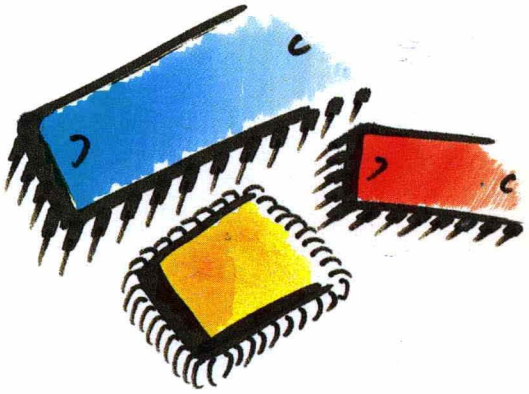
1988 Standard Products Data Book

NCR

1988 Standard Products Data Book

Microelectronics Division



Creating value

NCR STANDARD PRODUCTS DATABOOK
APRIL 1988

INDEX

OVERVIEW

NCR: A HISTORY OF QUALITY, SERVICE AND SUPPORT

The NCR Corporation was founded over 100 years ago with the invention of the cash register. Since then, its been one innovation after another. For example, as an early pioneer in electronic component technology, NCR performed advanced vacuum tube research as early as 1938. Later, in 1959, NCR produced the first commercial all solid-state computer.

Today, NCR Corporation is a multi-billion dollar manufacturer of diversified business and technology products including computer systems, retail and banking terminals, and semiconductors.

NCR established the microelectronics division in 1971 to design and manufacture advanced integrated circuits for internal use. In 1981, NCR Microelectronics became an autonomous division and began selling its services to the merchant market.

Just three years later in 1984, NCR achieved independent recognition as the industry's leading supplier of standard cell products—the fastest growing segment of the exploding ASIC market. This achievement not only made NCR one of the world's fastest growing ASIC suppliers, but assured its customers an early lead in getting their ASIC-based products to market.

In 1987, NCR introduced a technology breakthrough—the Design Advisor™. This first-of-its-kind application of artificial intelligence, assists design engineers by providing 1000 years of engineering knowledge. This product introduction, announced at New York City's Hayden Planetarium in unison with the Design Automation Conference (DAC), received very significant trade press and analyst coverage.

Today, NCR's Microelectronics Division consists of two modern manufacturing facilities located in Fort Collins and Colorado Springs, Colorado. In early 1988, the Miamisburg, Ohio, microelectronics center's operations were relocated in these two facilities.

These independent, but closely coupled plants, offer customers a new level of security of supply.

Fort Collins, Colorado

The Fort Collins facility is headquarters for NCR's commercial/industrial Application Specific Integrated Circuits (ASIC), Digital Signal Processing (DSP), Software Development, and Customer Owned Tooling (COT) business units. The Fort Collins ASIC business unit conducts a great deal of research and development in gate arrays, standard cell libraries, and "supercells," as well as customer owned tooling. The recently formed software development business unit is dedicated to providing leading edge software products.

The Fort Collins facility also houses volume wafer fabrication and test facilities and a complete small-volume/fast-turn assembly capability for popular plastic and ceramic devices.

Colorado Springs, Colorado

Colorado Springs is home for NCR's logic product, Military ASIC, and Automotive ASIC business units.

The logic products group supplies a variety of microprocessors, logic support devices, and special function chips. The military products group is dedicated to supplying both standard ICs and ASIC devices screened to military requirements. It offers a comprehensive quality system based on the MIL-Q-9858A. Devices can be screened and qualified to MIL-STD-883C, Method 5004/5005 and 5010 on a stand-alone basis. The automotive ASIC group supplies both standard and ASIC devices to the special requirements of the automotive industry.

The Colorado Springs facility is the site of NCR's newest advanced wafer fab. Currently, this fab is running geometies down to 1.5μm with 1.0μm in prototype development.

This *Standard Products Databook* highlights NCR's standard product offering. The complementary *ASIC Databook* (May 1988) includes ASIC product information for all applications.

If you desire a copy of the *ASIC Databook,* please contact the sales office in your area (see page 13) or call NCR's hotline: **1-800-334-5454.**

STRATEGY AND GOALS

SCSI Products

NCR is creating high value products for its customers in the SCSI Controller area. As the SCSI leader, NCR offers the broadest line of SCSI products ranging from the low cost 5380 family of chips to the high performance 53C90. SCSI products are designed using the semi-custom modular approach while maintaining sensitivity to the industry trend toward higher integration. An emphasis is placed on software compatibility with the 5380 and the 53C90 to provide customers a migration path to higher performance products. NCR's high volume marketing approach allows for an optimum price/performance ratio giving customers high value products.

Some of NCR's newest developments in the SCSI area include the 53C90 - a high performance SCSI processor which features sequenced commands and high speed synchronous data transfers, the 53C80-40 - a CMOS part pin compatible with the 5380 that also features a faster REQ-ACK SCSI handshake, and the 53C400 - a single chip host adapter chip compatible with the IBM PC/XT/AT Bus and the PS/2 Model 30 which also features two 128 byte buffers to provide speed matching.

NCR's SCSI Products group is continuing to provide customers with high perfromance, low cost quality products by using the latest process technology, Computer Aided Design tools, and quality assurance programs. NCR maintains a high degree of customer satisfaction by filling the market needs with performance products quickly and effectively.

Graphics Products

As users become more comfortable with personal computers, there will be an increase in demand for color monitors. Likewise, computer manufacturers will demand high performance color/graphics controllers for these monitors. NCR's graphics products provide a solution now. Our graphics products provide easy-to-use, highly integrated and cost effective solutions for the needs of the PC market.

NCR's foremost strategy is to provide industry standard solutions to the marketplace which preserve compatibility, while providing leadership, performance, features and integration. With this strategy, NCR has become a popular source of CRT controllers for both monochrome and color display systems.

NCR Microelectronics will compete only at the chip level in the marketplace. This allows us to concentrate on the production of CRT controller chips rather than the production of PC board-based display systems. With this strategy, NCR can provide the best product, service and commitment to our customers' needs.

Providing timely and current solutions for the needs of the display systems market is our strategy for the future. This will continue to maintain NCR's position as an industry leader.

Communication Products

The Communications Business Unit develops semiconductor circuits for use in local area networks and digital telephony. NCR develops local area digital communication network circuits, these circuits are for networks usually less than 50,000 meters and connecting up to 1024 nodes. Local area network speeds range from 1200 bits per second to greater than 100 MB/S. These circuits support 802 Standards, PC Networks, Factory Automation Networks, Office Automation Networks, CAE and CAD Networks and Backbone Networks.

Digital telephony circuits include Digital Telecom as well as Integrated Services Digital Networks (ISDN). These circuits support digital applications on existing twisted pair wiring circuits as well as fiber optic networks. They include ISDN interfaces, PBX's, and central office applications.

NCR Memory Products

NCR Memory products group has moved from Miamisburg, Ohio to Fort Collins, Colorado and is a part of the Customer Owned Tooling (COT) group.

NCR offers a full line of high performance CMOS and NMOS read-only memories (ROM) with a variety of pinouts, access times, and packages (DIP, PLCC, Fpp). NCR's strong manufacturing capability allows for fast turn around of both prototype and production quantities. NCR provides the customer service levels and the engineering support required of a major supplier of ROM's in today's market.

NCR Military Products

NCR Microelectronics Military group, now located in Colorado Springs, Colorado, is dedicated to providing both standard and ASIC products to the military market.

It offers a comprehensive quality system based on the MIL-Q-9858A. Devices can be screened and qualified to MIL-STD-883C, Method 5004/5005 and 5010 on a stand-alone basis.

The NCR Microelectronics Military group is expanding its capabilities and expects several new standard and ASIC products to be included in its offerings this year.

NCR QUALITY

NCR Microelectronics Division — A History of Quality

The NCR Corporation was founded over 100 years ago with the invention of the cash register. Since then it has been one innovation after the other. Today NCR Corporation is a multi-billion dollar manufacturer of diversified business and technology products including computer systems, peripherals, retail and banking terminals and semiconductors.

NCR established its first microelectronics laboratory in 1963 to stay abreast of the emerging semiconductor technology. The laboratory was expanded in 1966, and by 1968 NCR's first production MOS circuits were produced. NCR created the Microelectronics Division in 1971 with the opening of its first manufacturing plant located in Miamisburg, Ohio to manufacture advanced integrated circuits for internal use. NCR expanded its Microelectronics operation in 1975 with a second production facility in Colorado Springs, Colorado and in 1979 added a third facility in Fort Collins, Colorado.

In 1981 NCR Microelectronics became an autonomous division and began selling its products to the merchant market. And by 1984 the division had become the market leader in the standard cell sector of the semicustom application specific ICs (ASIC) market serving customers with commercial, industrial, automotive and military applications.

In addition to standard cell devices NCR Microelectronics also manufactures an extensive line of state-of-the-art VLSI devices including gate arrays, peripheral ICs, logic devices, digital signal processing devices, microprocessors, and memories.

A Commitment to Quality

"NCR Corporation is totally committed to maintaining a reputation for excellence second to none in the markets we serve. We intend to maintain this reputation by providing only defect-free products and error-free services."

Charles E. Exley Jr.
Chairman of the Board and President
NCR Corporation

Mr. Exley's statement, from our highest level of management, exemplifies the commitment to quality, reliability and customer satisfaction NCR has held for over 100 years.

The NCR goal in all products is to exceed the customer's quality and reliability requirements by building quality in. Of course, we test each device extensively, but testing alone cannot insure long-term reliability.

Each of NCR Microelectronics standard products, cell libraries and processes has been extensively characterized and qualified. Our quality assurance engineers work closely with standard product, standard cell and gate array IC designers, as well as, computer-aided design software engineers and process engineers to assure that long-term product quality is an integral part of every NCR standard or ASIC device.

At NCR we have learned that not only must each and every element of the design and manufacturing process be fully validated, but that they must also be validated as an entire system—from processing to workstation simulations. For example, in the design process, strict design rules with carefully controlled margining requirements are used to insure products meet or exceed customer quality and reliability requirements. In the fabrication and assembly operation, Statistical Process Control (SPC) is the key to maintaining a high level of confidence.

Of course each ASIC device receives full screening, testing and qualification prior to shipment, but without built-in quality attributes, long term reliability could not be ensured. And at NCR quality is not measured in percent of failed devices received by our customers, but rather the reliability of the devices over the long haul.

Additionally, NCR supports a program of continuous quality improvement. This includes working closely with customers to help insure that applications are consistent with the intended use of the product. When problems are encountered, NCR maintains a closed loop corrective action system to insure that problems are resolved and corrected in a timely manner.

All of this attention to quality and reliability has paid off—for NCR and our customers. As a result of our exceedingly high standards, NCR Microelectronics has an AQL level much better than the industry average.

Quality is a System

Behind NCR's reputation for excellence in quality and reliability is a comprehensive quality system based upon MIL-Q-9858A. This system provides detailed documentation of our design and manufacturing processes as well as for the handling, testing and inspections of the raw material and work-in-process. It also documents manufacturing operation controls, final acceptance and delivery. Here are some of the key elements of that system.

• Defect Prevention

NCR Microelectronics is committed to the philosophy that defect prevention is the most effective means of assuring the quality of all products and services. As a result of this commitment, defect prevention is practiced throughout every phase of the operation.

Quality is designed into our products through the use of documented design rules and procedures. Comprehensive design reviews are conducted to ensure the product design conforms to specifications. Processing is accomplished under documented controlled conditions thereby providing the highest quality throughout the fabrication process. Extensive characterization and reliability verification testing can be performed to help ensure that products are defect-free and perform reliably over an extended period of time.

