 group does not have port P7.

APPENDIX

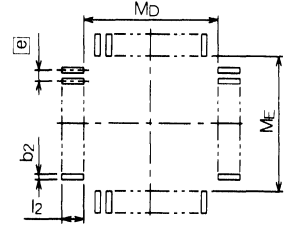
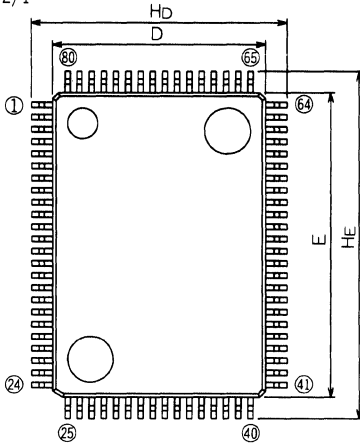
Appendix 4. Package outlines

Appendix 4. Package outlines 80P6N-A

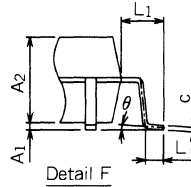
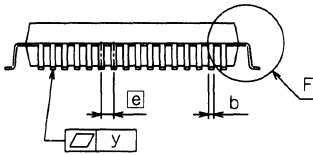
Plastic 80pin 14x20mm body QFP

EIAJ Package Code *QFP080-P-1420	JEDEC Code --	Weight(g) 1.58
-------------------------------------	------------------	-------------------

Scale : 2/1



Recommended Mount Pad



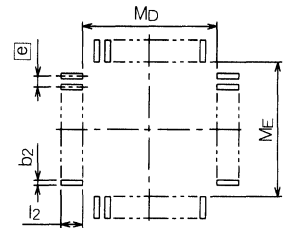
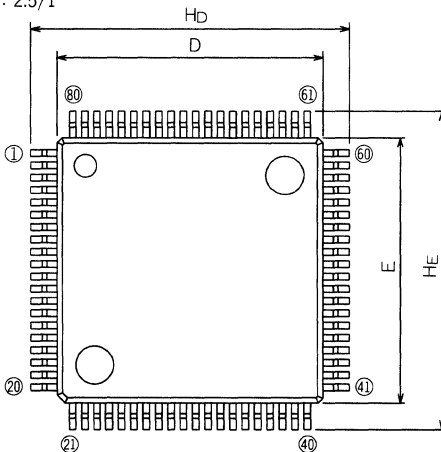
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A1	0	0.1	0.2
A2	—	2.8	—
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	—	0.8	—
H _D	16.5	16.8	17.1
H _E	22.5	22.8	23.1
L	0.4	0.6	0.8
L ₁	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b ₂	—	0.5	—
l ₂	—	1.3	—
M _D	—	14.6	—
M _E	—	20.6	—

80P6S-A

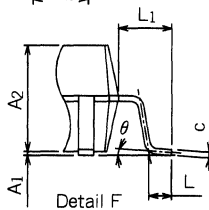
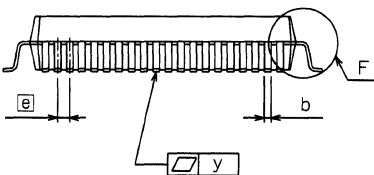
Plastic 80pin 14x14mm body QFP

EIAJ Package Code *QFP080-P-1414	JEDEC Code --	Weight(g) 1.11
-------------------------------------	------------------	-------------------

Scale : 2.5/1



Recommended Mount Pad



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A1	0	0.1	0.2
A2	—	2.8	—
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	—	0.65	—
H _D	16.5	16.8	17.1
H _E	16.5	16.8	17.1
L	0.4	0.6	0.8
L ₁	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b ₂	—	0.35	—
l ₂	—	1.3	—
M _D	—	14.6	—
M _E	—	14.6	—

APPENDIX

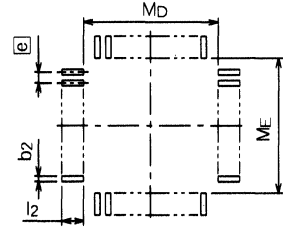
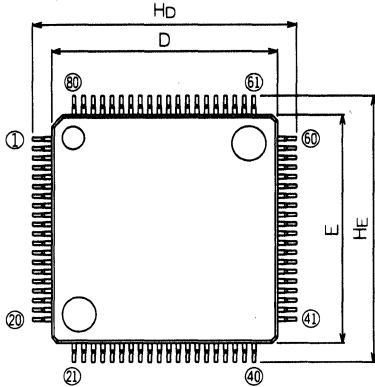
Appendix 4. Package outlines

80P6D-A

Plastic 80pin 12x12mm body QFP (FP)

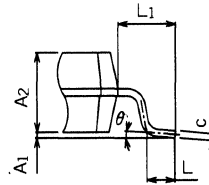
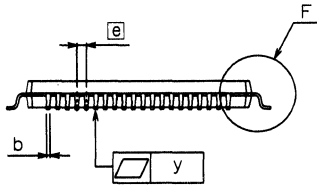
EIAJ Package Code	JEDEC Code	Weight(g)
* QFP80-P-1212	-	0.44

Scale : 2.5/1



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	1.7
A1	0	0.1	0.2
A2	-	1.4	-
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	11.9	12.0	12.1
E	11.9	12.0	12.1
e	-	0.5	-
Hd	13.8	14.0	14.2
HE	13.8	14.0	14.2
L	0.3	0.5	0.7
L1	-	1.0	-
y	-	-	0.1
θ	0°	-	10°
b2	-	0.225	-
l2	-	1.0	-
MD	-	12.4	-
ME	-	12.4	-



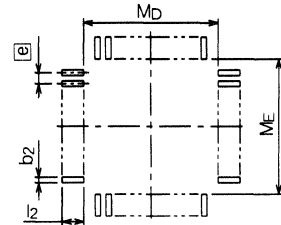
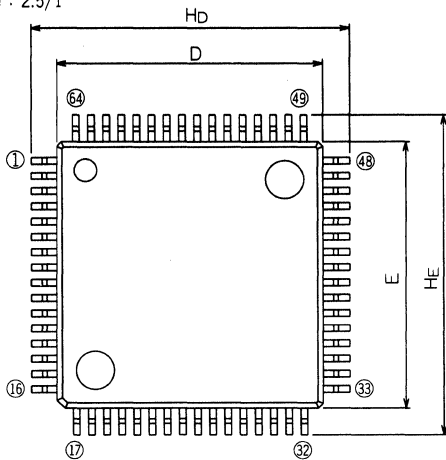
Detail F

64P6N-A

Plastic 64pin 14x14mm body QFP

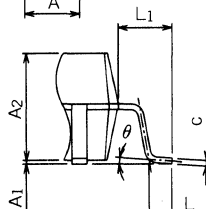
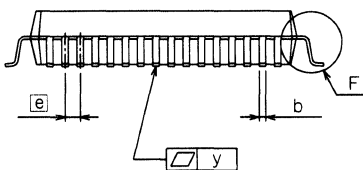
EIAJ Package Code	JEDEC Code	Weight(g)
* QFP64-P-1414	-	1.11

Scale : 2.5/1



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	3.05
A1	0	0.1	0.2
A2	-	2.8	-
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	13.8	14.0	14.2
e	-	0.8	-
Hd	16.5	16.8	17.1
HE	16.5	16.8	17.1
L	0.4	0.6	0.8
L1	-	1.4	-
y	-	-	0.1
θ	0°	-	10°
b2	-	0.5	-
l2	-	1.3	-
MD	-	14.6	-
ME	-	14.6	-



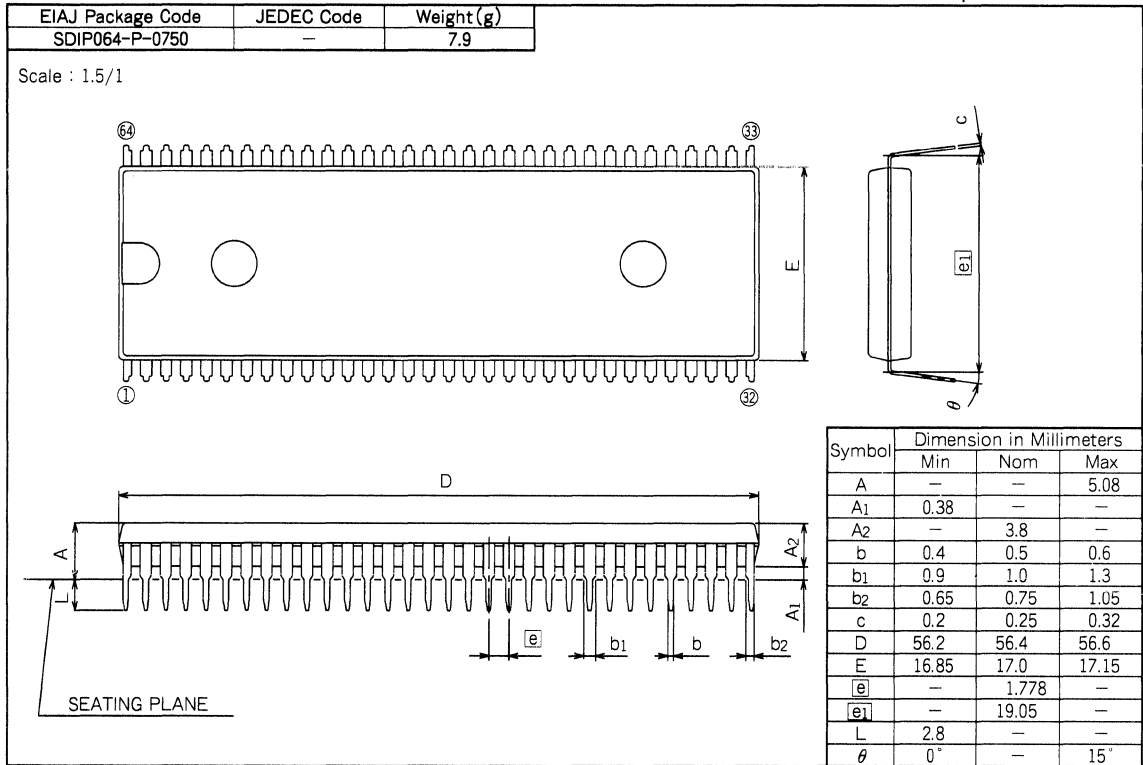
Detail F

APPENDIX

Appendix 4. Package outlines

64P4B

Plastic 64pin 750mil SDIP



APPENDIX

Appendix 5. Setting example of unused pins

Appendix 5. Setting example of unused pins

Table 4 Setting example of unused pins

Pin	Setting
Ports P4 ₃ -P4 ₇ , P5-P8	Set to input mode and connect to V _{ss} through a resistor (pull-down).
$\overline{\text{BHE}}$ (Note 1), ALE (Note 2), $\overline{\text{HLDA}}$, X _{OUT} (Note 3)	Open.
$\overline{\text{HOLD}}$, RDY	Connect to V _{cc} through a resistor (pull-up).
AV _{cc}	Connect to V _{cc} .
AV _{ss} , V _{REF}	Connect to V _{ss} .

Note 1: When BYTE="H".

Note 2: When BYTE="H" and address space is 64K bytes.

Note 3: When external clock is input to X_{IN} pin.

Note 4: M37730 group dose not have port P7, AV_{cc} pin, AV_{ss} pin and V_{REF} pin.

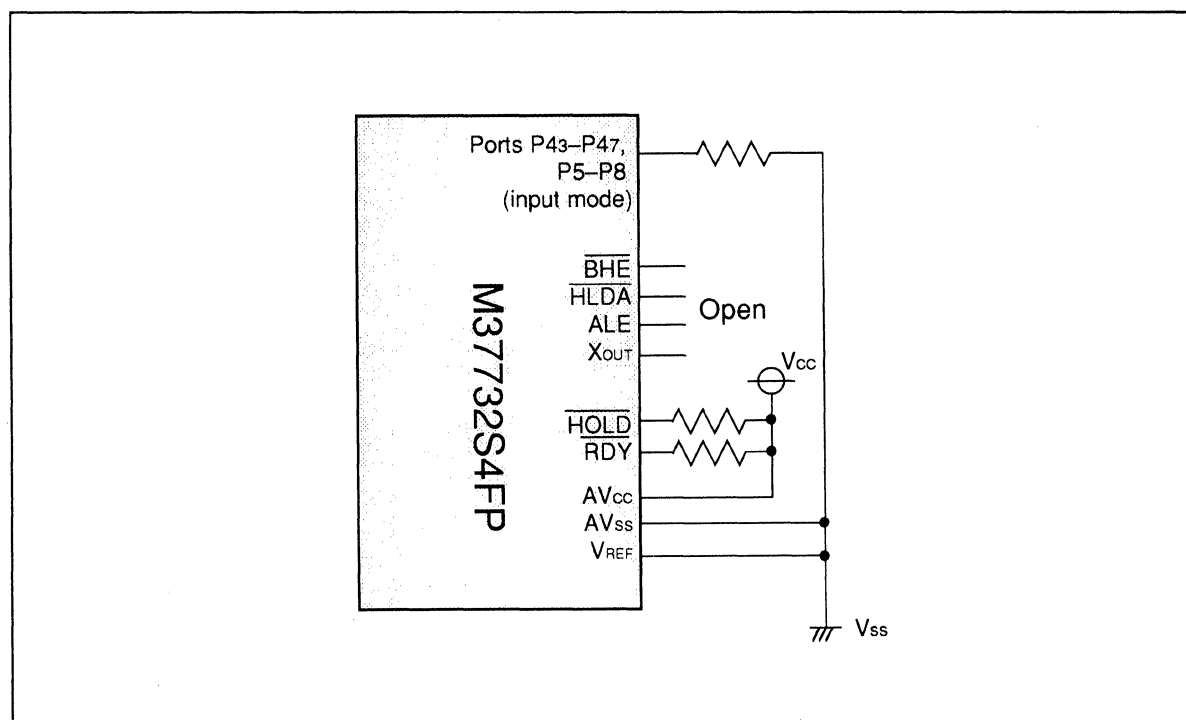


Fig. 1 Setting example of unused pins

APPENDIX

Appendix 6. Print foot pattern examples

Appendix 6. Print foot pattern examples

Examples of target board print pattern and the IC sockets are described below.

The print foot patterns are presented as examples. The user should consider the mounting condition and the standard when specifying the dimension for mass production board.

1. M37732S4/S4A/S4BFP (package: 80P6N-A)

(1) Print foot pattern example

Figure 7 shows the print foot pattern example for the 80P6N package. This pattern can mount the 80P6N package, and IC61-0804-034 and IC61-0804-046 sockets provided by YAMAICHI ELECTRONICS Co., Ltd. which are described in "(2) IC socket".

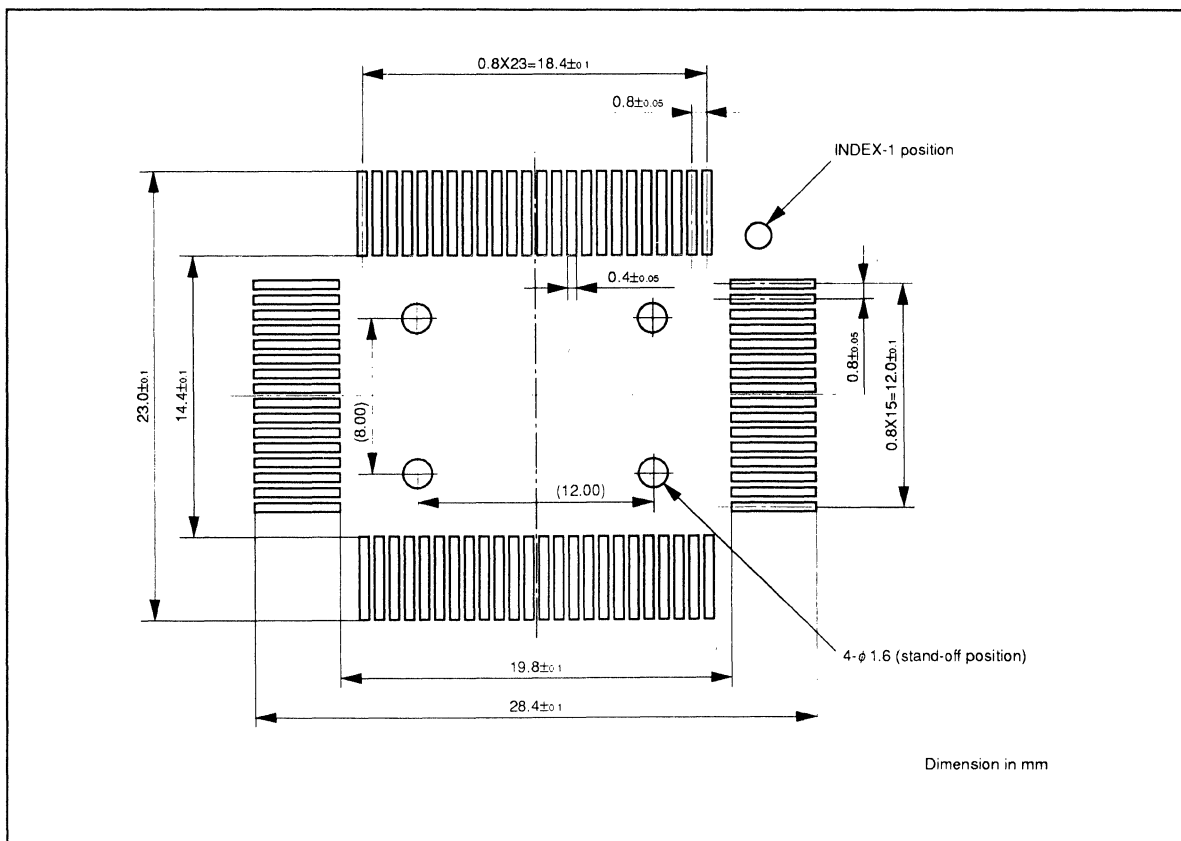


Fig. 2 Print foot pattern example for 80P6N package

APPENDIX

Appendix 6. Print foot pattern examples

(2) IC socket

The following IC sockets provided by YAMAICHI ELECTRONICS Co., Ltd. can be connected to the M37732 group support pod M37732T-HPD of the MELPS 7700 emulator PC4816.

- 80-pin LCC socket provided by YAMAICHI ELECTRONICS Co., Ltd.

Type name : IC61-0804-034

Type name : IC61-0804-046

The profile of IC61-0804-046 is higher than IC61-0804-034. IC61-0804-046 is an exclusive use of mounting on printed circuit board and easy to solder.

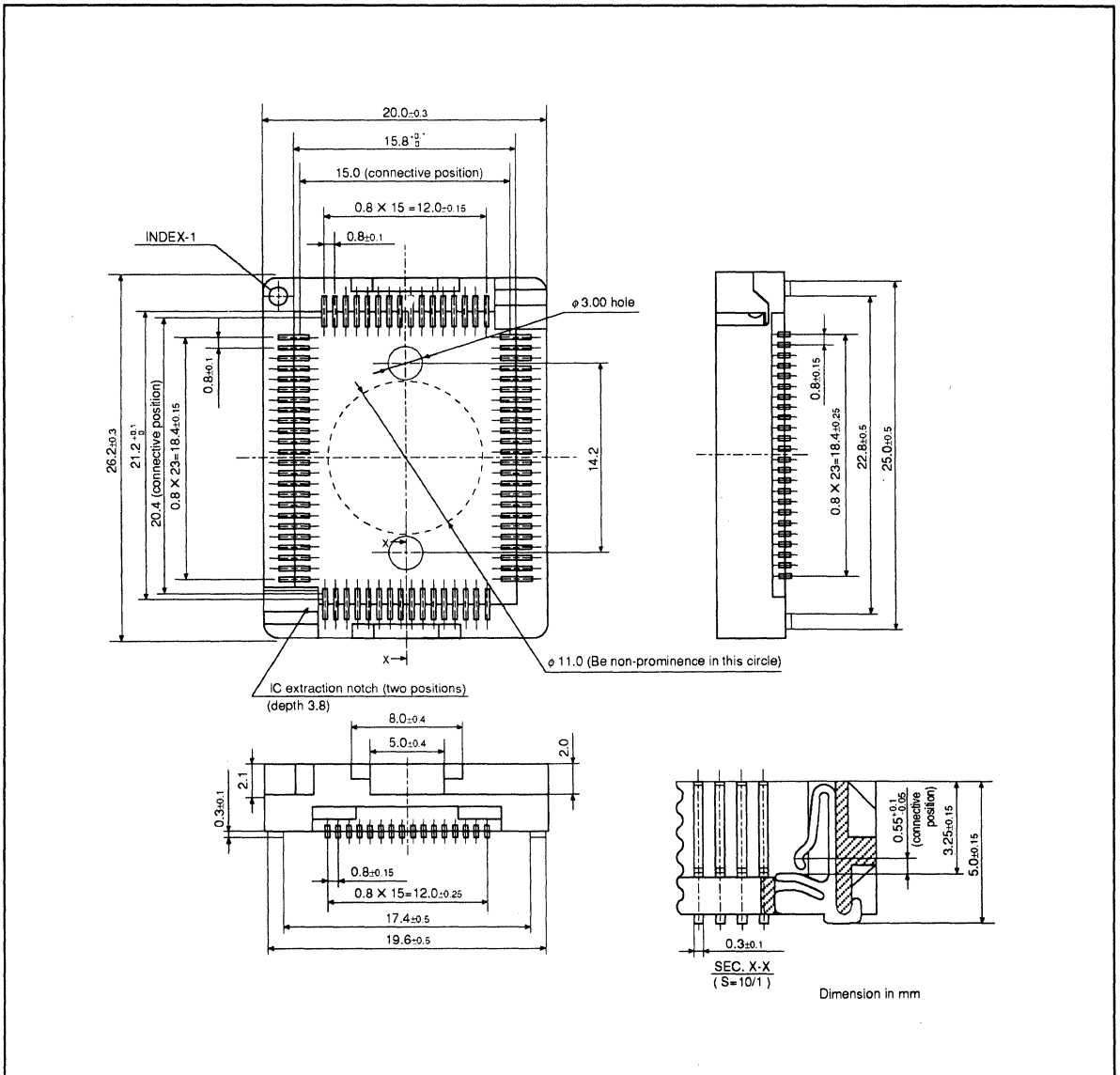


Fig. 3 IC socket (type name : IC61-0804-034) by YAMAICHI ELECTRONICS Co., Ltd. outline