• Closed Loop Corrective Action

A Closed Loop Corrective Action (CLCA) system is in place at NCR Microelectronics to ensure that problems are resolved in a timely manner. The basic elements of the system include:

- . . . Problem identification
- . . . Investigation and determination of cause
- . . . Corrective action to eliminate the problem and prevent its recurrence
- . . . Verification that the corrective action has eliminated the problem

• Configuration Control

Configuration control procedures ensure that product designs are fully documented and production changes are implemented in a controlled manner. With this procedure, customers are assured of complete configuration accountability of the end item. Engineering documentation is handled under strict change control procedure utilizing request for change and engineering change notice procedures. Engineering changes are documented using part number and revision letter control. Change implementation is monitored, tracked and recorded to ensure that changes are incorporated into production as scheduled. Configuration accountability records are maintained to allow complete design traceability of the completed end item.

• Calibration of Equipment

The procedures used by NCR for calibrating manufacturing and testing equipment are based upon MIL-C-45662, Calibration System Requirements. These procedures are designed to ensure that all measurements are accurate and reliable. Calibration of NCR's manufacturing and testing equipment is performed in accordance with the manufacturer's recommendations or those proven by the National Bureau of Standards. Calibrations are performed in-house or by a certified laboratory depending upon the type and complexity of the equipment. NCR's calibration system includes machine labeling and record documentation system designed to ensure that calibrations are performed on schedule.

• Supplier Quality Management

NCR is committed to doing business with selected suppliers who have demonstrated their ability to constantly deliver defect-free products as scheduled. Suppliers are selected by NCR based upon evaluation and audits of their quality systems and their capabilities for meeting the specified quality requirements. NCR's supplier evaluation criteria include:

- . . . Commitment to quality
- . . . Proven ability to produce defect-free product
- . . . Well documented processes that are under strict control
- . . . Use of statistics to evaluate process capabilities
- . . . Engineering expertise and design maturity
- . . . Engineering change control procedures
- . . . Test documentation, inspection records, certification of performance, quality reports on a regular basis
- . . . Corrective action system with ability to resolve problems in a timely manner
- . . . Continuous quality improvement program
- . . . Flow down procedures for second source suppliers

• Statistical Process Control

NCR uses Statistical Process Control (SPC) as one of the key elements in their systems to build in quality and reliability. For example, within wafer fabrication, critical process parameters are charted to monitor the on-going statistical performance of the process. The targeted performance standard of each monitored process is continually raised resulting in ever improving performance. Numerous other process parameters are monitored throughout the manufacturing, qualification and support process to assure conformance to documented requirements.

• Wafer Fabrication

After the device is approved for production, stringent process and assembly controls are used to build in quality and reliability. Within wafer fabrication, over 150 critical process parameters are monitored to assure performance of the process. The following quality assurance functions are just some of the functions performed during wafer fabrication:

- . . . After Develop Inspections (ADIs) performed at each masking level
- . . . After Etch Inspection (AEIs) performed at each masking level
- . . . All critical process operations implemented on statistical process control (SPC) chart
- . . . In-process QC audits and final QC wafer lot acceptance gate
- . . . CV sampling performed for every lot
- . . . Profile testing of critical parameters
- . . . 100% die probe
- . . . QC lot acceptance for assembly

Quality and Reliability Testing Procedures

NCR offers complete quality assurance services to provide the optimum high reliability solution to meet your system requirements. At NCR, end items are thoroughly tested to ensure conformance to specified requirements. For military and high reliability applications, NCR supports the stringent MIL-STD-883C, Method 5004/5005 and Method 5010 screening and quality conformance inspection test methods and on a stand-alone basis. These testing procedures are designed to insure that the quality and reliability of the completed device conforms with the applicable procurement document. Additionally, NCR can support customer requirements for special screening and qualification testing procedures.

Methods 5004 and 5005 are companion test methods establishing total lot screening and quality conformance inspection procedures for microcircuits. Method 5010 is a relatively new all inclusive screening and qualification procedure for custom and semicustom microcircuits with low production volume requirements.

Using Method 5010 as an example, the testing and evaluation procedure involves four major areas:

. . . **Element Evaluation**—This series of tests involve the materials for device assembly, i.e. wafers, die and packages.

. . . **Process Control**—These tests are performed to ensure the integrity of the die attachment and wire bonding process.

. . . **100% Device Screening**—This series of rigorous tests are performed on assembled devices to ensure the quality and reliability levels specified by the device specification has been achieved.

. . . **Quality Conformance Inspection**—This group of tests establishes qualification and quality conformance inspection procedures for initial device qualification, requalification and retention of qualification. These quality conformance tests consist of Group A, B, C and D.

- Group A—Electrical tests performed on an inspection lot that has already passed 100% screening.

- Group B—Environmental tests designed to test the integrity of the fabrication and assembly processes.

- Group C—Die related test conducted on a three month periodic basis, as required.

- Group D—Package related tests conducted on a six month periodic basis, as required.

• High Rel ASIC

These products are available in commercial, industrial, automotive and military temperature ranges and a variety of screening procedures including MIL-STD-883C Method 5004/5005 and Method 5010, on a stand alone basis.

NCR's ASIC CMOS product family is comprised of both standard cells and gate arrays all screened for operation in the automotive temperature ranges and military temperature range. Standard cells are offered in 2μm double-level-metal (DLM) and 3μm single-level-metal (SLM). Gate arrays are provided in 25m double-level-metal with complexity ranging from 600 to 8,500 usable gates.

NCR's automotive products group is based in Colorado Springs, Colorado. This plant is NCR's newest microelectronics factory and features an automotive design center, 1.5-micron fabrication capability, testing, and qualification. The automotive products group supplies full custom, cell based and gate array products to the automotive and high-temp industrial markets.

The Miamisburg plant is home to the Microelectronics Division's Military Products Unit. The plant offers a military applications design center, on-site fabrication, and back-end screening, testing and qualification to a variety of Mil standards. The Military Products Group is dedicated to supplying both standard and ASIC devices screened to military requirements.

NCR's CMOS devices, using our semicustom design approach, have the characteristics essential to military and automotive systems including high reliability in rugged environments, low power consumption, and high immunity to noise and power supply variations.

A Commitment to Customer Satisfaction

NCR's commitment to quality, reliability and customer satisfaction is an integral part of corporate policy and philosophy. And it's been that way since 1884 when NCR's founder, John H. Patterson, set the standards that NCR follows today.

Patterson's premise was simple. Build each product the best it could be built and provide a level of service to customers which exceeds their expectations. Patterson knew that if each employee believed and lived that premise, success for his newly founded company would be assured. Today, as then, everyone at NCR is committed to continue its successful growth in the markets we serve by striving to provide defect-free products and error-free services to each of our customers.

Every NCR employee is motivated to ensure our customer's satisfaction in our products and services. It's this commitment to customer service that forms the underlying foundation on which our Quality Assurance Program is based. At NCR we feel this is a fundamental part of quality. Testing for quality is an important function, but we believe that the long-term reliability cannot be tested-in. Reliability begins and ends with people. People committed to product and service excellence.

At NCR, when we say, "We Take Customer Satisfaction Personally," - we mean it.

NCR PLANTS/PRODUCT LINES

FORT COLLINS

Commercial ASIC Products
Digital Signal Processing Products
Customer Owned Tooling/Memory Products
Telecommunications Products
Software Development

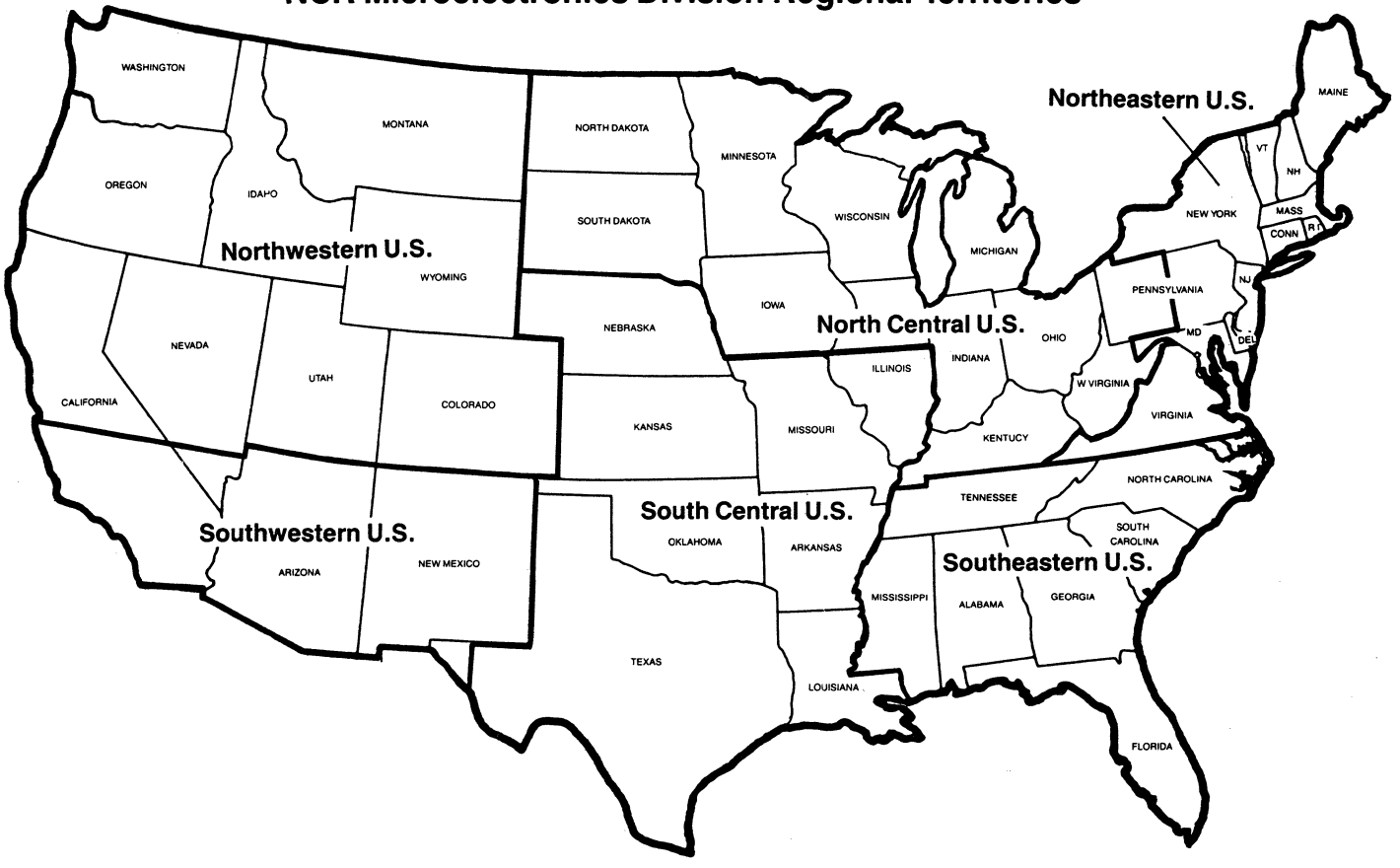
2001 Danfield Court
Fort Collins, CO 80525
(303) 226-9500

COLORADO SPRINGS

SCSI Products
Graphics Products
Military ASIC Products
Automotive ASIC Products
Distribution
Internal Marketing

1635 Aeroplaza Drive
Colorado Springs, CO 80916
(303) 596-5611 or
(800) 525-2252

NCR Microelectronics Division Regional Territories



NORTHEASTERN AREA SALES OFFICE

NCR Microelectronics Division
400 W. Cummings Park
Suite 2750
Woburn, MA 01801
Phone: (617) 933-0778

SOUTHEASTERN OFFICE

NCR Microelectronics Division
700 Old Roswell Lakes Pkwy.
Suite 250
Roswell, GA 30076
Phone: (404) 587-3736

NORTH CENTRAL OFFICE

NCR Microelectronics Division
33 West Higgins Road
South Barrington, IL 60010
Phone: (312) 426-4600