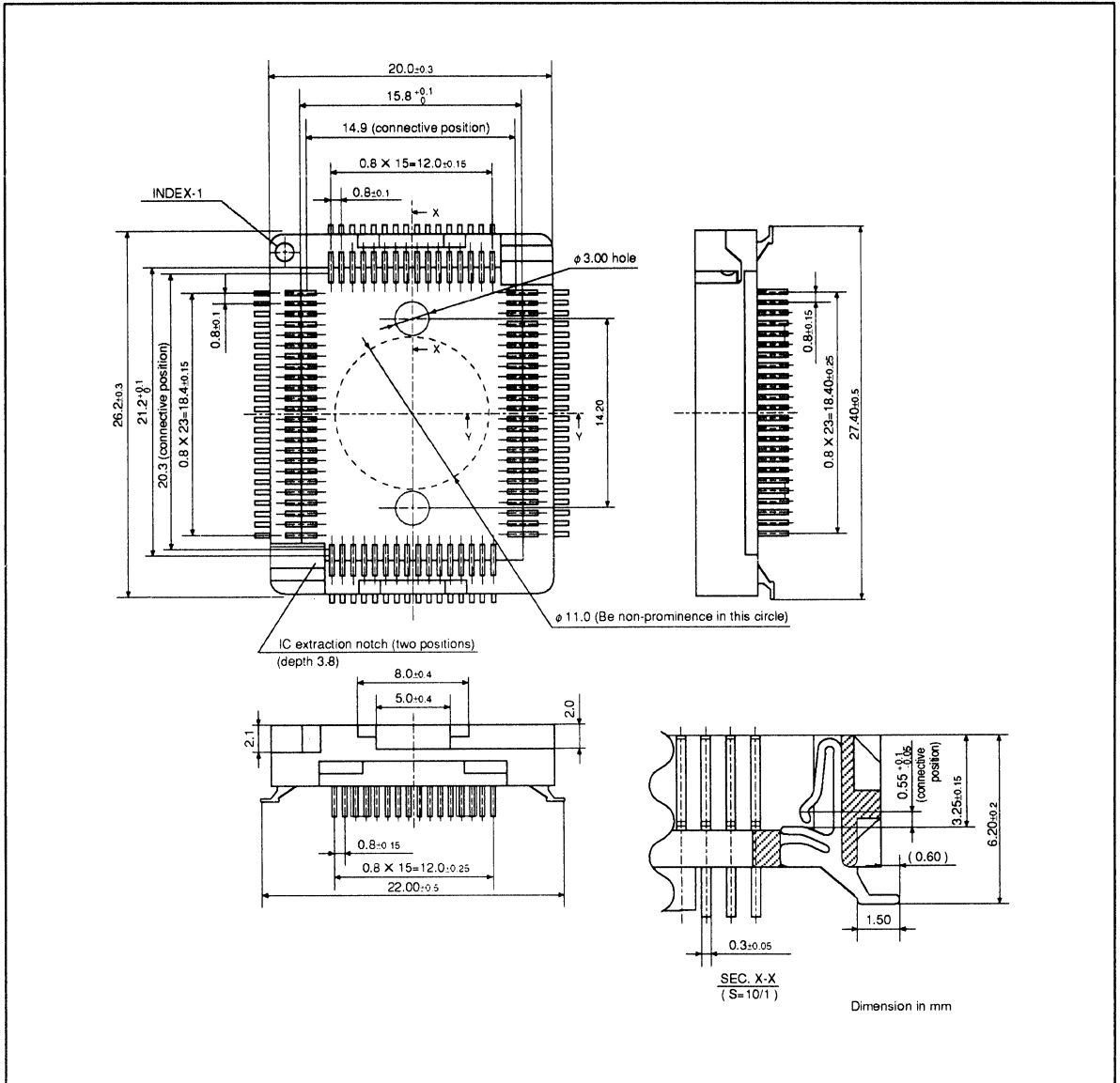


Fig. 4 IC socket (type name : IC61-0804-046) by YAMAICHI ELECTRONICS Co., Ltd. outline

2. M37732S4LGP (package: 80P6S-A)

(1) Print foot pattern example

Figure 10 shows the print foot pattern example for the 80P6S package. This pattern can mount the 80P6S package, and the 80P6-80P6S conversion adapter which is described in "(2) Conversion adapter".

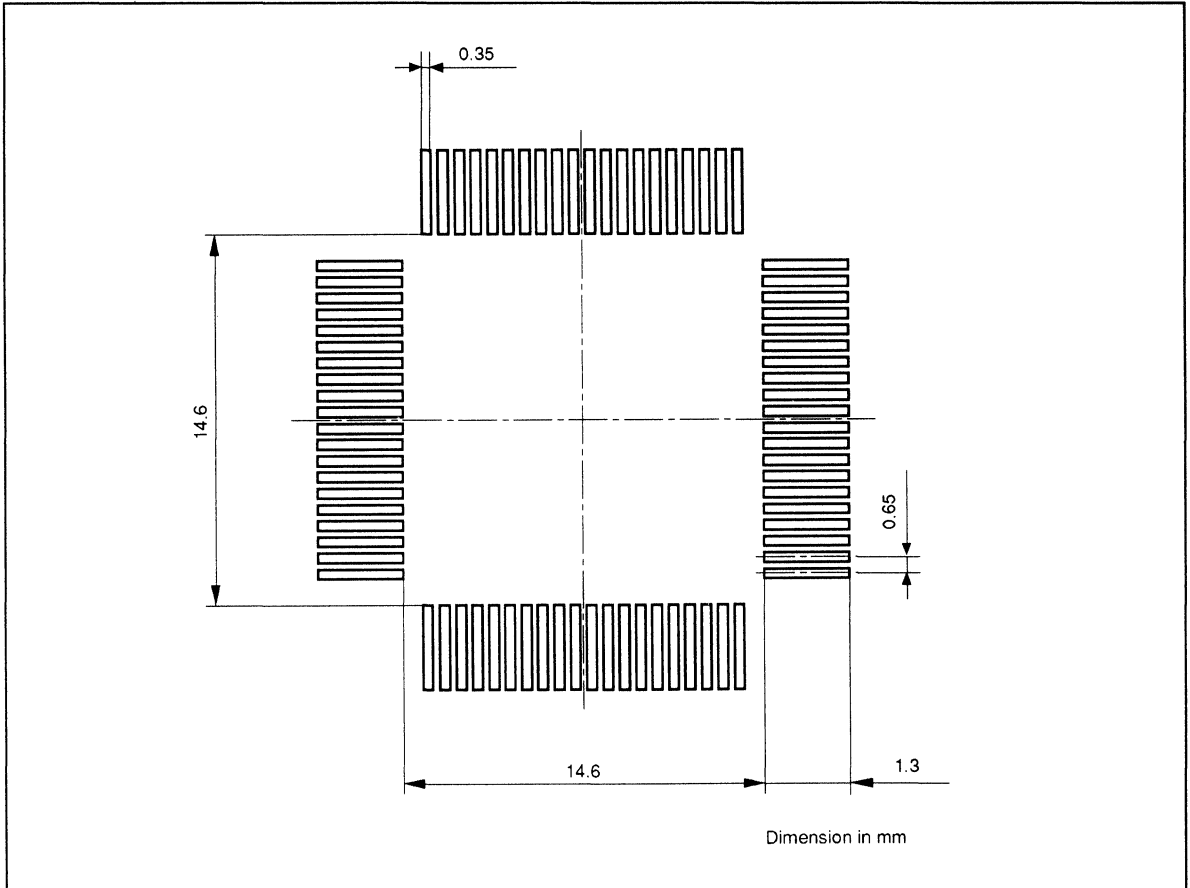


Fig. 5 Print foot pattern example for 80P6S package

(2) Conversion adapter

The 80P6-80P6S conversion adapter shown in Figure 11 converts an 80LCC connector to 80P6S pattern.

The tips of the M37732 group support pod M37732T-HPD of the MELPS 7700 emulator PC4816 can be connected to the 80LCC connector on this adapter to perform emulation in the range of $5V \pm 5\%$. Note that emulation cannot be performed outside the voltage range of $5V \pm 5\%$ (3V for example).

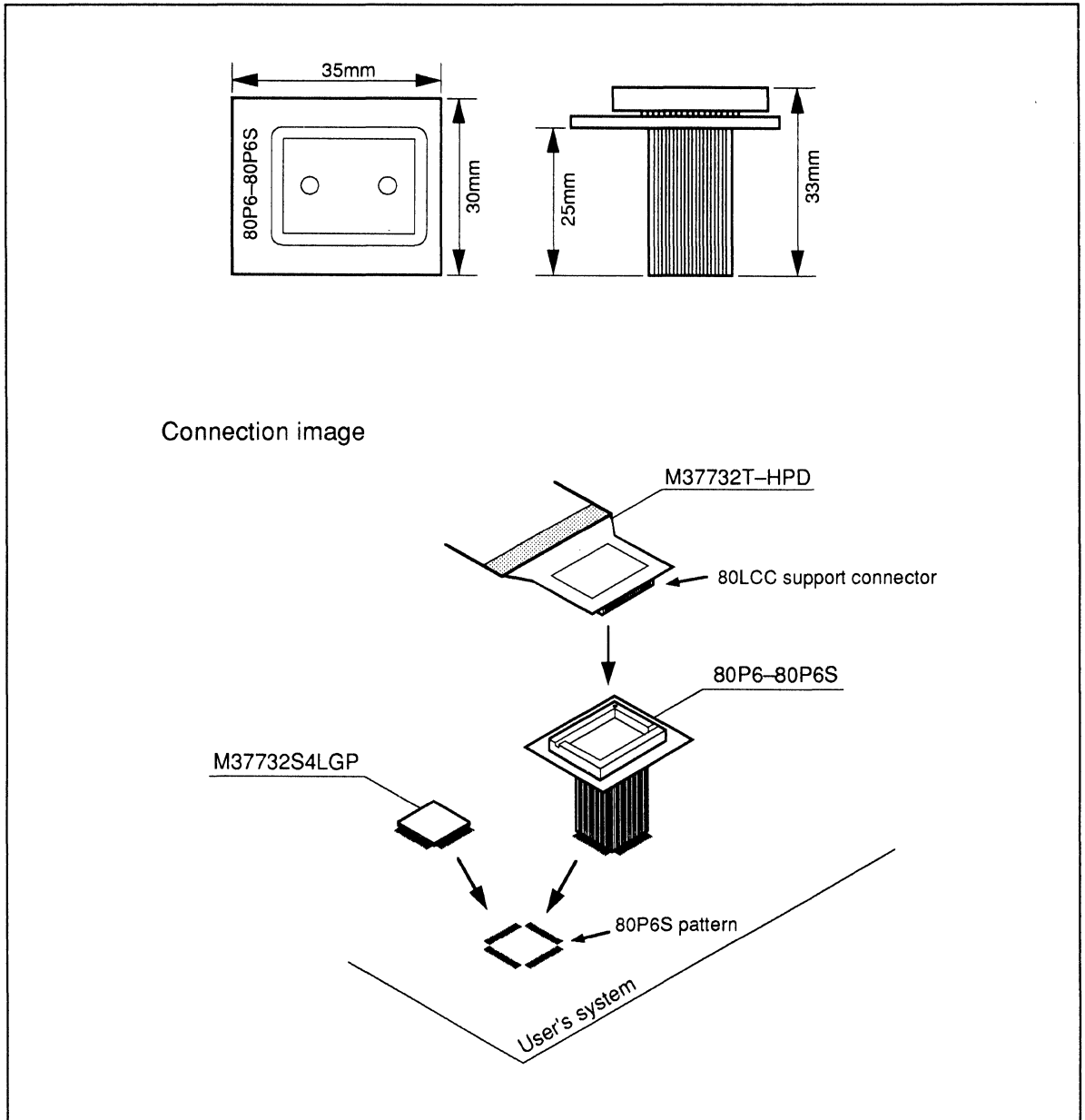


Fig. 6 80P6-80P6S conversion adapter

3. M37732S4LHP (package: 80P6D-A)

(1) Print foot pattern example

Figure 12 shows the print foot pattern example for the 80P6D package. This pattern can mount the 80P6D package.