SOUTH CENTRAL OFFICE

NCR Microelectronics Division
400 Chisholm Place, Suite 100
Plano, TX 75075
Phone: (214) 578-9113

NORTHWESTERN AREA SALES OFFICE

NCR Microelectronics Division
3130 De la Cruz Blvd.
Suite 209
Santa Clara, CA 95054-2410
Phone: (408) 727-6575

SOUTHWESTERN REGIONAL SALES OFFICE

NCR Microelectronics Division
1940 Century Park East
Los Angeles, CA 90067
Phone: (213) 556-5231

NCR SALES REPS AND DESIGN CENTERS

Cross Referenced by State/International Area
February 15, 1988

ALABAMA
Electronic Manufacturers Agents
309 Jordan Lane
Huntsville, AL 35805
205-830-4030

ARIZONA
Sun State Technical, Inc.
Suite 115
2323 East Magnolia
Phoenix, AZ 85034
602-220-0595

ARKANSAS
ION Associates, Inc.
1504 109th Street
Grand Prairie, TX 75050
214-647-8225

CALIFORNIA
(North)
Quorum Technical Sales
Building 12
4701 Patrick Henry Drive
Santa Clara, CA 95054
408-980-0812

(South)
Leading Concept Technologies, Inc.
3900 Birch Street, Suite 104
Newport Beach, CA 92660
714-851-0654

(San Diego and Imperial Counties)
Earle Associates, Inc.
Suite 200
7585 Ronson Road
San Diego, CA 92111
619-278-5441

COLORADO
Electrodyne
Suite 110
2620 South Parker Road
Aurora, CO 80014
303-695-8903

CONNECTICUT
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

DELAWARE
OPEN

DISTRICT OF COLUMBIA
OPEN

FLORIDA
Delmac Sales, Inc.
1701 West Hillsboro Boulevard
Suite 304
Deerfield, FL 33441
305-427-7788

Delmac Sales, Inc.
1963 Corporate Square Drive
Suite 106
Longwood, FL 32750
305-831-0040

Delmac Sales, Inc.
1170 Woodlawn Street
Clearwater, FL 34616
813-443-2622

GEORGIA
Electronic Manufacturers Agents
620 Colonial Park Drive
Roswell, GA 30075
404-992-7240

IDAHO
Electrodyne
Suite 120
825 East 4800 South Street
Salt Lake City, UT 84107
801-264-8050

ILLINOIS
(South of I-74)
Dy-Tronix, Inc.
3407 Bridgeland Drive
Bridgeton, MO 63044
314-291-4777

ILLINOIS (Continued)
(North of I-74)
Eagle Technical Sales
1805-B Hicks Road
Rolling Meadows, IL 60008
312-991-0700

(Rock Island County)
Aldridge Associates, Inc.
7138 Shady Oak Road
Eden Prairie, MN 55344-3517
612-944-8433

INDIANA
Bailey's Electronics Sales
87 Bali Hai
Carmel, IN 46032
317-846-5578

IOWA
Aldridge Associates, Inc.
7138 Shady Oak Road
Eden Prairie, MN 55344-3517

KANSAS
Dy-Tronix, Inc.
Suite 322
1999 Amidon
Wichita, KS 67203-2124
316-838-0884

Dy-Tronix, Inc.
Suite 106
5001 College Boulevard
Leawood, KS 66211
913-339-6333

KENTUCKY
(W of I-75)
Bailey's Electronics Sales
87 Bali Hai
Carmel, IN 46032
317-846-5578

(E of I-75)
Arthur Baier Company
7480 Kingswood Drive
Westchester, OH 45069
513-779-2395

LOUISIANA
ION Associates, Inc.
1504 109th Street
Grand Prairie, TX 75050
214-647-8225

MAINE
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

MARYLAND
OPEN

MASSACHUSETTS
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

MICHIGAN
Rathsburg Associates, Inc.
17600 Northland Park Court
Suite 100, P.O. Box 5037
Southfield, MI 48086-5037
313-559-9700

MINNESOTA
Aldridge Associates, Inc.
7138 Shady Oak Road
Eden Prairie, MN 55344-3517

MISSISSIPPI
Electronic Manufacturers Agents
309 Jordan Lane
Huntsville, AL 35805
205-830-4030

MISSOURI
Dy-Tronix, Inc.
3407 Bridgeland Drive
Bridgeton, MO 63044
314-291-4777

MONTANA
Electrodyne
Suite 120
825 East 4800 South Street
Salt Lake City, UT 84107
801-264-8065

NEBRASKA
Dy-Tronix, Inc.
Suite 322
1999 Amidon
Wichita, KS 67203-2124
316-838-0884

NEVADA
(All, excluding)
Quorum Technical Sales
4701 Patrick Henry Drive, Bldg. 12
Santa Clara, CA 95054
408-980-0812

(Clark County)
Sun State Technical, Inc.
Suite 115
2323 East Magnolia
Phoenix, AZ 85034
602-220-0595

NEW HAMPSHIRE
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

NEW JERSEY
(North)
Comtronic Associates, Inc.
555 Braod Hollow Road
Melville, NY 11747
516-249-0505

(South)
TCA, Inc.
1570 McDaniel Drive
West Chester, PA 19380
215-692-6853

NEW MEXICO
Nelco Electronix
4801 General Bradley, N.E.
Albuquerque, NM 87111
505-293-1399

NEW YORK
(All except L.I.)
Ontec Electronic Marketing
167 Flanders Street
Rochester, NY 14619
716-464-8636

NEW YORK (Continued)
Ontec Electronic Marketing
16 Gabriella Road
P.O. Box 525
Wappinger Falls, NY 12590
914-462-7374

(Long Island)
Comtronic Associates, Inc.
555 Broad Hollow Road
Melville, NY 11747
516-249-0505

NORTH CAROLINA
Electronics Manufacturers Agents
Suite 204
6604 Six Forks Road
Raleigh, NC 27609
919-846-6888

NORTH DAKOTA
Aldridge Associates, Inc.
7138 Shady Oak Road
Eden Prairie, MN 55344-3517

OHIO
Arthur Baier Company
Suite 106
6690 Beta Drive
Cleveland, OH 44143
216-461-6161

Arthur Baier Company
4940 Profit Way
Dayton, OH 45414
513-276-4128

Arthur Baier Company
7395 Mapleleaf Boulevard
Worthington, OH 43085
614-764-1144

Arthur Baier Company
7480 Kingswood Drive
Westchester, OH 45069
513-779-2395

OKLAHOMA
ION Associates, Inc.
1504 109th Street
Grand Prairie, TX 75050
214-647-8225

OREGON
Westerberg & Associates
7165 SW Fir Loop
Portland, OR 97224
503-620-1931

PENNSYVLANIA
(West)
Arthur Baier Company
Suite 106
6690 Beta Drive
Cleveland, OH 44143
216-461-6161

(East)
TCA, Inc.
1570 McDaniel Drive
West Chester, PA 19380
215-692-6853

RHODE ISLAND
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

SOUTH CAROLINA
Electronics Manufacturers Agents
309 Jordan Lane
Huntsville, AL 35805
205-830-1947

SOUTH DAKOTA
Aldridge Associates, Inc.
7138 Shady Oak Road
Eden Prairie, MN 55344-3517
612-944-8433

TENNESSEE
East (Eastern Time Zone)
Electronic Manufacturers Agents
620 Colonial Park Drive
Roswell, GA 30075
404-992-7240

West (Central Time Zone)
Electronic Manufacturers Agents
309 Jordan Lane
Huntsville, AL 35805
205-830-4030

TEXAS
(All, excluding)
ION Associates, Inc.
1504 109th Street
Grand Prairie, TX 75050
214-647-8225

ION Associates, Inc.
Suite #100, Bldg. A
12731 Research Boulevard
Austin, TX 78759
512-331-7251

ION Associates, Inc.
14347-A Torrey Chase Boulevard
Houston, TX 77014-1696
713-537-7717

(El Paso County)
Nelco Electronix
4801 General Bradley, N.E.
Albuquerque, NM 87111
505-293-1399

UTAH
Electrodyne
Suite 120
825 E. 4800 South Street
Salt Lake City, UT 84107
801-264-8050

VERMONT
John E. Boeing Co., Inc.
139 Billerica Road
Chelmsford, MA 01824-3501
617-256-5800

VIRGINIA
OPEN

WASHINGTON
Westerberg & Associates
12505 Bell-Red Road, Suite 112
Bellevue, WA 98005
206-453-8881

WEST VIRGINIA
Arthur Baier Company
Suite 106
6690 Beta Drive
Cleveland, OH 44143
513-461-6161

WISCONSIN
Eagle Technical Sales
1805-B Hicks Road
Rolling Meadows, IL 60008
312-991-0700

WYOMING
Electrodyne
Suite 110
2620 South Parker Road
Aurora, CO 80014-1660
303-695-8903

CANADA, excluding
Cantec Representative, Inc.
1573 Laperriere Avenue
Kttawa, Ontario, CANADA
K1Z 7T3
613-725-3704

Cantec Representatives, Inc.
8 Strathearn Avenue - Unit #18
Brampton, Ontario, CANADA
L6T 4L8
416-791-5922

Cantec Representatives, Inc.
3639 Sources Road, Suite 116
Dollard des Ormeaux
Quebec, CANADA H9B 2K4
514-683-6131

(Alberta & Br. Columbia)
OPEN

CUIDAD, MEXICO (only)
Nelco Electronix
4801 General Bradley, N.E.
Albuquerque, NM 87111
505/293-1399

UNITED KINGDOM AND CENTRAL EUROPE
Manhattan Skyline, Ltd.
Manhattan House, Bridge Road
Maidenhead, Berkshire
England, S16 8DB
44-628/75851

UNITED KINGDOM AND CENTRAL EUROPE (Continued)
Manhattan Skyline Halbleiter und Computerperipherie Vertriebs GmbH
Weisbadener Str. 5/5a
D-6204 Taunusstein-Hahn
West Germany
49-6128/23044

ITALY
Skylab s.r.l.
Electronic Components
Piazza Carbonari 12
20125 Milano
Italy
011/39-2-688-3806

FRANCE
Futur IDS
26, Rue de Versailles
78150 Le Chesnay
France
011/39-63-2626

JAPAN
Japan Macnics Corporation
516 Imaimanami-Cho
Kawasaki City, Kanagawa
211 Japan
011/81-44-711-0022

HONG KONG
Printed Circuits Int'l (HK), Ltd.
3rd Floor, Kin Yip Factory Bldg.
9 Cheung Yee Street
Cheung Sha Wan
Kowloon, Hong Kong
011/85-3743-1366

KOREA
Dongah Trading Corporation
7F, Dongah Building
14--2 Yoido-Dong, Youngdeungpo-GU
C.P.O. Box 1815
Seoul, Korea
011/82-2-784-8312

SINGAPORE
Datasource Electronics Private Ltd.
1296 Toa Payoh Lor 1
#04-01 Siong Hoe Int'l Building
Singapore 1231
011/65-258-2752 or 5066

TAIWAN
Prospect Technology Corporation
5, Lane 55, Long-Chiang Road
Taipei, Taiwan
011/886-2-721-9533

AUSTRAILIA, NEW ZEALAND
Energy Control Pty Ltd.
26 Boron Street
Sumner Park Qld. 4074
P.O. Box 6502, Goodna Qld. 4300
Brisbane, Australia
011/61-7-376-2955

53C90 ENHANCED SCSI PROCESSOR

Introduction

General Description

The NCR 53C90 Enhanced SCSI Processor (ESP) is a high performance CMOS device which implements the ANSI X3.131-1986 SCSI Standard. The 53C90 operates in both the initiator and target roles and can therefore be used in host adapter and control unit applications.

The 53C90 performs such functions as bus arbitration, selection of a target or reselection of an initiator. It handles message, command, status and data transfer between the SCSI bus and its 16 byte internal FIFO or a buffer memory.