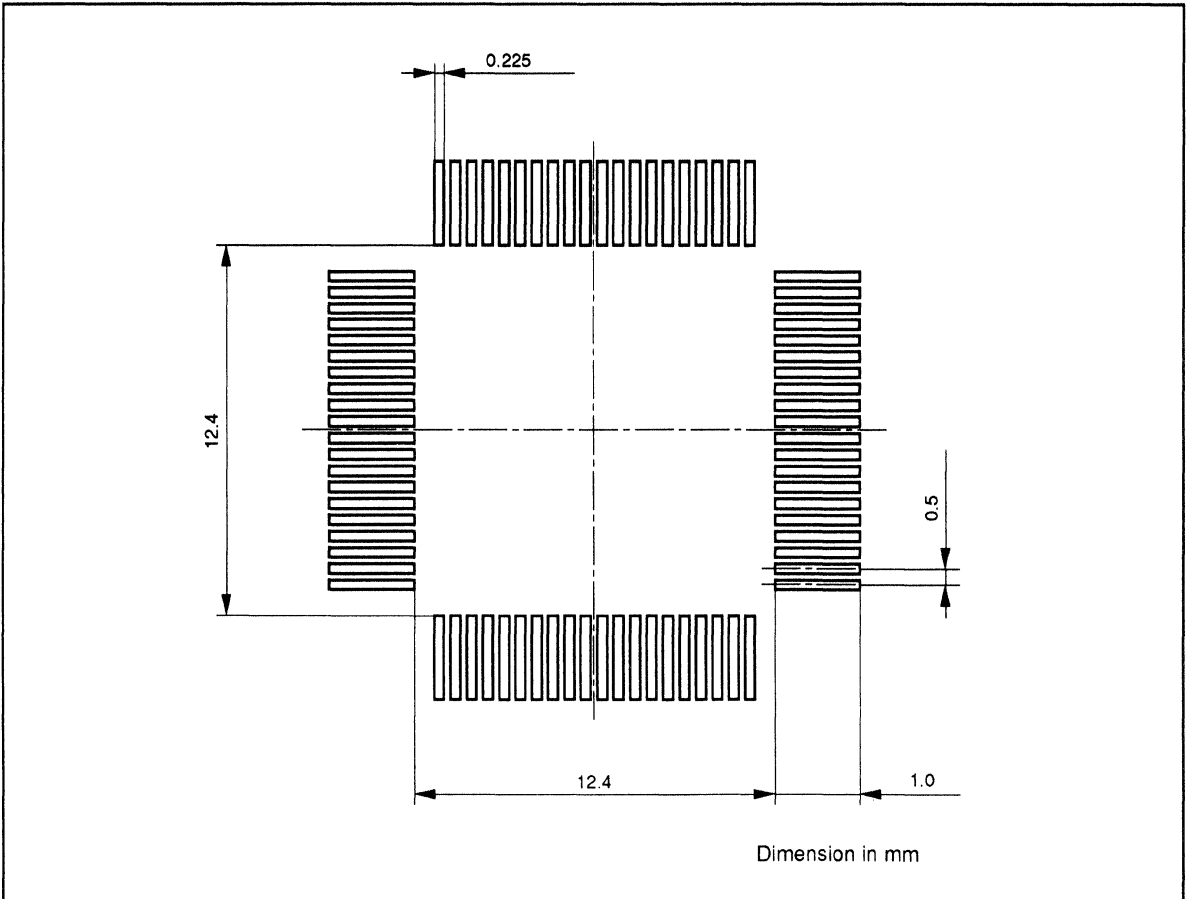


Fig. 7 Print foot pattern example for 80P6D package

(2) Conversion adapter

Conversion adapter 80LCC-QSD for converting 80LCC connector to 80P6D pattern is also available. This conversion adapter can also be used to perform emulation similar to the conversion adapter 80P6-80P6S described on the previous page. For more information on conversion adapters, contact the address at the end of this manual.

4. M37730S2/S2A/S2BFP (package: 64P6N-A)

(1) Print foot pattern example

Figure 13 shows the print foot pattern example for the 64P6N package. This pattern can mount the 64P6N package, and IC61-0644-052 socket provided by YAMAICHI ELECTRONICS Co., Ltd. which is described in "(2) IC socket".

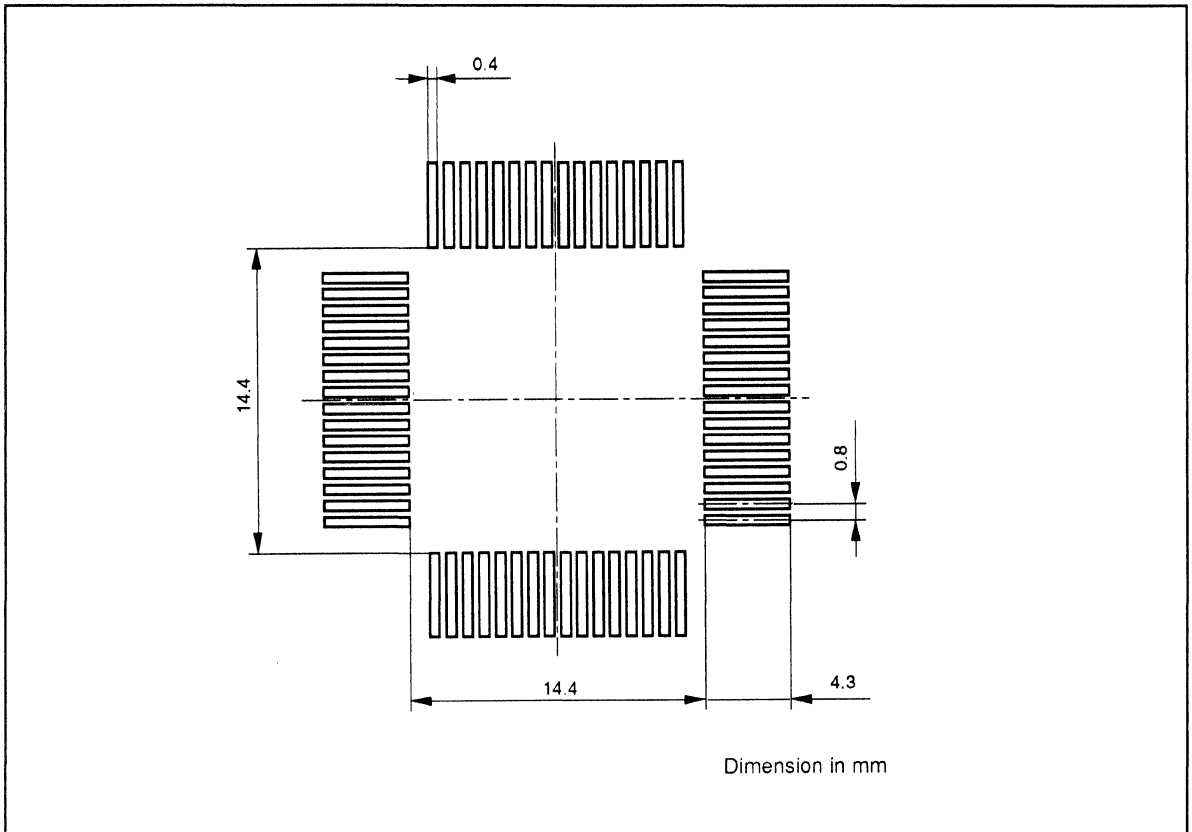


Fig. 8 Print foot pattern example for 64P6N package

(2) IC socket

The following IC socket provided by YAMAICHI ELECTRONICS Co., Ltd. can be connected to the M37730 group support pod M37730T-HPD of the MELPS 7700 emulator PC4816.

- 64-pin LCC socket provided by YAMAICHI ELECTRONICS Co., Ltd.