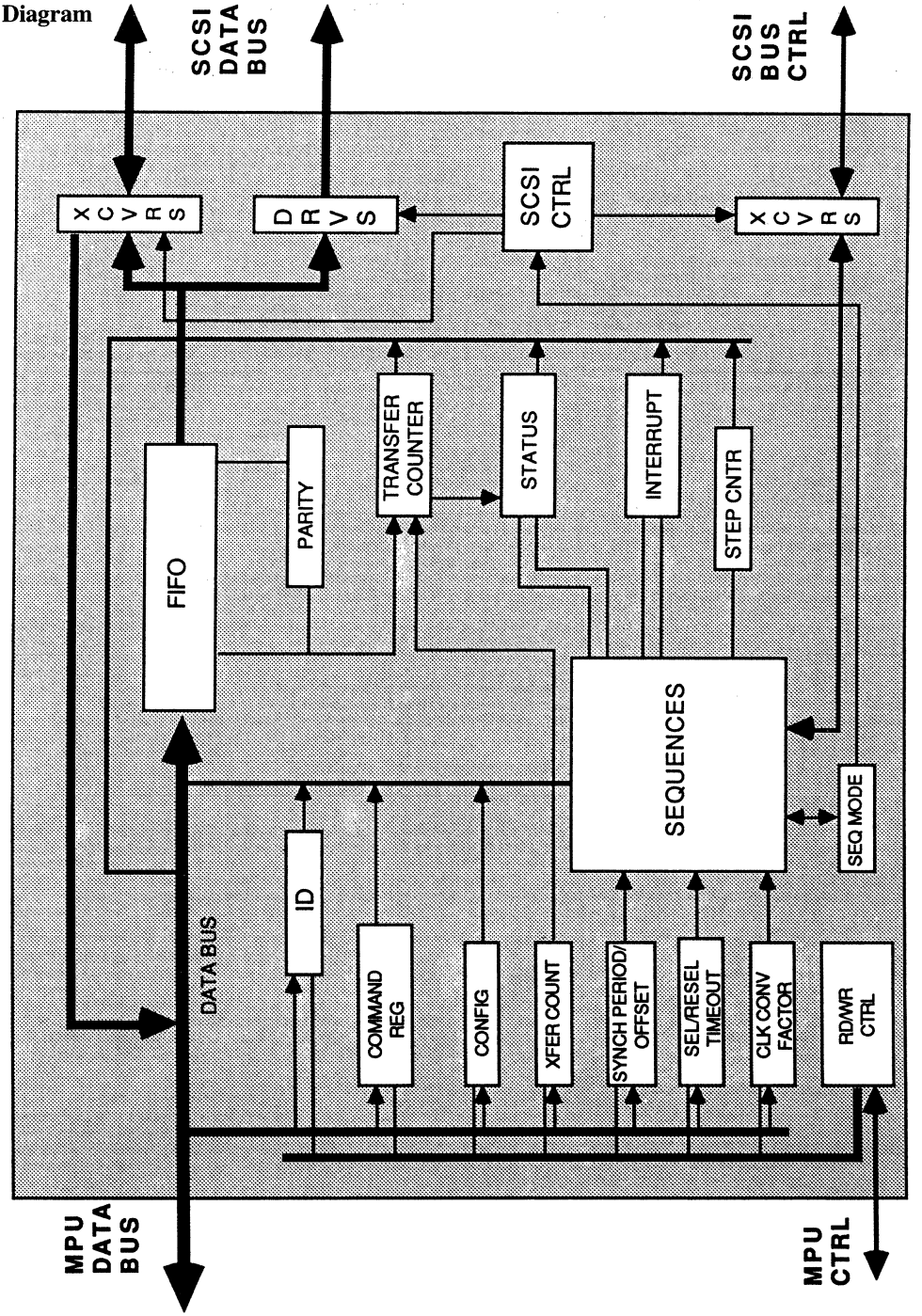
With its high level of integration, the ESP replaces SCI interface circuitry, which typically consists of discrete devices, external drivers, and/or lower performance SCSI controllers.

The ESP maximizes protocol efficiency by utilizing a command pipeline, composed of a dual-ranked command register and transfer counter, and combination commands to minimize host intervention. The 53C90 also maximizes Transfer Rates by sustaining Asynchronous data rates over 3.0 MBytes/sec and Synchronous data rates of 5.0 MBytes/sec. With its on-chip 48 mA single-ended drivers and receivers, the ESP can be directly connected to the SCSI bus, hence minimizing board space requirement. The highly integrated structure utilized by the ESP provides several benefits to its users.

Enhanced SCSI Processor (ESP) Features Summary

- Support of ANSI X3.131-1986 SCSI standard
- Buffer Controller interface for I/O and fast DMA
- On-chip 48 mA single-ended drivers and receivers
- Control logic for differential transceivers
- Parity generation with optional checking
- Initiator and target roles supported
- Asynchronous data transfers over 3.0Mbytes per second
- Synchronous data transfers to 5.0Mbytes per second
  - —Programmable synchronous transfer period
  - —Programmable synchronous transfer offsets up to 15
- Sixteen-byte data FIFO between the DMA and SCSI channels
- Pipelined command structure
- Common SCSI sequences without microprocessor intervention
  - —Selection sequence, from arbitration through command
  - —Reselection sequence, from arbitration through message
  - —Bus-initiated selection through received command
  - —Bus-initiated reselection through received message
  - —Command complete sequences
  - —Terminate and disconnect sequences
- Interrupts microprocessor only when service is required
  - —Disconnect or bus reset
  - —Selection/reselection sequence complete
  - —Target mode command complete or ATN detected
  - —Initiator mode command complete or phase change detected
  - —Waits until the phase is stable and REQ is asserted
- Clock rate up to 25 MHz
- Interfaces to eight-bit microprocessor data bus with minimal support logic
- CMOS low power requirements
- 68 pin PLCC and 80 pin QFP packages

Block Diagram



ESP Pin Description

The ESP pins are described below. The pin type is indicated by "I" for input, "O" for output and "B" for bi-directional.

DMA Interface Pins

DMA Interface			
Pin #	**Type**	**Signal**	**Description**
4-1, 68-65	B	DB7-DB0	Active-high data bus connected to the DMA Controller, CPU and buffer memory. Each pad contains a pullup to Vdd (12.5K Ω minimum).
60-57	I	A3-A0	Active-high address bus which specifies one of the ESP's internal registers for reading or writing.
56	I	CS/	Active-low chip select signal which enables access to the ESP's internal registers. CS/ and DACK/ must never be active at the same time.
55	I	RD/	Active-low read signal which enables ESP data onto DB7-DB0. CS/ or DACK/ must also be active.
54	I	WR/	Active-low write signal which strobes DB7-DB0 data into the ESP. CS/ or DACK/ must also be active.
53	O	DREQ	Active-high DMA request to the DMA Controller.
52	I	DACK/	Active-low DMA acknowledge signal from the DMA Controller. CS/ and DACK/ must never be active at the same time.
61	O	INT/	Active-low, open drain interrupt signal to the microprocessor.

Miscellaneous Pins

Miscellaneous Pins			
Pin #	**Type**	**Signal**	**Description**
50	O	IGS	Active-high initiator group select signal. This pin is high whenever the ESP is in initiator mode. It is used in differential mode to enable the initiator signals (ACKO/, ATNO/). When low the ESP should be receiving these signals.
49	O	TGS	Active-high target group select signal. This pin is high whenever the ESP is in target mode. It is used in differential mode to enable the target signals (REQO/, MSGO/, CDO/, IOO/). When low the ESP should be receiving these signals.
5	I	DIFFM	Differential mode enable. When this pin is grounded the ESP operates in single-ended mode, with separate SCSI data input and output buses. When this pin is high the ESP operates in differential mode, with bidirectional SCSI data on the SDI pins and active high differential transceiver enables on the SDO pins.
63	I	RESET	Active-high chip reset. When this pin is high the ESP registers and sequencers are initialized. This pin must not be connected to the RESETO pin.
62	O	RESETO	Active-high reset output. This pin is asserted whenever the RESET input pin is active. It is also asserted when a SCSI reset interrupt timeout occurs.
51	I	CLK	Square wave clock which generates internal chip timing. The maximum frequency is 25 MHz. For synchronous transfers, the minimum is 13 MHz. A minimum of 10MHz is required for asynchronous transfers.
15, 48		VDD	+5V power input (2 pins)
16, 21 27, 32 38, 64		VSS	Ground reference (6 pins)

SCSI Bus Interface Pins

SCSI Bus Interface			
Pin #	**Type**	**Signal**	**Description**
25-22 20-17 26	O O	SDO7/-SDO0/ SDOP/	48 mA, open drain SCSI data/parity bus. In single-ended mode (DIFFM=0), these pins output active-low data signals to the SCSI bus. In differential mode (DIFFM=1), these pins are used to control the direction of external differential transceivers. When a pin is high, the direction is out to the SCSI bus.
13-6 14	B	SDI7/-SDI0/ SDIP/	Schmitt trigger, active-low SCSI data/parity bus. In single-ended mode (DIFFM-0), these pins are used strictly for input from the SCSI bus. In differential mode (DIFFM=1), these pins carry bidirectional data/parity to and from the external SCSI bus transceivers.
28	O	SELO/	48 mA, open drain SCSI select signal. In single-ended mode, this pin is active low. In differential mode, this pin is active high. The pin is asserted by the chip when it attempts to select or reselect a SCSI device.
29	O	BSYO/	48 mA, open drain SCSI busy signal. In single-ended mode, this pin is active low. In differential mode, this pin is active high.
36	O	ATNO/	48 mA, open drain, active-low SCSI attention signal. This pin is only asserted when the ESP is in initiator mode.
31	O	ACKO/	48 mA, open drain, active-low SCSI acknowledge signal. This pin is only asserted when the ESP is in the initiator mode.
30	O	REQO/	48 mA, open drain, active-low SCSI request signal. This pin is only asserted when the ESP is in target mode.
33-35	O	MSGO/, CDO/, IOO/	48 mA, open drain, active-low SCSI phase signals. These pins are only asserted when the ESP is in target mode.
37	O	RSTO/	48 mA, open drain SCSI reset signal. In single-ended mode, this pin is active low. In differential mode, this pin is active high.
39	I	SELI/	Schmitt trigger, active-low SCSI select signal.
40	I	BSYI/	Schmitt trigger, active-low SCSI busy signal.
41	I	REQI/	Schmitt trigger, active-low SCSI request signal.
42	I	ACKI/	Schmitt trigger, active-low SCSI acknowledge signal.
43	I	MSGI/	Schmitt trigger, active-low SCSI message signal.
44	I	CDI/	Schmitt trigger, active-low SCSI control/data signal.
45	I	IOI/	Schmitt trigger, active-low SCSI input/output signal.
46	I	ATNI/	Schmitt trigger, active-low SCSI attention signal.
47	I	RSTI/	Schmitt trigger, active-low SCSI reset signal.

Differential Mode Enable Pin

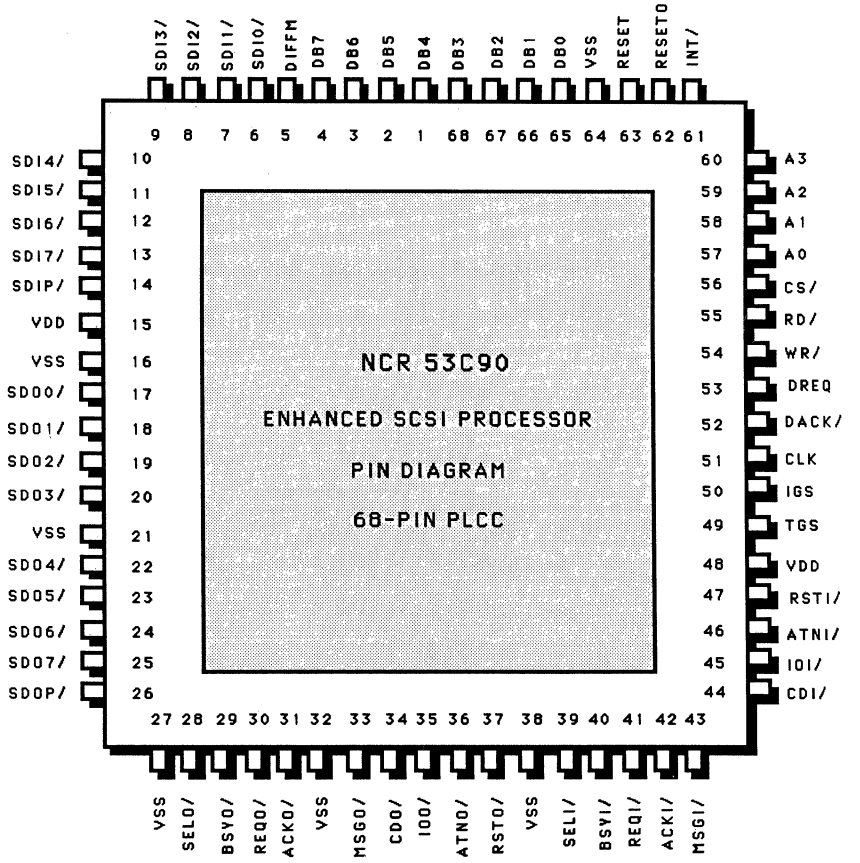
The differential mode enable pin alters the SCSI data paths in the ESP. When the pin is grounded, the ESP operates in single-ended mode with the SDI and SDO buses providing separate input data and output data, respectively. That is, the ESP drives data out on SDO7/-SDO0/ and receives data on SD17/-SDI0/. The SCSI data and control pins are all negative true.

When the differential mode enable pin is high, the SDI bus becomes a bidirectional data bus and the SDO bus is used to control the external differential transceivers. The transceivers have positive true output enables and negative true input enables. They are configured as all driving or all receiving, except during arbitration when only the bus ID bit is driven and all other bits are received. In addition, the polarity of the RSTO/, SELO/, and BSYO/ pins are set to positive true to enable external differential drivers for these signals.

Since the TGS and IGS output pins are true when the ESP is in target or initiator mode, respectively, they may be used to enable external differential drivers for the target and initiator groups of signals:

Target Signals	Initiator Signals
REQO/	ACKO/
MSGO/	ATNO/
CDO/	
IOO/	

ESP 68 Pin PLCC Pin Diagram



User-Accessible Register and Counters

The ESP has a number of registers which are used to configure, command, monitor, and pass data to the chip. These registers are listed in the table below and more fully described in the paragraphs that follow.

#	Read	Write
0	Transfer counter lo (LSB)	Transfer count lo (LSB)
1	Transfer counter hi (MSB)	Transfer count hi (MSB)
2	FIFO	FIFO
3	Command	Command
4	Status	Select/reselect bus ID
5	Interrupt status	Select/reselect timeout
6	Sequence step	Sync period
7	FIFO flags	Sync offset
8	Configuration	Configuration
9	NCR Reserved	Clock conversion factor
A	NCR Reserved	Test

Transfer Counter - Address 0,1 (Read only)

This 16-bit counter is used by the DMA commands and by the command sequence. It is loaded at the beginning of a DMA command (the contents of the Transfer Count Register is loaded into it) and when receiving a SCSI command (it is loaded with the decoded length of the command). The counter may be read to help determine the number of bytes remaining to be transferred if a sequence terminates early.

Transfer Count Register - Address 0,1 (Write only)

This 16-bit register is normally loaded prior to writing a command to the Command Register. The value in this register is transferred to the Transfer Counter at the beginning of a command which used DMA. The Transfer Count need only be loaded once for successive DMA command using the same count. Zero specifies maximum count (65536).

FIFO Register - Address 2 (Read/Write)

The FIFO is a 16-byte deep, first-in-first-out buffer between the SCSI bus and buffer memory. It is accessible by the microprocessor via this register.

Command Register - Address 3 (Read/Write)

The Command Register is an eight-bit read/write register used to give commands to the ESP. The register is double ranked, enabling the microprocessor to stack commands to the ESP. Once the bottom command has been completed and reported, the top command (if present) will fall to the bottom and execute. The value of the last executed (or executing) command will be contained in the bottom of the Command Register.

Bit 7 Enable DMA

When this bit is reset (0) and sending, the chip transmits data to the SCSI bus from the FIFO until it is empty. It does not modify the transfer counter.

When this bit is reset (0) and receiving, the chip receives a single byte into the FIFO. It does not modify the transfer counter.

When this bit is set (1) and sending, data will be transferred through the DMA channel into the FIFO. The counter is decremented as bytes are received by the DMA channel. Data will continue to be transmitted to the SCSI bus from the FIFO until it is empty and the transfer counter is zero.

When this bit is set (1) and receiving, the chip receives from the SCSI bus the number of bytes specified in the transfer counter into the FIFO, decrementing the transfer counter as bytes are received. Data will be transferred to the DMA channel from the FIFO until it is empty and the transfer counter is zero.

When this bit is receiving synchronous data in the Initiator mode, the transfer counter is decremented as bytes are removed by the DMA channel. The transfer completes when the transfer counter is zero.

Bit 6-0 Command Code

The ESP responds to 26 commands, by which the microprocessor can initiate data transfers, prepare the controller for selection, etc. Bits 6-4 specify the command mode:

Bit 6 Disconnected Mode

Bit 5 Target Mode

Bit 4 Initiator Mode

Bit 3-0 specify the command within the mode group.

Command Summary

The ESP's Command set is listed in the following table.

Bits 7654 3210	Command	Int
	MISC	
x000 0000	NOP	no
x000 0001	Flush FIFO	no
x000 0010	Reset Chip	no
x000 0011	Reset SCSI bus	no*
	DISCONNECTED	
x100 0000	Reselect sequence	yes
x100 0001	Select without ATN sequence	yes
x100 0010	Select with ATN sequence	yes
x100 0011	Select with ATN and stop sequence	yes
x100 0100	Enable selection/reselection	no
x100 0101	Disable selection/reselection	yes
	TARGET	
x010 0000	Send message	yes
x010 0001	Send status	yes
x010 0010	Send data	yes
x010 0011	Disconnect sequence	yes
x010 0100	Terminate sequence	yes
x010 0101	Target command complete sequence	yes
x010 0111	Disconnect	no
x010 1000	Receive message sequence	yes
x010 1001	Receive command	yes
x010 1010	Receive data	yes
x010 1011	Receive command sequence	yes
	INITIATOR	
x001 0000	Transfer information	yes
x001 0001	Initiator command complete sequence	yes
x001 0010	Message accepted	yes
x001 1000	Transfer pad	yes
x001 1010	Set ATN	no

*External connection of the RSTO/ pin to RSTI/ pin causes an interrupt if the SCSI reset interrupt is not disabled in the Configuration Register.

An illegal interrupt is generated if the ESP is not currently in the mode specified by bits 6-4, or if bits 3-0 specify an unsupported command.

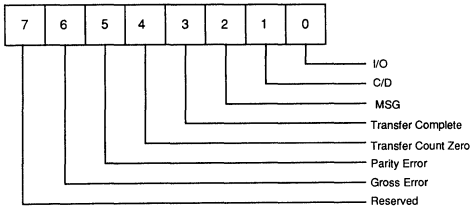
Certain conditions clear the Command Register, as summarized below:

- Chip reset
- SCSI bus reset or disconnect
- Completion of bus-initiated selection or reselection
- Completion of a select command
- Completion of a Reselect Command if ATN is asserted
- Select or reselect timeout
- Completion of a target disconnect or terminate command
- Bad parity received in the target mode
- Assertion of ATN in target mode
- Phase change during an initiator transfer
- Unexpected phase change during an initiator sequence
- Illegal command

Note the RESET chip and RESET SCSI Bus commands execute as soon as they are loaded into the top of the Command Register.

Status Register - Address 4 (Read only)

This eight-bit read-only register contains fields to indicate the status of the chip and to qualify the reason for an interrupt. Because several of these fields are reset when the Interrupt Register is read, the Status Register should always be read prior to the Interrupt Register. The fields within the Status Register are defined below.



Bit 7 Reserved

Bit 6 Gross Error
This status bit is set (1) when one of the following has occurred:

1. The top of the FIFO has been overwritten
2. The top of the Command Register has been overwritten
3. A DMA transfer occurs in the wrong direction
4. A phase change occurs in the middle of an initiator synchronous data transfer.

No interrupt is generated by the ESP when a gross error is detected. Bit 6 resets (0) if the Interrupt Register is read when the interrupt pin is active.

Bit 5 Parity Error
This bit is set to a one when parity is enabled in the Configuration Register and the ESP detects a parity error on a byte received from the SCSI bus. It is cleared if the Interrupt Register is read when the interrupt pin is active.

Bit 4 Transfer Count Zero
This bit is set (1) when the Transfer Counter decrements to zero. It resets when the Transfer Counter is loaded.

Bit 3 Transfer Complete
This status bit is set to a one during the target command sequence if the received command is in one of the non-reserved groups (0, 1, 5, 6 or 7). It is reset, if the Interrupt Register is read when the interrupt pin is active.

Bits 2-0 SCSI Phase
Bits 2 through 0 indicate the state of the SCSI MSG, C/D and I/O signals, respectively. These bits define the information phase being asserted by the target.

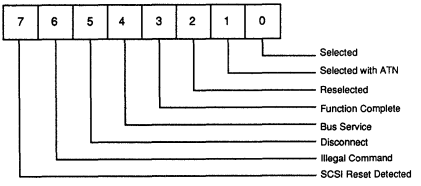
Bits 2 1 0	SCSI Bus Phase
0 0 0	Data out
0 0 1	Data in
0 1 0	Command
0 1 1	Status
1 0 0	Reserved for future standardization
1 0 1	Reserved for future standardization
1 1 0	Message out
1 1 1	Message in

Select/Reselect Bus ID Register - Address 4 (Write only)

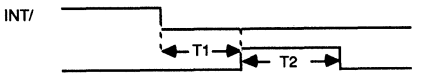
The Select/Reselect Bus ID Register is a three-bit write-only register which specified the encoded destination bus ID for a Select or Reselect command. (least significant 3 bits)

Interrupt Status Register - Address 5 (Read only)

This eight-bit read-only register is used in conjunction with the Status Register and Sequence Counter to determine the cause of an interrupt.



Bit 7 SCSI Reset Detected
This bit is set if the SCSI reset interrupt disable bit is zero in the Configuration Register and the chip detects a reset on the SCSI bus. If the interrupt is not serviced in T1 ms, the chip will assert RESETO for T2 μs. (See below)



T1 = 2 (Clock Period) [(Clock Conversion Factor •3841)-1]

T2 = 2 (Clock Period) [(65) (Clock Conversion Factor)]

Example: Clock = 24MHz, CCF = 5

T1 = 1.60033ms
T2 = 27.0833μs

Bit 6 Illegal Command
This bit is set when the ESP detects an unused code or illegal command.

Bit 5 Disconnect
This bit is set when the ESP is in initiator mode and the target disconnects from the bus, or if a selection or reselection timeout occurs. In the target mode, this bit is set when the Terminate Sequence or Command Complete Sequence command disconnects the ESP from the SCSI bus.

Bit 4 Bus Service
In target mode, this bit is set when the ATN signal is asserted, indicating the initiator is requesting service. In initiator mode, this bit will be set on command completion unless a message is received and ACK is still asserted.

Bit 3 Function Complete
This bit is set to indicate that an interrupting-type disconnected or target state command has completed. In initiator mode, this bit will be set when a message has been received and ACK is being asserted.