Type name : IC61-0644-052

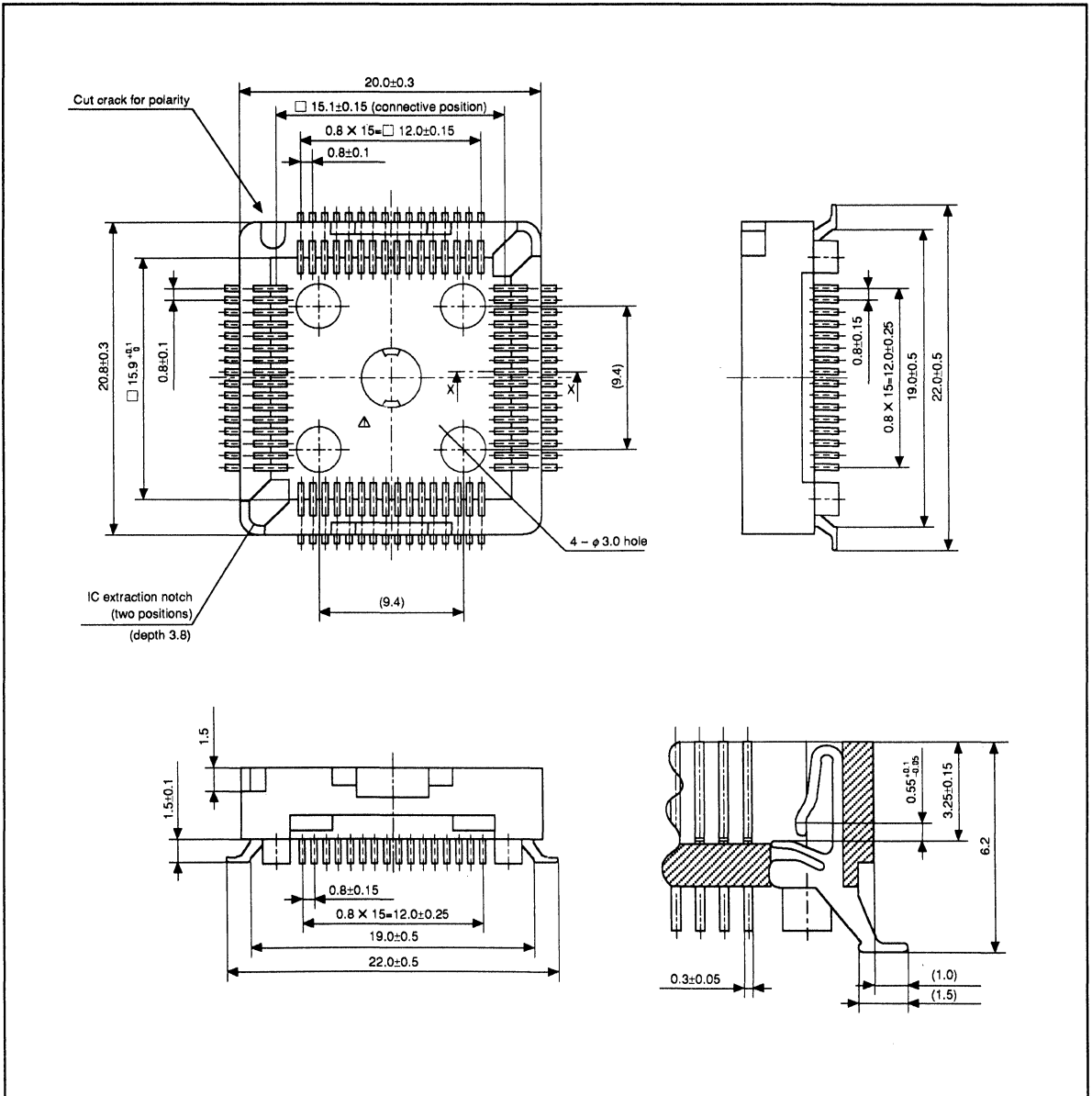


Fig. 9 IC socket (type name : IC61-0644-052) by YAMAICHI ELECTRONICS Co., Ltd. outline

APPENDIX

Appendix 7. System development precautions

Appendix 7. System development precautions

When developing an M37732 group or M37730 group system, be sure to low the V_{SS} line impedance (V_{SS} line reinforced). This is necessary to drive the address bus at 24 bits.

Figures 15 and 16 show the print foot pattern examples with the V_{SS} line reinforced.

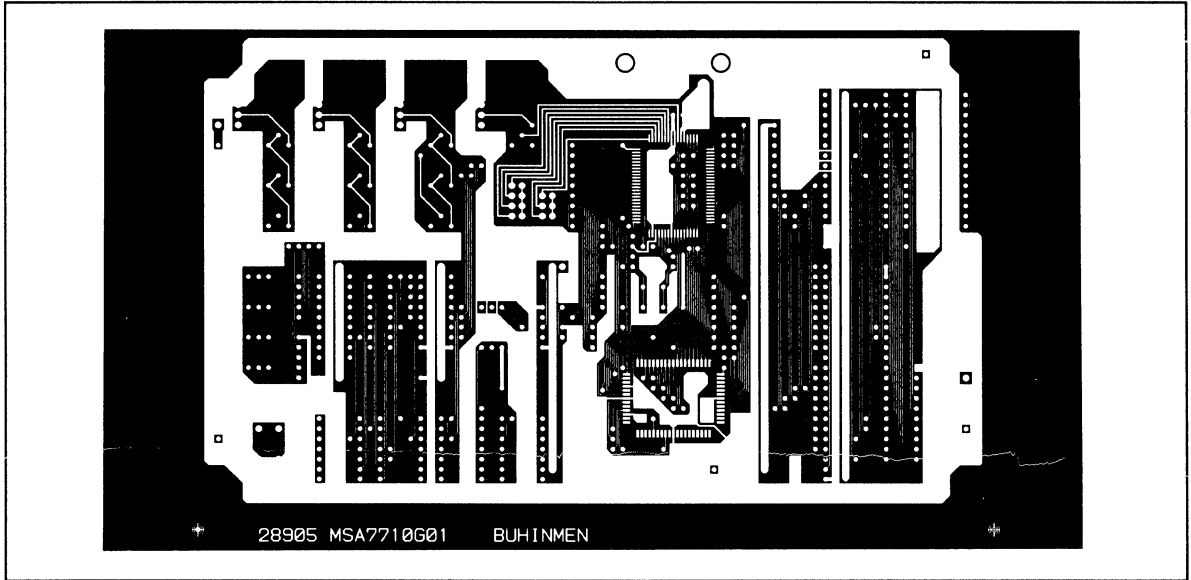


Fig. 10 Print foot pattern example (parts view)

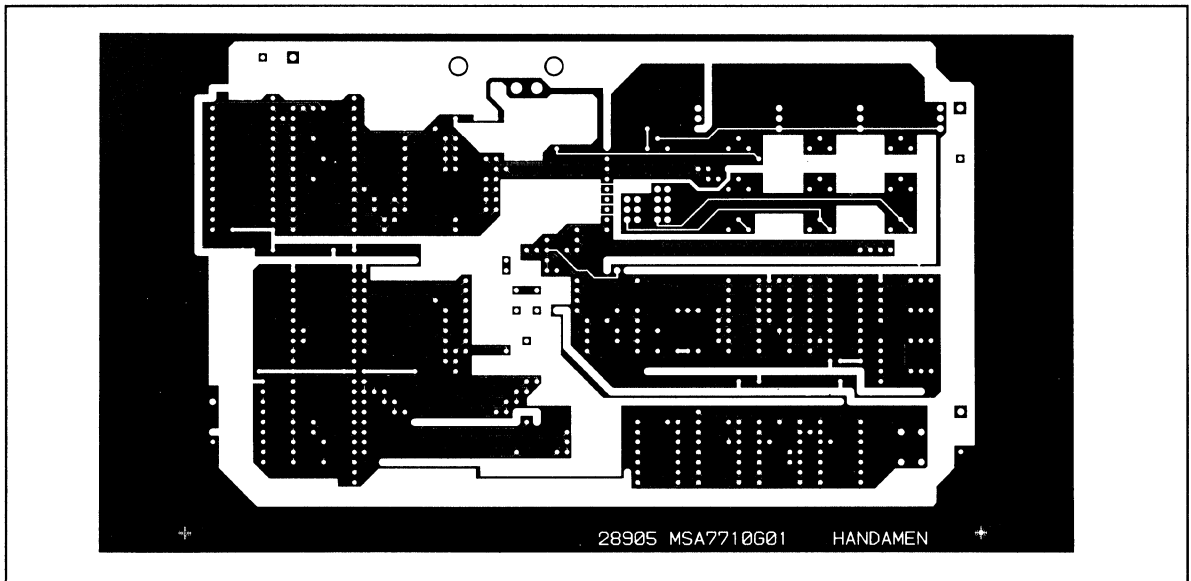


Fig. 11 Print foot pattern example (solder view)

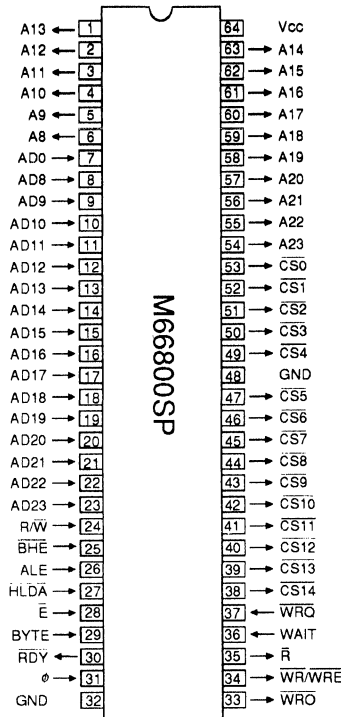
Appendix 8. M66800SP/FP

1. Overview

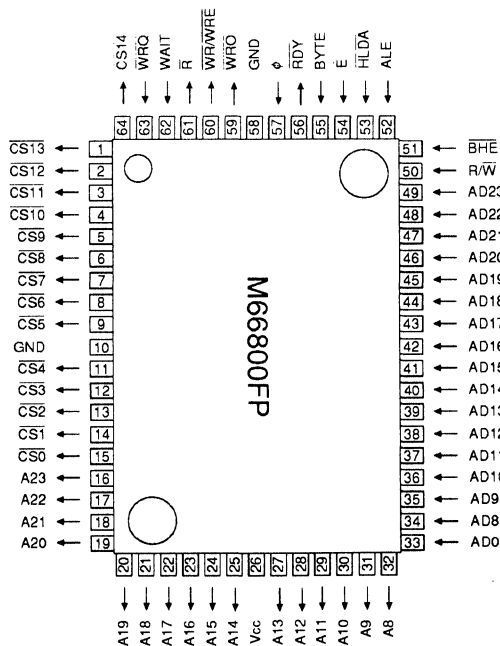
The M66800SP/FP is a 16M-byte external memory control IC for the M37732 group. It uses a MAST (Mitsubishi Advanced Schottky TTL) process to enable fast operation.

2. Features

- Supports no wait operation of the M37732 group under the following conditions:
 - (1) Using 120ns (Max.) access time memory at external clock input frequency of 12MHz
 - (2) Using 100ns (Max.) access time memory at external clock input frequency of 13MHz
- Internal read/write control circuit with independent read/write output
- Two internal octal latch circuits for address
- Internal chip selectable circuit that can fully decode 16M bytes
- Internal Ready signal generating circuit
- Pin assignment that facilitates connection to M37732 group



Outline 64P4B



Outline 64P6W

Fig. 12 M66800SP/FP pin configuration (top view)

Appendix 9. Machine instructions



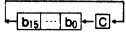
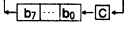
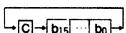
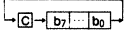
MACHINE INSTRUCTIONS