Bit 2 Reselected
This bit is set to indicate the chip has been reselected as an initiator.

Bit 1 Selected with ATN
This bit is identical to the selected bit except it indicates that the initiator asserted ATN during selection.

Bit 0 Selected
This bit is set to indicate the chip has been selected as a target.

Reading the Interrupt Register while the interrupt pin is active, causes the interrupt pin, Interrupt Status Register, and Sequence Step Registers to be cleared. Always read the Status Register (and the Sequence Step Register) prior to reading the Interrupt Register. If the ESP is waiting for the interrupt to be cleared to report the status of its command, the interrupt pin will be reasserted.

Select/Reselect Timeout Register - Address 5 (Write only)

The Select/Reselect Timeout Register is an eight-bit write-only register which specifies the amount of time to wait for a response during selection or reselection. This register is typically loaded to specify a timeout period of 250 ms (93H at 24 MHz). To determine the value for other timeouts use the following equation:

8192*CCF*tcp = TU

T/TU = Register Value (convert to Hex)

Where: CCF = Clock Conversion Factor (See next page)
tcp = Input Clock Period
TU = Time Unit
T = Desired Timeout

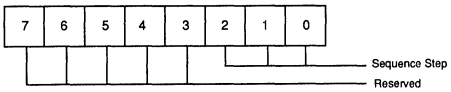
Note: Maximum timeout = 255TU

Sequence Step - Address 6 (Read only)

The Sequence Step Register is an eight-bit read-only register which indicates the current substep within certain ESP sequences when an interrupt occurs. The least significant three bits from an incrementing sequence to indicate which steps of a sequence were executed prior to the interrupt. The upper five bits are reserved. (Refer to page 45)

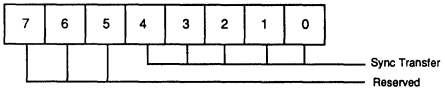
The sequences that utilize this register are:

Target	Initiator
Bus Initiated Selection with ATN Bus Initiated Selection	Select with ATN
Receive Command Sequence	Select with ATN and Stop
Command Complete Sequence	Select without ATN
Disconnect Sequence	
Terminate Sequence	



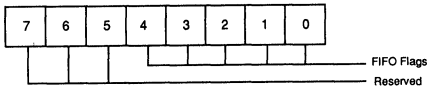
Synchronous Transfer Period Register - Address 6 (Write only)

The Synchronous Transfer Period Register is a five-bit write-only register which specifies the minimum time between leading edges of successive REQ or ACK pulses. The time unit is the input clock period. The minimum transfer period is five cycles and the maximum is 35 cycles. The default value is five cycles.



FIFO Flags Register - Address 7 (Read only)

This is a read-only register. The least significant five bits encode the sixteen FIFO full flags to indicate the number of bytes remaining in the FIFO. The remaining bits are reserved.

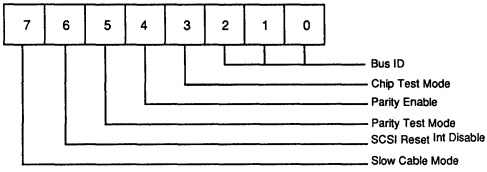


Synchronous Offset Register - Address 7 (Write only)

The Synchronous Offset Register is a four-bit write-only register (bits 3-0) which specifies the maximum REQ/ACK offset allowed during synchronous transfers. An offsett of zero specifies an asynchronous operation.

Configuration Register - Address 8 (Read/Write)

This eight-bit read/write register is used to specify different operating options for the ESP. The bits are defined on next page.



Bit 7 Slow Cable Mode
When set, this bit increases the minimum data setup time from two cycles to three when transmitting to the SCSI bus. The minimum asynchronous and synchronous transmit periods are also increased to four and six cycles, respectively. This bit should be enabled if excessive capacitance on the SCSI cable causes SCSI timing violations.

Bit 6 SCSI Reset Interrupt Disable
This bit disables the reporting of a SCSI reset condition. If a SCSI reset occurs with this bit set, the ESP will disconnect from the SCSI bus and remain in the idle state without reporting an interrupt or timing out the response.

Bit 5 Parity Test Mode
This bit causes the parity output to the SCSI bus to be generated from Bit 7 of the transmitted byte, rather than from the parity generator. This facilitates system debug by allowing the MPU to force bad parity on any byte. This bit must not be enabled in normal operation.

Bit 4 Parity Enable
When set (1), this bit enables parity checking whenever data is read from SCSI bus except during arbitration and when receiving pad bytes.

Bit 3 Chip Test Mode
When set (1), this bit enables special test circuitry which is used during chip test. This mode must not be enabled in normal operation.

Bits 2-0 Bus ID
The bus ID is a three-bit field which specifies the encoded value of the SCSI Bus ID to which the ESP chip responds during a bus-initiated selection or reselection, and the bus ID to assert during arbitration and selection or reselection.

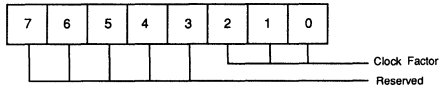
Clock Conversion Register - Address 9 (Write only)

The Clock Conversion Factor is a three-bit write-only register which is used for generating all timings in the ESP longer than 400 ns. The Clock Conversion Factor must never be loaded with the value 1.

Clock Conversion Factor = 200/tcp

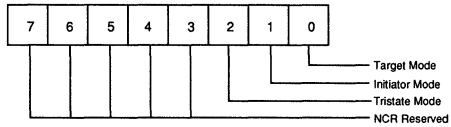
Where: tcp = input clock period (ns)

CLK Freq (MHz)	CLK Conv. Factor
10	2
10.01 to 15	3
15.01 to 20	4
20.01 to 25	5



Test Register - Address A (Write only)

This write-only register utilizes the least significant three bits to place the chip in various mades for testing. To set these bits, first put the ESP into test mode by setting bit 3 of the Configuration Register. Next load the appropriate bit value into the Test Register. To clear the test mode, the ESP must be reset by either asserting the RESET input or issuing a RESET chip command. Caution must be exercised when using this register.



Bit 2 Tri-State Test Mode
This mode allows the ESP to support board-level automatic testing. When this bit is set, all bidirectional and output pins are forced into a high impedance state, regardless of the chip inputs.

Bit 1 Initiator Mode
This bit, when set, will put the ESP into the initiator mode.

Bit 0 Target Mode
When this bit is set, the ESP is placed into the target mode.

Sequencer

The sequencer modules are briefly described below. These modules are used internally by the ESP and are not user accessible. However, they are used by commands/registers as described. Refer to the functional description section for more information (See page 37).

Bus Delay Module

The bus delay module detects whether the SCSI bus has been in the bus free state for more than a bus settle delay plus a bus free delay (1200 ns min).

The module also generates the timing for a bus settle delay (400 ns min), an arbitration delay (2200 ns min), a reset hold delay (25 μs min), a selection abort delay (200 μs), and the select/reselect timeout clock (1.7 ms). The module contains a twelve bit counter clocked by the conversion clock to generate the timing.

Phase Monitor Module

The phase monitor module monitors the SCSI bus to detect phase changes. If the ESP is an initiator and BSY and SEL are both released, a disconnect interrupt is generated and the sequencer is halted.

DMA Controller Interface

The DMA controller interface performs the actual transfers over the DMA channel. The module is coupled to the transfer counter and the FIFO.

SCSI Interface Modules

The SCSI interface modules perform the actual transfers over the SCSI bus. The mode of operation depends on whether the transfers are synchronous or asynchronous, single or multi-byte, target or initiator, and single ended or differential. The SCSI interface modules are coupled to the Transfer Counter, the FIFO, Synchronous Transfer Period Counter, and Synchronous Offset Counter. These modules are used internally by the ESP and are not accessible. However, these are used by the various commands/registers.

Odd parity is generated for the SCSI data out bus. If the number of asserted data bits is even, the parity bit is asserted. Parity is checked on the SCSI data-in bus if parity checking is enabled in the Configuration Register. During arbitration and when receiving pad bytes, parity is not checked.

Bus Initiated Selection/Reselection

The Bus Initiated Selection/Reselection sequence responds to selection by a SCSI initiator, including checking for and receiving an identify message and a command block. It also responds to reselection by a SCSI target, including checking for and receiving a message byte.

Command Sequence

The Command Sequence module decodes the length of an incoming command from information in the first byte, loads the transfer counter, and receives the appropriate number of bytes as described in the functional description (See page 37).

Disconnected State Commands

The Disconnected State Commands module arbitrates and selects a SCSI target with optional message and command transmission, or reselects a SCSI initiator with message byte transmission, as described in the functional description (See page 39).

Target Commands

The Target Commands module implements the target mode commands, as described in the functional description (See page 40).

Initiator Commands

The Initiator Commands module implements the initiator mode commands, as described in the functional description (See page 42).

Miscellaneous Blocks

The following section describes various functional blocks that are utilized within the ESP. These modules/blocks are for information only and are not user accessible.

ID Module

The ID module determines whether there is an ID match by using the three-bit bus ID contained in the configuration register to pick a bit in the SCSI data-in bus. It also determines if the proper number of bits are set.

The ID module also checks the SCSI data bus for a set bit which is higher than the bus ID contained in the configuration register. If there is none, the ESP has won arbitration.

If the ESP is in the Arbitration Phase, the ID module generates the destination ID which is the OR of the arbitration ID and the decoded select/reselect bus ID. The parity bit is always asserted, which forces bad parity if the select/reselect ID is inadvertently set equal to the arbitration ID.

Read/Write Control

The Read/Write Control Module handles the I/O interface with the DMA Controller when the microprocessor wishes to access one of the user-accessible registers in the ESP.

Select/Reselect Timer

The Select/Reselect timer is an eight-bit timer which is clocked by the sequencer bus delay module. With a 24 MHz chip clock, the timer is clocked approximately once every 1.7 ms. The timer is compared with the Select/Reselect Timeout Register to determine whether a select/reselect timeout has occurred (See page 11).

Synchronous Offset Counter

The synchronous offset counter is a four-bit up/down counter used to control synchronous data transfers.

The offset counter also contains logic to detect whether the counter is zero or if it is equal to the contents of the synchronous offset register.

Command Decode

The command decode circuit decodes bit 7-5 of the first byte received in the command phase to determine the command group and thus the length of the incoming command. The byte is decoded from the top of the FIFO.

SCSI Command Block Description

Group	Bytes Stored	Description
0	6	Six-byte command
1	10	Ten-byte command
2	6	Reserved
3	6	Reserved
4	6	Reserved
5	12	Twelve-byte command
6	6	Vendor unique (six bytes)
7	10	Vendor unique (ten bytes)

Sequencer Mode Register

The Sequencer Mode Register stores information about the sequencer. The fields within the register are defined below.

Target
Indicates that the ESP is a target. This bit drives the TGS output pin which controls the target group of transceivers when operating with differential drivers.

Initiator
Indicates that the ESP is an initiator. This bit drives the IGS output pin which controls the initiator group of transceivers when operating with differential drivers.

Select/Reselect Enable
Indicates that the ESP has received an Enable Selection/Reselection command, enabling it to perform the bus-initiated selection and reselection sequences.

Select/Reselect DMA Enable
Indicates that the Enable Selection/Reselection command was issued with the DMA bit set. If this bit is set and the microprocessor issues a Select or Reselect command with DMA, an illegal command interrupt is generated.

Functional Description

The following section contains the functional descriptions for the ESP including Bus Initiated Sequences, Commands and Flow Charts.

Initialization

The ESP has three levels of reset. The hard reset (H), applied when the reset pin is high or when a Reset Chip command is executed, resets the entire chip. The soft reset (R), applied when HRST or SCSI bus reset is active, resets a subset of the functions reset by the hard reset. The disconnect reset (D), applied when the soft reset is active or a disconnect occurs, resets a subset of the functions reset by the soft reset.