Symbol	Function	Details	Addressing mode																					
			IMP	IMM	A	DIR	DIR,b	DIR,X	DIR,Y	(DIR)	(DIR,X)	(DIR),Y												
			op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #											
ADC (Note 1,2)	$Acc.C \leftarrow Acc + M + C$	Adds the carry, the accumulator and the memory contents. The result is entered into the accumulator. When the D flag is "0", binary additions is done, and when the D flag is "1", decimal addition is done.		69	2	2		65	4	2		75	5	2		72	6	2	61	7	2	71	8	2
				42	4	3		42	6	3		42	7	3		42	8	3	42	9	3	42	10	3
				69				65				75				72			61			71		
AND (Note 1,2)	$Acc \leftarrow Acc \wedge M$	Obtains the logical product of the contents of the accumulator and the contents of the memory. The result is entered into the accumulator.		29	2	2		25	4	2		35	5	2		32	6	2	21	7	2	31	8	2
				42	4	3		42	6	3		42	7	3		42	8	3	42	9	3	42	10	3
				29				25				35				32			21			31		
ASL (Note 1)	$m=0$ $C \leftarrow [b_{15} \dots b_0] \leftarrow 0$ $m=1$ $C \leftarrow [b_7 \dots b_0] \leftarrow 0$	Shifts the accumulator or the memory contents one bit to the left. "0" is entered into bit 0 of the accumulator or the memory. The contents of bit 15 (bit 7 when the m flag is "1") of the accumulator or memory before shift is entered into the C flag.					0A	2	1		06	7	2		16	7	2							
								42	4	2														
								0A																
BBC (Note 3,5)	$Mb=0?$	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "0".																						
BBS (Note 3,5)	$Mb=1?$	Tests the specified bit of the memory. Branches when all the contents of the specified bit is "1".																						
BCC (Note 3)	$C=0?$	Branches when the contents of the C flag is "0".																						
BCS (Note 3)	$C=1?$	Branches when the contents of the C flag is "1".																						
BEQ (Note 3)	$Z=1?$	Branches when the contents of the Z flag is "1".																						
BMI (Note 3)	$N=1?$	Branches when the contents of the N flag is "1".																						
BNE (Note 3)	$Z=0?$	Branches when the contents of the Z flag is "0".																						
BPL (Note 3)	$N=0?$	Branches when the contents of the N flag is "0".																						
BRA (Note 4)	$PC \leftarrow PC \pm \text{offset}$ $PG \leftarrow PG + 1$ (carry occurred) $PG \leftarrow PG - 1$ (borrow occurred)	Jumps to the address indicated by the program counter plus the offset value.																						
BRK	$PC \leftarrow PC + 2$ $M(S) \leftarrow PG$ $S \leftarrow S - 1$ $M(S) \leftarrow PC_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PC_L$ $S \leftarrow S - 1$ $M(S) \leftarrow PS_H$ $S \leftarrow S - 1$ $M(S) \leftarrow PS_L$ $S \leftarrow S - 1$ $I \leftarrow 1$ $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ $PG \leftarrow 00_{16}$	Executes software interruption.	00	15	2																			
BVC (Note 3)	$V=0?$	Branches when the contents of the V flag is "0".																						
BVS (Note 3)	$V=1?$	Branches when the contents of the V flag is "1".																						
CLB (Note 5)	$Mb \leftarrow 0$	Makes the contents of the specified bit in the memory "0".										14	8	3										
CLC	$C \leftarrow 0$	Makes the contents of the C flag "0".	18	2	1																			
CLI	$I \leftarrow 0$	Makes the contents of the I flag "0".	58	2	1																			
CLM	$m \leftarrow 0$	Makes the contents of the m flag "0".	08	2	1																			
CLP	$PSb \leftarrow 0$	Specifies the bit position in the processor status register by the bit pattern of the second byte in the instruction, and sets "0" in that bit.					C2	4	2															
CLV	$V \leftarrow 0$	Makes the contents of the V flag "0".	B8	2	1																			
CMP (Note 1,2)	$Acc \leftarrow M$	Compares the contents of the accumulator with the contents of the memory.		C9	2	2		C5	4	2		D5	5	2		D2	6	2	C1	7	2	D1	8	2
				42	4	3		42	6	3		42	7	3		42	8	3	42	9	3	42	10	3
				C9				C5				D5				D2			C1			D1		

Symbol	Function	Details	Addressing mode																												
			IMP		IMM		A		DIR		DIR,b		DIR,X		DIR,Y		(DIR)		(DIR,X)		(DIR),Y										
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #							
LDA (Note 1,2)	Acc ← M	Enters the contents of the memory into the accumulator.			A9	2	2			A5	4	2			B5	5	2			B2	6	2	A1	7	2	B1	8	2			
					42	4	3			42	6	3			42	7	3			42	8	3	42	9	3	42	10	3			
					A9					A5					B5					B2			A1			B1					
LDM (Note 5)	M ← IMM	Enters the immediate value into the memory.								64	4	3			74	5	3														
LDT	DT ← IMM	Enters the immediate value into the data bank register.			89	5	3																								
					C2																										
LDX (Note 2)	X ← M	Enters the contents of the memory into index register X.			A2	2	2			A6	4	2								B6	5	2									
LDY (Note 2)	Y ← M	Enters the contents of the memory into index register Y.			A0	2	2			A4	4	2			B4	5	2														
LSR (Note 1)	m=0 0 ← [b ₁₅ ... b ₀] → C m=1 0 ← [b ₇ ... b ₀] → C	Shifts the contents of the accumulator or the contents of the memory one bit to the right. The bit 0 of the accumulator or the memory is entered into the C flag. "0" is entered into bit 15 (bit 7 when the m flag is "1".)						4A	2	1	46	7	2			56	7	2													
					42	4	2			4A																					
MPY (Note 2,11)	B, A ← A * M	Multiplies the contents of accumulator A and the contents of the memory. The higher order of the result of operation are entered into accumulator B, and the lower order into accumulator A.			89	16	3			89	18	3			89	19	3			89	20	3	89	21	3	89	22	3			
					09					05					15					12			01			11					
MVN (Note 8)	Mn+i ← Mm+i	Transmits the data block. The transmission is done from the lower order address of the block.																													
MVP (Note 9)	Mn-i ← Mm-i	Transmits the data block. Transmission is done from the higher order address of the data block.																													
NOP	PC ← PC+1	Advances the program counter, but performs nothing else.	EA	2	1																										
ORA (Note 1,2)	Acc ← Acc VM	Logical sum per bit of the contents of the accumulator and the contents of the memory is obtained. The result is entered into the accumulator.						09	2	2			05	4	2			15	5	2			12	6	2	01	7	2	11	8	2
					42	4	3			42	6	3			42	7	3			42	8	3	42	9	3	42	10	3			
					09					05					15					12			01			11					
PEA	M(S) ← IMM ₂ S ← S-1 M(S) ← IMM ₁ S ← S-1	The 3rd and the 2nd bytes of the instruction are saved into the stack, in this order.																													
PEI	M(S) ← M((DPR) + IMM + 1) S ← S-1 M(S) ← M((DPR) + IMM) S ← S-1	Specifies 2 sequential bytes in the direct page in the 2nd byte of the instruction, and saves the contents into the stack.																													
PER	EAR ← PC + IMM ₂ IMM ₁ M(S) ← EAR _H S ← S-1 M(S) ← EAR _L S ← S-1	Regards the 2nd and 3rd bytes of the instruction as 16-bit numerals, adds them to the program counter, and saves the result into the stack.																													
PHA	m=0 M(S) ← A _H S ← S-1 M(S) ← A _L S ← S-1 m=1 M(S) ← A _L S ← S-1	Saves the contents of accumulator A into the stack.																													
PHB	m=0 M(S) ← B _H S ← S-1 M(S) ← B _L S ← S-1 m=1 M(S) ← B _L S ← S-1	Saves the contents of accumulator B into the stack.																													

APPENDIX

Appendix 9. Machine instructions

Symbol	Function	Details	Addressing mode																									
			IMP		IMM		A		DIR		DIR,b		DIR,X		DIR,Y		(DIR)		(DIR,X)		(DIR),Y							
			op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n	op	n		
PLY	$x=0$ $S \leftarrow S+1$ $Y_L \leftarrow M(S)$ $S \leftarrow S+1$ $Y_H \leftarrow M(S)$ $x=1$ $S \leftarrow S+1$ $Y_L \leftarrow M(S)$	Restores the contents of the stack on the index register Y.																										
PSH (Note 6)	$M(S) \leftarrow A, B, X \dots$	Saves the registers among accumulator, index register, direct page register, data bank register, program bank register, or processor status register, specified by the bit pattern of the second byte of the instruction into the stack.																										
PUL (Note 7)	$A, B, X \dots \leftarrow M(S)$	Restores the contents of the stack to the registers among accumulator, index register, direct page register, data bank register, or processor status register, specified by the bit pattern of the second byte of the instruction.																										
RLA (Note 13)	$m=0$ n bit rotate left  $m=1$ n bit rotate left 	Rotates the contents of the accumulator A, n bits to the left.			89	6	3																					
ROL (Note 1)	$m=0$  $m=1$ 	Links the accumulator or the memory to C flag, and rotates result to the left by 1 bit.					2A	2	1	26	7	2			36	7	2											
ROR (Note 1)	$m=0$  $m=1$ 	Links the accumulator or the memory to C flag, and rotates result to the right by 1 bit.					6A	2	1	66	7	2			76	7	2											
RTI	$S \leftarrow S+1$ $PS_L \leftarrow M(S)$ $S \leftarrow S+1$ $PS_H \leftarrow M(S)$ $S \leftarrow S+1$ $PC_L \leftarrow M(S)$ $S \leftarrow S+1$ $PC_H \leftarrow M(S)$ $S \leftarrow S+1$ $PG \leftarrow M(S)$	Returns from the interruption routine.	40	11	1																							
RTL	$S \leftarrow S+1$ $PC_L \leftarrow M(S)$ $S \leftarrow S+1$ $PC_H \leftarrow M(S)$ $S \leftarrow S+1$ $PG \leftarrow M(S)$	Returns from the subroutine. The contents of the program bank register are also restored.	8B	8	1																							
RTS	$S \leftarrow S+1$ $PC_L \leftarrow M(S)$ $S \leftarrow S+1$ $PC_H \leftarrow M(S)$	Returns from the subroutine. The contents of the program bank register are not restored.	60	5	1																							
SBC (Note 1,2)	$Acc, C \leftarrow Acc - M - \bar{C}$	Subtracts the contents of the memory and the borrow from the contents of the accumulator.			E9	2	2			E5	4	2			F5	5	2			F2	6	2	E1	7	2	F1	8	2
					42	4	3			42	6	3			42	7	3			42	8	3	42	9	3	42	10	3
					E9					E5					F5					F2		E1		F1				

Symbols in machine instructions table

Symbol	Description	Symbol	Description
IMP	Implied addressing mode	∇	Exclusive OR
IMM	Immediate addressing mode	—	Negation
A	Accumulator addressing mode	←	Movement to the arrow direction
DIR	Direct addressing mode	ACC	Accumulator
DIR, b	Direct bit addressing mode	ACC _H	Accumulator's upper 8 bits
DIR, X	Direct indexed X addressing mode	ACC _L	Accumulator's lower 8 bits
DIR, Y	Direct indexed Y addressing mode	A	Accumulator A
(DIR)	Direct indirect addressing mode	A _H	Accumulator A's upper 8 bits
(DIR, X)	Direct indexed X indirect addressing mode	A _L	Accumulator A's lower 8 bits
(DIR), Y	Direct indirect indexed Y addressing mode	B	Accumulator B
L (DIR)	Direct indirect long addressing mode	B _H	Accumulator B's upper 8 bits
L (DIR), Y	Direct indirect long indexed Y addressing mode	B _L	Accumulator B's lower 8 bits
ABS	Absolute addressing mode	X	Index register X
ABS, b	Absolute bit addressing mode	X _H	Index register X's upper 8 bits
ABS, X	Absolute indexed X addressing mode	X _L	Index register X's lower 8 bits
ABS, Y	Absolute indexed Y addressing mode	Y	Index register Y
ABL	Absolute long addressing mode	Y _H	Index register Y's upper 8 bits
ABL, X	Absolute long indexed X addressing mode	Y _L	Index register Y's lower 8 bits
(ABS)	Absolute indirect addressing mode	S	Stack pointer
L (ABS)	Absolute indirect long addressing mode	PC	Program counter
(ABS, X)	Absolute indexed X indirect addressing mode	PC _H	Program counter's upper 8 bits
STK	Stack addressing mode	PC _L	Program counter's lower 8 bits
REL	Relative addressing mode	PG	Program bank register
DIR, b, REL	Direct bit relative addressing mode	DT	Data bank register
ABS, b, REL	Absolute bit relative addressing mode	DPR	Direct page register
SR	Stack pointer relative addressing mode	DPR _H	Direct page register's upper 8 bits
(SR), Y	Stack pointer relative indirect indexed Y addressing mode	DPR _L	Direct page register's lower 8 bits
BLK	Block transfer addressing mode	PS	Processor status register
C	Carry flag	PS _H	Processor status register's upper 8 bits
Z	Zero flag	PS _L	Processor status register's lower 8 bits
I	Interrupt disable flag	PS _b	Processor status register's b-th bit
D	Decimal operation mode flag	M(S)	Contents of memory at address indicated by stack pointer
x	Index register length selection flag	M _b	b-th memory location
m	Data length selection flag	AD _G	Value of 24-bit address's upper 8-bit (A ₂₃ ~A ₁₆)
V	Overflow flag	AD _H	Value of 24-bit address's middle 8-bit (A ₁₅ ~A ₈)
N	Negative flag	AD _L	Value of 24-bit address's lower 8-bit (A ₇ ~A ₀)
IPL	Processor interrupt priority level	op	Operation code
+	Addition	n	Number of cycle
-	Subtraction	#	Number of byte
*	Multiplication	i	Number of transfer byte or rotation
/	Division	i ₁ , i ₂	Number of registers pushed or pulled
∧	Logical AND		
∨	Logical OR		

APPENDIX

Appendix 9. Machine instructions

The number of cycles shown in the table is described in case of the fastest mode for each instruction. The number of cycles shown in the table is calculated for $DPR_L=0$. The number of cycles in the addressing mode concerning the DPR when $DPR_L \neq 0$ must be incremented by 1. The number of cycles shown in the table differs according to the bytes fetched into the instruction queue buffer, or according to whether the memory read/write address is odd or even. It also differs when the external region memory is accessed by $BYTE="H"$.

Note 1. The operation code at the upper row is used for accumulator A, and the operation at the lower row is used for accumulator B.

Note 2. When setting flag $m=0$ to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 3. The number of cycles increments by 2 when branching.

Note 4. The operation code on the upper row is used for branching in the range of $-128 \sim +127$, and the operation code on the lower row is used for branching in the range of $-32768 \sim +32767$.

Note 5. When handling 16-bit data with flag $m=0$, the byte in the table is incremented by 1.

Note 6.

Type of register	A	B	X	Y	DPR	DT	PG	PS
Number of cycles	2	2	2	2	2	1	1	2

The number of cycles corresponding to the register to be pushed are added. The number of cycles when no pushing is done is 12. i_1 indicates the number of registers among A, B, X, Y, DPR, and PS to be saved, while i_2 indicates the number of registers among DT and PG to be saved.

Note 7.

Type of register	A	B	X	Y	DPR	DT	PS
Number of cycles	3	3	3	3	4	3	3

The number of cycles corresponding to the register to be pulled are added. The number of cycles when no pulling is done is 14. i_1 indicates the number of registers among A, B, X, Y, DT, and PS to be restored, while $i_2=1$ when DPR is to be restored.

Note 8. The number of cycles is the case when the number of bytes to be transferred is even.
When the number of bytes to be transferred is odd, the number is calculated as;

$$7 + (i/2) \times 7 + 4$$

Note that, $(i/2)$ shows the integer part when i is divided by 2.

Note 9. The number of cycles is the case when the number of bytes to be transferred is even.
When the number of bytes to be transferred is odd, the number is calculated as;

$$9 + (i/2) \times 7 + 5$$

Note that, $(i/2)$ shows the integer part when i is divided by 2.

Note 10. The number of cycles is the case in the 16-bit÷8-bit operation. The number of cycles is incremented by 16 for 32-bit÷16-bit operation.

Note 11. The number of cycles is the case in the 8-bitX8-bit operation. The number of cycles is incremented by 8 for 16-bit X16-bit operation.

Note 12. When setting flag $x=0$ to handle the data as 16-bit data in the immediate addressing mode, the number of bytes increments by 1.

Note 13. When flag m is 0, the byte in the table is incremented by 1.

APPENDIX

Appendix 10. Instruction code table

Appendix 10. Instruction code table

INSTRUCTION CODE TABLE-1

D ₇ ~D ₄ Hexadecimal notation	D ₃ ~D ₀	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA		ORA	SEB	ORA	ASL	ORA	PHP	ORA	ASL		SEB	ORA	ASL	ORA
		A,(DIR,X)			A,SR	DIR,b	A,DIR	DIR	A,L(DIR)		A,IMM	A	PHD	ABS,b	A,ABS	ABS	A,ABL
0001	1	BPL	ORA	ORA	ORA	CLB	ORA	ASL	ORA	CLC	ORA	DEC		CLB	ORA	ASL	ORA
		A,(DIR,Y)	A,(DIR)	A,(SR),Y	DIR,b	A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y	A	TAS		ABS,b	A,ABS,X	ABS,X	A,ABL,X
0010	2	JSR	AND	JSR	AND	BBS	AND	ROL	AND	PLP	AND	ROL		BBS	AND	ROL	AND
		ABS	A,(DIR,X)	ABL	A,SR	DIR,b,R	A,DIR	DIR	A,L(DIR)		A,IMM	A	PLD		ABS,b,R	A,ABS	ABS
0011	3	BMI	AND	AND	AND	BBC	AND	ROL	AND	SEC	AND	INC		BBC	AND	ROL	AND
		A,(DIR),Y	A,(DIR)	A,(SR),Y	DIR,b,R	A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y	A	TSA		ABS,b,R	A,ABS,X	ABS,X	A,ABL,X
0100	4	RTI	EOR	Note 1	EOR	MVP	EOR	LSR	EOR	PHA	EOR	LSR		JMP	EOR	LSR	EOR
		A,(DIR,X)		A,SR		A,DIR	DIR	A,L(DIR)		A,IMM	A	PHG		ABS	A,ABS	ABS	A,ABL
0101	5	BVC	EOR	EOR	EOR	MVN	EOR	LSR	EOR	CLI	EOR	PHY		JMP	EOR	LSR	EOR
		A,(DIR),Y	A,(DIR)	A,(SR),Y		A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y		TAD		ABL	A,ABS,X	ABS,X	A,ABL,X
0110	6	RTS	ADC	PER	ADC	LDM	ADC	ROR	ADC	PLA	ADC	ROR		JMP	ADC	ROR	ADC
		A,(DIR,X)		A,SR	DIR	A,DIR	DIR	A,L(DIR)		A,IMM	A	RTL		(ABS)	A,ABS	ABS	A,ABL
0111	7	BVS	ADC	ADC	ADC	LDM	ADC	ROR	ADC	SEI	ADC	PLY		JMP	ADC	ROR	ADC
		A,(DIR),Y	A,(DIR)	A,(SR),Y	DIR,X	A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y		TDA		(ABS,X)	A,ABS,X	ABS,X	A,ABL,X
1000	8	BRA	STA	BRA	STA	STY	STA	STX	STA	DEY	Note 2	TXA	PHT	STY	STA	STX	STA
		A,(DIR,X)	REL	A,SR	DIR	A,DIR	DIR	A,L(DIR)						ABS	A,ABS	ABS	A,ABL
1001	9	BCC	STA	STA	STA	STY	STA	STX	STA	TYA	STA	TXS	TXY	LDM	STA	LDM	STA
		A,(DIR),Y	A,(DIR)	A,(SR),Y	DIR,X	A,DIR,X	DIR,Y	A,L(DIR),Y		A,ABS,Y				ABS	A,ABS,X	ABS,X	A,ABL,X
1010	A	LDY	LDA	LDX	LDA	LDY	LDA	LDX	LDA	TAY	LDA	TAX	PLT	LDY	LDA	LDX	LDA
		IMM	A,(DIR,X)	IMM	A,SR	DIR	A,DIR	DIR	A,L(DIR)		A,IMM			ABS	A,ABS	ABS	A,ABL
1011	B	BCS	LDA	LDA	LDA	LDY	LDA	LDX	LDA	CLV	LDA	TSX	TYX	LDY	LDA	LDX	LDA
		A,(DIR),Y	A,(DIR)	A,(SR),Y	DIR,X	A,DIR,X	DIR,Y	A,L(DIR),Y		A,ABS,Y				ABS,X	A,ABS,X	ABS,Y	A,ABL,X
1100	C	CPY	CMP	CLP	CMP	CPY	CMP	DEC	CMP	INY	CMP	DEX	WIT	CPY	CMP	DEC	CMP
		IMM	A,(DIR,X)	IMM	A,SR	DIR	A,DIR	DIR	A,L(DIR)		A,IMM			ABS	A,ABS	ABS	A,ABL
1101	D	BNE	CMP	CMP	CMP	PEI	CMP	DEC	CMP	CLM	CMP	PHX	STP	JMP	CMP	DEC	CMP
		A,(DIR),Y	A,(DIR)	A,(SR),Y		A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y				L(ABS)	A,ABS,X	ABS,X	A,ABL,X
1110	E	CPX	SBC	SEP	SBC	CPX	SBC	INC	SBC	INX	SBC	NOP	PSH	CPX	SBC	INC	SBC
		IMM	A,(DIR,X)	IMM	A,SR	DIR	A,DIR	DIR	A,L(DIR)		A,IMM			ABS	A,ABS	ABS	A,ABL
1111	F	BEQ	SBC	SBC	SBC	PEA	SBC	INC	SBC	SEM	SBC	PLX	PUL	JSR	SBC	INC	SBC
		A,(DIR),Y	A,(DIR)	A,(SR),Y		A,DIR,X	DIR,X	A,L(DIR),Y		A,ABS,Y				(ABS,X)	A,ABS,X	ABS,X	A,ABL,X

Note 1 : 42₁₆ specifies the contents of the INSTRUCTION CODE TABLE-2.
 About the second word's codes, refer to the INSTRUCTION CODE TABLE-2.
 2 : 89₁₆ specifies the contents of the INSTRUCTION CODE TABLE-3.
 About the third word's codes, refer to the INSTRUCTION CODE TABLE-2.