Initialization for Different Levels of Reset

Reset	Description
H	Set clock conversion factor to 2
	Clear configuration register bits:
H	Chip test mode = 0
H	Parity enable = 0
H	Parity test mode = 0
H	SCSI reset interrupt disable = 0
H	Slow cable mode = 0
H	Initialize FIFO to empty
H	Set synchronous period to 5
H	Set synchronous offset to 0
H	Release SCSI RSTO signal*
H	Release interrupt pin
H	Clear interrupt register to 0
	Clear status register bits:
H	Transfer complete = 0
H	Parity error = 0
H	Gross error = 0
HR	Transfer count zero = 0
HR	Reset DMA interface
HR	Reset bus-initiated selection/reselection module
HR	Reset command sequence module
HR	Reset sequence step
	Clear sequencer mode bits:
HR	Enable sel/resel = 0
HRD	Target = 0
HRD	Initiator = 0
HRD	Initialize command register FIFO to empty
HRD	Release all SCSI signals*
HRD	Reset disconnect, initiator, and target command modules

*Note: RSTO/ is only released by a Hard (H) reset

When the RESET input signal is asserted, it is recognized immediately by the ESP, but the chip releases it internally in a synchronous manner, 0.5 to 1.5 cycles after RESET is released. RESET must be asserted for a minimum of two clock cycles.

The reset output pin, RESETO, is the OR or RESET and the SCSI reset interrupt timeout, described on next page. RESETO is generally used to reset the microprocessor and other onboard circuitry.

The microprocessor should program the following registers as part of its start up procedure:

-Configuration Register
-Clock Conversion Factor

The microprocessor must initialize the following registers prior to their use, since they are not initialized by a chip reset:

-Bus ID (in the Configuration Register)
-Transfer Count low and high bytes
-Select/Reselect bus ID
-Select/Reselect timeout

Bus Initiated Sequences

BUS INITIATED SEQUENCES
Selection
Reselection
SCSI Bus Reset

Bus Initiated sequences occur during the disconnected state when the ESP chip is selected or reselected by another initiator or target. The Enable Selection/Reselection command must have been issued before either sequence will be initiated. Bit 7 of the Enable Selection/Reselection command enables DMA during the Bus Initiated Sequences. If bit 7 is zero, any information transferred during one of these sequences will be stored in the FIFO. If bit 7 is a one, the information will be stored in memory using DMA.

These sequences reduce processor overhead by automating the processes and creating an interrupt only after a process is complete or an abnormal condition occurs.

The Select and Reselect command sequences automatically disable Bus Initiated selection/reselection after completing the selection or reselection. Normally, the microprocessor has 250 ms (the recommended select/reselect timeout) to issue the Enable Selection/ Reselection command to the ESP chip after the chip disconnects from the bus. If this time is exceeded, an initiator or target which is attempting to select or reselect the ESP chip may timeout.

While a Bus Initiated Sequence is in progress, the command register is held reset. If a command is issued while a Bus Initiated sequence is in progress, the command will be ignored. If a Select or Reselect command was executing when the bus-initiated sequence began, the command will abort.

Selection

The Selection sequence will be performed when the Enable Selection/Reselection command has been issued and the chip is selected as a target by an initiator.

If the sequence completes normally, the following data will have been transferred from the initiator:

Selection bus ID	1 Byte
Identify message	1 Byte
Command descriptor block	6,10 or 12 Bytes

This data will be in the FIFO if the Enable Selection/Reselection command was issued without DMA enabled.

The selection bus ID byte is stored to enable the microprocessor to determine the initiator's ID. (The byte is not encoded). The absence of an initiator ID indicates that the system is in single-initiator mode.

The sequence terminates early if the received message byte is not an Identify message, if ATN is still asserted after the Identify message has been received, or if bad parity is received.

Reselection

The Reselection sequence will be performed when the Enable Selection/Reselection command has been issued and the chip is reselected as an initiator by a target.

If the sequence completes normally, the following data will have been transferred from the target:

Reselection bus ID	1 Byte
Message In (Identify)	1 Byte

The data will be in the FIFO if the Enable Selection/Reselection command was issued without DMA enabled.

The reselection bus ID byte is stored to enable the microprocessor to determine the target's ID. The byte is not encoded.

If the message byte was received, ACK will still be asserted on the SCSI bus and the host processor must issue a Message Accepted command to the ESP chip to release it. The host processor may reject the message by issuing the Assert ATN command prior to the Message Accepted command.

The sequence terminates early if Message In Phase was not asserted by the target.

SCSI Bus Reset

A SCSI bus reset may occur when the ESP is in any mode. The reset causes the ESP to disconnect from the SCSI bus and reset several internal states. (See Initialization, page 37).

If the SCSI reset interrupt has not been disabled in the configuration register, the ESP will set the interrupt bit and generate an interrupt to the microprocessor. If the interrupt is not serviced in approximately 1.7 ms, the ESP will assert the RESETO pin for 25 μs. (See page 32). It then loops back to wait another 1.7 ms for the interrupt to be serviced. This provides a watchdog timeout on the SCSI reset interrupt which can be used to reset the rest of the board in the event that the microprocessor is locked in an erroneous state.

If the SCSI bus reset is still active when the microprocessor clears the interrupt, a new interrupt will be generated which must be serviced.

Commands

The microprocessor issues commands to the ESP by writing to the dual-ranked Command Register. Caution should be exercised when overlapping commands to the ESP chip. Commands which transfer data in one direction should not be overlapped with commands that transfer data in the opposite direction. Also, for overlapped DMA commands, the transfer count must be loaded prior to loading the corresponding command. Care should be taken when overlapping non-DMA transmit commands, since many commands will completely empty the FIFO leaving nothing for the second command to transmit.

The 53C90 command set is divided into Disconnected, Target, Initiator, and Miscellaneous types:

Disconnected State Commands

The following commands are valid only when the 53C90 is in the disconnected state.

DISCONNECTED STATE COMMANDS		
Non-DMA	DMA	
40	C0	Reselect Sequence
41	C1	Select without ATN Sequence
41	C2	Select with ATN Sequence
42	C3	Select with ATN and Stop Sequence
44	C4	Enable Selection/Reselection
45	C5	Disable Selection/Reselection

If any of these commands are issued when the chip is not in a disconnected state, they will be ignored, the chip will generate an illegal command interrupt, and the Command Register will be cleared.

These sequences reduce processor overhead by automating the Selection or Reselection process. An interrupt is generated only after the entire sequence is complete or an abnormal condition occurs.

Reselect Sequence

This command will start the Reselect Sequence to connect the ESP as a target to an initiator.

Prior to issuing this command the microprocessor must set the timeout register, Select/Reselect bus ID registers and, if DMA is enabled, the Transfer Count Register must be set to 1. If DMA is not enabled, the message must be loaded into the FIFO prior to issuing this command to the ESP.

When the Reselect Sequence completes, the following data will have transferred to the Initiator:

Identify Message	1 byte

The sequence terminates early if a Reselect timeout occurs. This sequence does not use the Sequence Step Register.

Select without ATN Sequence

This command connects the chip as an intiator. A command block is sent to the target.

Prior to issuing this command the microprocessor must set the Timeout Register, Select/Reselect bus ID Registers and, if DMA is enabled, the Transfer Count Register must be set to the total length of the command. If DMA is not enabled, the data must be loaded into the FIFO prior to giving the command.

When this command completes normally, the following data will have transferred to the target:

Command Descriptor Block	6, 10, or 12 bytes

This command terminates early if:

-A select timeout occurs
-Target does not assert Command Phase
-Target removes Command Phase early

Select with ATN Sequence

The Select with ATN sequence command connects the chip as an initiator. A message and command block are sent to the target.

Prior to issuing this command the host processor must set the Timeout Register, Select/Reselect bus ID Registers and, if DMA is enabled, the Transfer Count Register must be set to the total length of the message and command. If DMA is not enabled, the data must be loaded into the FIFO prior to issuing this command.

The following data is transferred if Select with ATN completes normally:

Identify Message	1 byte
Command Descriptor Block	6, 10, or 12 bytes

The message can only be one byte in length. This command terminates early if:

-Select timeout occurs
-Target does not assert Message Out or Command Phase
-Target removes Command Phase early

Select with ATN and Stop Sequence

The Select with ATN and Stop sequence command should be used instead of the Select with ATN sequence if a message longer than one byte must be transferred. This command will start the Select sequence and stop after one message byte has been sent. ATN will remain asserted on the SCSI bus after the sequence completes. This allows the Initiator to send another message after the ID message has been sent.

This command terminates early if:

-Select timeout occurs
-Target does not assert Message Out Phase

This command is used to transfer extended messages such as Modify Data Pointers, Synchronous Data Transfer Requests and Extended Identify.

Enable Selection/Reselection

This command allows the ESP to respond to bus-initiated selection or reselection. Normally, this command is issued by the microprocessor to the ESP within 250ms (recommended select/reselect timeout) after the chip disconnects from the bus.

If DMA is enabled, the ESP will DMA the received data to buffer memory. If this bit is a zero, the received data will be left in the FIFO.

Disable Selection/Reselection

The Disable Selection/Reselection command disables the ESP from responding to bus-initiated selection or reselection. If the ESP has not yet begun a bus-initiated selection or reselection sequence, this command resets the internal mode bits which were previously set by the Enable Selection/Reselection command. The ESP then generates a function complete interrupt to the microprocessor.

If bus initiated selection or reselection has already begun, the Command Register is held reset so that all incoming commands are ignored. The ESP will generate selected or reselected interrupt when it completes.

Target Commands

The following commands are valid only when the chip is connected as a target.

TARGET COMMANDS		
Non-DMA	DMA	
20	A0	Send Message
21	A1	Send Status
22	A2	Send Data
23	A3	Disconnect Sequence
24	A4	Terminate Sequence
25	A5	Target Command Complete Sequence
27	A7	Disconnect
28	A8	Receive Message Sequence
29	A9	Receive Command
2A	AA	Receive Data
2B	AB	Receive Command Sequence

If any of these commands are issued when the chip is not in target mode, it will be ignored, the chip will generate an illegal command interrupt, and the Command Register will be cleared.

Prior to issuing a send command, if DMA is enabled, the transfer counter must be set to the total number of bytes to be read from buffer memory. If DMA is not enabled, the data must be loaded into the FIFO. Prior to issuing a receive command, if DMA is enabled the Transfer Count must be loaded with the total number of bytes to transfer from the SCSI bus. If DMA is not enabled, only one byte will be received into the FIFO.

Normally, the commands terminate with a function complete interrupt. If ATN is asserted, the bus service interrupt is also set and the Command Register is cleared.

If ATN is asserted when the ESP is idle, a bus service interrupt is generated without the function complete bit set and the Command Register will be cleared.

Send Message (Message In Phase)

This command will change the phase lines to the Message In Phase and transmit bytes to the SCSI bus.

A function complete interrupt will be generated when the command is completed. If ATN is asserted during the execution of the command, the bus service interrupt will also be set, the Command Register will be cleared, and the command will terminate early.

Send Status (Status Phase)

This command will change the phase lines to the Status Phase and transmit bytes to the SCSI bus.

A function complete interrupt will be generated when the command is completed. If ATN is asserted during the execution of the command, the bus service interrupt will also be set, the Command Register will be cleared, and the command will terminate early.

Send Data (Data In Phase)

This command will change the phase lines to the Data In Phase and transmit bytes to the SCSI bus.

A function complete interrupt will be generated when the command is completed. If ATN is asserted during the execution of the command, the bus service interrupt will also be set, the Command Register will be cleared, and the command will terminate early.

Disconnect Sequence

This command will change the phase lines to the Message In Phase, send two message bytes to the initiator, and then disconnect from the SCSI bus. The Save Data Pointers and Disconnect messages will normally be sent. A function-complete and disconnect interrupt will be generated when the command is completed.