APPENDIX

Appendix 10. Instruction code table

INSTRUCTION CODE TABLE-2 (The first word's code of each instruction is 42₁₆)

D ₇ ~D ₄	D ₃ ~D ₀ Hexadecimal notation	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		ORA B.(DIR,X)		ORA B.SR		ORA B.DIR		ORA B.L(DIR)		ORA B.IMM	ASL B			ORA B.ABS		ORA B.ABL
0001	1		ORA B.(DIR),Y	ORA B.(DIR)	ORA B.(SR),Y		ORA B.DIR,X		ORA B.L(DIR),Y		ORA B.ABS,Y	DEC B	TBS		ORA B.ABS,X		ORA B.ABL,X
0010	2		AND B.(DIR,X)		AND B.SR		AND B.DIR		AND B.L(DIR)		AND B.IMM	ROL B			AND B.ABS		AND B.ABL
0011	3		AND B.(DIR),Y	AND B.(DIR)	AND B.(SR),Y		AND B.DIR,X		AND B.L(DIR),Y		AND B.ABS,Y	INC B	TSB		AND B.ABS,X		AND B.ABL,X
0100	4		EOR B.(DIR,X)		EOR B.SR		EOR B.DIR		EOR B.L(DIR)	PHB	EOR B.IMM	LSR B			EOR B.ABS		EOR B.ABL
0101	5		EOR B.(DIR),Y	EOR B.(DIR)	EOR B.(SR),Y		EOR B.DIR,X		EOR B.L(DIR),Y		EOR B.ABS,Y		TBD		EOR B.ABS,X		EOR B.ABL,X
0110	6		ADC B.(DIR,X)		ADC B.SR		ADC B.DIR		ADC B.L(DIR)	PLB	ADC B.IMM	ROR B			ADC B.ABS		ADC B.ABL
0111	7		ADC B.(DIR),Y	ADC B.(DIR)	ADC B.(SR),Y		ADC B.DIR,X		ADC B.L(DIR),Y		ADC B.ABS,Y		TDB		ADC B.ABS,X		ADC B.ABL,X
1000	8		STA B.(DIR,X)		STA B.SR		STA B.DIR		STA B.L(DIR)			TXB			STA B.ABS		STA B.ABL
1001	9		STA B.(DIR),Y	STA B.(DIR)	STA B.(SR),Y		STA B.DIR,X		STA B.L(DIR),Y	TYB	STA B.ABS,Y				STA B.ABS,X		STA B.ABL,X
1010	A		LDA B.(DIR,X)		LDA B.SR		LDA B.DIR		LDA B.L(DIR)	TBY	LDA B.IMM	TBX			LDA B.ABS		LDA B.ABL
1011	B		LDA B.(DIR),Y	LDA B.(DIR)	LDA B.(SR),Y		LDA B.DIR,X		LDA B.L(DIR),Y		LDA B.ABS,Y				LDA B.ABS,X		LDA B.ABL,X
1100	C		CMP B.(DIR,X)		CMP B.SR		CMP B.DIR		CMP B.L(DIR)		CMP B.IMM				CMP B.ABS		CMP B.ABL
1101	D		CMP B.(DIR),Y	CMP B.(DIR)	CMP B.(SR),Y		CMP B.DIR,X		CMP B.L(DIR),Y		CMP B.ABS,Y				CMP B.ABS,X		CMP B.ABL,X
1110	E		SBC B.(DIR,X)		SBC B.SR		SBC B.DIR		SBC B.L(DIR)		SBC B.IMM				SBC B.ABS		SBC B.ABL
1111	F		SBC B.(DIR),Y	SBC B.(DIR)	SBC B.(SR),Y		SBC B.DIR,X		SBC B.L(DIR),Y		SBC B.ABS,Y				SBC B.ABS,X		SBC B.ABL,X

APPENDIX

Appendix 10. Instruction code table

INSTRUCTION CODE TABLE-3 (The first word's code of each instruction is 89₁₆)

D ₇ ~D ₄ Hexadecimal notation	D ₃ ~D ₀	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0		MPY (DIR,X)		MPY SR		MPY DIR		MPY L(DIR)		MPY IMM				MPY ABS		MPY ABL
0001	1		MPY (DIR),Y	MPY (DIR)	MPY (SR),Y		MPY DIR,X		MPY L(DIR),Y		MPY ABS,Y				MPY ABS,X		MPY ABL,X
0010	2		DIV (DIR,X)		DIV SR		DIV DIR		DIV L(DIR)	XAB	DIV IMM				DIV ABS		DIV ABL
0011	3		DIV (DIR),Y	DIV (DIR)	DIV (SR),Y		DIV DIR,X		DIV L(DIR),Y		DIV ABS,Y				DIV ABS,X		DIV ABL,X
0100	4										RLA IMM						
0101	5																
0110	6																
0111	7																
1000	8																
1001	9																
1010	A																
1011	B																
1100	C			LDT IMM													
1101	D																
1110	E																
1111	F																

CONTACT ADDRESSES FOR FURTHER INFORMATION

JAPAN

Overseas Marketing Division
Electronics Products
Mitsubishi Electric Corporation
2-3, Marunouchi 2-chome
Chiyoda-ku, Tokyo 100, Japan
Telephone: (03) 3218-2863
Facsimile: (03) 3218-2852

Overseas Marketing Manager
Kita-Itami Works
4-1, Mizuhara, Itami-shi,
Hyogo-ken 664, Japan
Telex: 526408 KMEELCO J
Telephone: (0727) 82-5131
Facsimile: (0727) 72-2329

HONG KONG

Mitsubishi Electric (H.K.) Ltd.
41st fl., Manulife Tower, 169,
Electric Road, North Point, Hong Kong
Telex: 60800 MELCO HX
Telephone: 510-0555
Facsimile: 510-9830, 510-9822,
510-9803

SINGAPORE

MELCO SALES SINGAPORE PTE,
LTD.
307 Alexandra Road # 05-01/02
Mitsubishi Electric Building
Singapore 0315
Telex: RS 20845 MELCO
Telephone: 4732308
Facsimile: 4738944

TAIWAN

MELCO-TAIWAN CO., Ltd.
1st fl., Chung-Ling Bldg.,
363, Sec. 2, Fu-Hsing S Road,
Taipei R.O.C.
Telephone: (02) 735-3030
Facsimile: (02) 735-6771
Telex: 25433 CHURYO "MELCO-
TAIWAN"

U.S.A.

NORTHWEST

Mitsubishi Electronics America, Inc.
1050 East Arques Avenue
Sunnyvale, CA 94086
Telephone: (408) 730-5900
Facsimile: (408) 730-4972

SAN DIEGO

Mitsubishi Electronics America, Inc.
16980 Via Tazon, Suite 220
San Diego, CA 92128
Telephone: (619) 451-9618
Facsimile: (619) 592-0242

DENVER

Mitsubishi Electronics America, Inc.
4600 South Ulster Street
Metropoint Building, 7th Floor
Denver, CO 80237
Telephone: (303) 740-6775
Facsimile: (303) 694-0613

SOUTHWEST

Mitsubishi Electronics America, Inc.
991 Knox Street
Torrance, CA 90502
Telephone: (213) 515-3993
Facsimile: (213) 217-5781

SOUTH CENTRAL

Mitsubishi Electronics America, Inc.
1501 Luna Road, Suite 124
Carrollton, TX 75006
Telephone: (214) 484-1919
Facsimile: (214) 243-0207

NORTHERN

Mitsubishi Electronics America, Inc.
15612 Highway 7 # 243
Minnetonka, MN 55345
Telephone: (612) 938-7779
Facsimile: (612) 938-5125

NORTH CENTRAL

Mitsubishi Electronics America, Inc.
800 N. Bierman Circle
Mt. Prospect, IL 60056
Telephone: (312) 298-9223
Facsimile: (312) 298-0567

NORTHEAST

Mitsubishi Electronics America, Inc.
200 Unicorn Park Drive
Woburn, MA 01801
Telephone: (617) 932-5700
Facsimile: (617) 938-1075

MID-ATLANTIC

Mitsubishi Electronics America, Inc.
800 Cottontail Lane
Somerset, NJ 08873
Telephone: (201) 469-8833
Facsimile: (201) 469-1909

SOUTH ATLANTIC

Mitsubishi Electronics America, Inc.
2500 Gateway Center Blvd., Suite 300
Morrisville, NC 27560
Telephone: (404) 368-4850
Facsimile: (404) 662-5208

SOUTHEAST

Mitsubishi Electronics America, Inc.
Town Executive Center
6100 Glades Road #210
Boca Raton, FL 33433
Telephone: (407) 487-7747
Facsimile: (407) 487-2046

CANADA

Mitsubishi Electronics America, Inc.
6185 Ordan Drive, Unit #110
Mississauga, Ontario, Canada L5T 2E1
Telephone: (416) 670-8711
Facsimile: (416) 670-8715

Mitsubishi Electronic Sales Canada,
Inc.
340 March Road, Suite 502
Kanata, Ontario, Canada K2K 2E4
Telephone: (613) 591-3348
Facsimile: (613) 591-3948

GERMANY

Mitsubishi Electric Europe GmbH
Headquarters
Gothaer Str. 8
4030 Ratingen 1, Germany
Telephone: 2102-486-0
Facsimile: 2102-486-367

Mitsubishi Electric Europe GmbH
Munich Office
Fraunhoferstr. 9
8045 Ismaning, Germany
Telephone: 89-96 07 94 30
Facsimile: 89-96 07 94 11

Mitsubishi Electric Europe GmbH
Stuttgart Office
Zettachring 12
7000 Stuttgart 80, Germany
Telephone: 711-728 74 70
Facsimile: 711-72 47 21

Mitsubishi Electric Europe GmbH
Frankfurt Office
(Power Semiconductors)
Promenadestr. 16
W. 6140 Bensheim
Telephone: (06251)62097/98
Facsimile: (06251)62099

FRANCE

Mitsubishi Electric France S. A.
55, Avenue de Colmar
92563 Rueil Malmaison Cedex, France
Telephone: 1-47. 08. 78. 00
Facsimile: 1-47. 51. 36. 22

ITALY

Mitsubishi Electric Europe GmbH
Milano Branch Office
Centro Direzionale Colleoni
Palazzo Perseo 2
20041 Agrate Brianza
Milano, Italy
Telephone: 39-605 31
Facsimile: 39-605 32 12

SWEDEN

Mitsubishi Electric Europe GmbH
Stockholm Office
Lastbilsvägen 6b
19149 Sollentuna, Sweden
Telephone: 8-96 04 60
Facsimile: 8-92 76 97

U.K.

Mitsubishi Electric (U.K.) Ltd.
Travellers Lane
Hatfield
Herts AL10 8XB, England, U.K.
Telephone: 707-27 61 00
Facsimile: 707-27 86 92

AUSTRALIA

Mitsubishi Electric Australia Pty. Ltd.
348 Victoria Road
Rydalmere Nsw 2116, Australia
Private Bag No.2 Rydalmere Nsw 2116
Telex: MESYDAA 126614
Telephone: (02) 684-7200
Facsimile: (02) 638-7072

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL M37732/M37730 Group

Dec. First Edition 1992

Edited by

Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by

Mitsubishi Electric Corp., Semiconductor Marketing Division

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.

©1992 **MITSUBISHI ELECTRIC CORPORATION**

MITSUBISHI SEMICONDUCTORS

M37732 Group

M37730 Group



MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLDG., MARUNOUCHI, TOKYO 100 TELEX: J24532 CABLE: MELCO TOKYO

If these products or technologies
fall under Japanese and/or COCOM
strategic restrictions, diversion
contrary thereto is prohibited.