If ATN is asserted, the function complete and bus-service bits will be set, the Command Register is cleared, and the command will terminate early without disconnecting.

Terminate Sequence

This command will change the phase lines to the Status Phase and send one status byte, change to the Message In Phase and send one message byte, and then disconnect from the SCSI bus. This sequence is used to complete a command and will normally send the status followed by a Command Complete Message. A function complete and disconnect interrupt will be generated when the command is completed.

If ATN is asserted, the function-complete and bus-service bits will be set, the Command Register is cleared, and the command will terminate early without disconnecting.

Target Command Complete Sequence

This command will change the phase lines to the Status Phase and send one status byte, then change to the Message In Phase and send one message byte. This sequence is used to complete a linked command and will normally send the status followed by a Command Complete message. A function complete interrupt will be generated when the command is completed.

If ATN is asserted, the function-complete and bus-service bits will be set, the Command Register is cleared, and the command will terminate early without disconnecting.

Disconnect

This command releases all SCSI bus signals except RSTO and returns the ESP to a disconnected state (refer to the earlier initialization description). No interrupt is generated to the microprocessor.

Receive Message Sequence (Message Out Phase)

This command causes the ESP chip to change the phase lines to the Message Out Phase and receive bytes from the SCSI bus. The sequence generates a function complete interrupt when the command is completed. If ATN is still asserted, the bus service interrupt bit will also be set and the Command Register will be cleared.

If a parity error is detected, the ESP chip will receive message bytes and discard them until ATN is false. A function complete interrupt is generated, and the Command Register is cleared.

Receive Command (Command Phase)

This command will change the phase lines to the Command Phase and receives bytes from the SCSI bus. A function complete interrupt will be generated when the command is completed. Also, if bad parity was received during the transfer the Command Register will be cleared when the command completes.

If ATN is asserted during the execution of the command, the bus service interrupt will also be set, the Command Register will be cleared, and the command will terminate early.

Receive Data (Data Out Phase)

This command will change the phase lines to the Data Out Phase and receive bytes from the SCSI bus.

A function complete interrupt will be generated when the command is completed. Also, if bad parity was received during the transfer, the Command Register will be cleared.

If ATN is asserted during the execution of the command, the bus service interrupt will also be set, the Command Register will be cleared, and the command will terminate early.

Receive Command Sequence (Command Phase)

This command will change the phase lines to the Command Phase and receive bytes from the SCSI bus. The ESP examines the first byte received to determine the length of the command. If the length is unknown, the Transfer Counter will be loaded with 5 and the Transfer Complete bit in the Status Register will be cleared. Otherwise, the command is decoded, the Transfer Counter is loaded with the appropriate value (5, 9 or 11), and the Transfer Complete bit is set.

If bad parity is detected, the sequence will terminate early and the Command Register will be cleared. If ATN is set, the sequence does not terminate early but the bus service bit will be set and the Command Register will be cleared when the sequence completes.

Initiator Commands

The following commands are valid only when the chip is connected as an initiator. If any of these commands are issued when the chip is not in Initiator mode they will be ignored, the chip will generate an illegal command interrupt, and the Command Register will be cleared.

INITIATOR COMMANDS		
Non-DMA	DMA	
10	90	Transfer Information
11	91	Initiator Command Complete Sequence
12	92	Message Accepted
18	98	Transfer Pad
1A	9A	Set ATN

If BSY becomes false, the ESP generates a disconnect interrupt and returns to the disconnected state. (See page 37) Due to synchronization latency, a disconnect is detected by the ESP 1.5 to 3.5 cycles after it actually occurs on the bus.

Prior to issuing a command to send data to the SCSI bus, if DMA is enabled the transfer count must be set to the total number of bytes to read from buffer memory. If DMA is not enabled, the data must be loaded into the FIFO.

Prior to issuing a command to receive data from the SCSI bus, if DMA is enabled the transfer count must be loaded with the total number of bytes to transfer from the SCSI bus. If DMA is not enabled, only one byte will be accepted by the FIFO.

Received bytes are checked for parity if the parity enable bit is set in the Configuration Register. If a parity error is detected, the chip asserts ATN prior to deasserting ACK for the byte which was error. It also sets the parity error status bit.

Transfer Information

The Transfer Information command is used to perform general transfers over the SCSI bus while in initiator mode. The ESP will continue to Transfer Information until one of four terminating events occurs:

- The transfer is complete. If the phase is Message Out, the chip removes ATN prior to asserting ACK for the last byte of the message. When the target asserts REQ, a bus service interrupt is generated.
- The target changes the information phase before the expected number of bytes was transferred. The ESP clears the Command Register and generates a bus service interrupt when REQ is asserted by the target.
- The target releases BSY, which generates a disconnect interrupt and a disconnect reset (refer to the earlier initialization description).
- The chip receives the last byte of a Message In Phase. The chip leaves ACK asserted and generates a function complete interrupt.

A Transfer Information command to transfer synchronous data must use DMA. Synchronous data in transfers are unique in that the target may at any time send data and REQ pulses to the initiator. These bytes must be stored in the FIFO as they are sent. Initiator transfers are handled internally as follows:

- All Message In and Status Phase transfer are handled one byte at a time. If DMA is enabled, the next byte will not be received until the current byte has been written to buffer memory and the FIFO is empty.

If the phase changes to Synchronous Data In, the DMA interface is immediately disabled to ensure that no data bytes are transferred to buffer memory. Also, if bad parity is detected on the incoming data bytes, it is not immediately reported. This is because the status register must reflect the status of the completing command, not the new data. The status bit and ATN are set when the next Transfer Info command begins executing.

- All Message Out and Command Phase transfers are handled one byte at a time. If DMA is enabled, the DMA interface only issues one DMA request at a time. If the phase changes to synchronous Data In, the DMA interface is already inactive so no extra bytes will be loaded into the FIFO.

If DMA is not enabled and the phase changes to Synchronous Data In, there may be bytes remaining in the FIFO which have not yet been transmitted. The FIFO flags register is latched to indicate the number of non-data bytes that were in the FIFO, and then the FIFO is flushed of these bytes. The FIFO will contain only the incoming data bytes.

- In synchronous Data Out Phase, the offset counter is incremented as REQ pulses are received. The transfer completes when the FIFO is empty and the transfer counter is zero. The offset counter may or may not be zero, depending on whether the target continues requesting synchronous data.

- In synchronous Data In Phase, the transfer counter is clocked by the DMA interface as bytes are unloaded from the FIFO, rather than being clocked as bytes are loaded into the FIFO. The transfer completes when the transfer counter is zero. The FIFO may or may not be empty, depending on whether the target continues sending synchronous data.

Initiator Command Complete Sequence

The command will normally be issued when a status phase is received. If the sequence completes normally, the following data will have been transferred from the target:

Status	1 Byte
Message In	1 Byte

The data will be in the FIFO if the command was not issued with DMA enabled.

The sequence terminates early if the target does not assert the Message In Phase or if it disconnects. This sequence does not utilize the Sequence Step Register.

Message Accepted

The Message Accepted command releases the ACK signal. Upon completion of this command, the chip waits for the target to assert REQ and then generates a bus service interrupt.

When the chip receives the last byte of a message, it leaves the ACK signal asserted so that ATN can be asserted with the Set ATN command if the message is to be rejected. The Message Accepted command must always be issued to release ACK if it was left asserted whether or not ATN is asserted.

Transfer Pad

The Transfer Pad command is similar to the Transfer Information command, except that the information being transferred is null data. For an output transfer, the ESP uses the FIFO to generate pads for the SCSI bus. For an input transfer, the ESP receives data from the SCSI bus into the top of the FIFO without checking parity, and then discards it from the bottom of the FIFO.

A Transfer Pad command to transmit pad bytes must have DMA enabled. No DMA requests are actually made, but the ESP uses the Transfer Counter to determine when the transfer is complete.

The command terminates under the same conditions as the Transfer Information command, except that the chip does not leave ACK asserted on the last byte of a message in phase. If the command terminates early (due to a phase change or a disconnect), the FIFO may contain pad bytes.

Set ATN

The Set ATN command causes the ATN signal to be asserted. No interrupt is generated upon completion of this command.

The ESP automatically asserts the ATN signal for the following cases:

- A Select with ATN command is issued and arbitration is won.
- A parity error is detected on a byte received in initiator mode. ATN is asserted prior to releasing ACK.

The ESP automatically releases the ATN signal during the last byte of the Transfer Information command for a Message Out Phase. ATN is released prior to asserting the ACK signal.

Miscellaneous Commands

Miscellaneous Commands described in this section are valid in all modes. These commands do not check for ATN. If the ESP is a Target device and ATN happens to be asserted during the execution of one of these commands no bus service interrupt will be generated. It will be reported on completion of the next Target command, if the chip does not become disconnected.

MISCELLANEOUS COMMANDS		
Non-DMA	DMA	
00	80	NOP
01	81	Flush FIFO
02	82	Reset Chip
03	83	Reset SCSI Bus

NOP Command

The NOP command performs no operation. It is also used following a Reset Chip command to free the ESP Command Register. No interrupt is generated on completion of this command.

Flush FIFO Command

The Flush FIFO command initializes the FIFO to the empty condition. No interrupt is generated on completion of this command.

Reset Chip Command

The Reset Chip command resets all functions in the chip and returns it to a disconnected state. The command has the same effect as a hardware reset (refer to the earlier initialization description).

The Reset Chip command will remain in the top of the Command Register (locking the ESP and its registers in a reset state) until a new command is loaded. The new command must be a NOP command. This command might not be readable or executable.

Reset SCSI Bus Command

The Reset SCSI Bus command asserts the SCSI RSTO pin for 25μs (see page 32). The chip is returned to a disconnected state. No interrupt is generated on completion of this command. Since the RSTI pin will be externally connected to the RSTO pin, an SCSI reset interrupt will be generated if it is not disabled in the Configuration Register.

Illegal Command

An illegal command is a command that is not allowed in the current ESP mode as specified by bits 6-4 of the Command Register or is an unsupported command as determined by bits 3-0. The chip will generate an illegal command interrupt to the microprocessor and clear the Command Register when an illegal command occurs.

An illegal command interrupt will also be generated if an initiator Transfer Information, Transfer Pad, or Command Complete command is issued when ACK is still asserted. Also, an illegal command interrupt is generated when a Select or Reselect command is issued with DMA if the Enable Selection/Reselection command was previously issued with DMA enabled.

To determine if an illegal command was written to the chip, check for the occurrence of an interrupt (i.e. INT/active) and bit 6 of the Interrupt Status Register (register 5) being set.

Sequence Steps and Flow Charts

Sequence Steps

Bus Initiated Selection with ATN (Target)

Bus Initiated Selection with ATN (Target)		
SEQ STEP 2 1 0	INT REG 7 6 5 4 3 2 1 0	DESCRIPTION
0 0 0	0 0 0 0 0 0 1 0	Selected with ATN, received bus ID and one message byte. Check parity bit in Status Register.
0 0 0	0 0 0 1 0 0 1 0	Same as above; ATN on
0 0 1	0 0 0 0 0 0 1 0	Command Phase; all bytes not received because of parity error. Check Transfer Complete bit in Status Register. Check FIFO Flags/Counter.
0 0 1	0 0 0 1 0 0 1 0	Same as above; ATN on
0 1 0	0 0 0 0 0 0 1 0	Command Phase completed. Check transfer complete bit in Status Register
0 1 0	0 0 0 1 0 0 1 0	Same as above; ATN on

Bus Initiated Selection (Target)

Bus Initiated Selection (Target)		
SEQ STEP 2 1 0	INT REG 7 6 5 4 3 2 1 0	DESCRIPTION
0 0 0	0 0 0 0 0 0 0 1	Selected; received Initiator ID
0 0 1	0 0 0 0 0 0 0 1	Command Phase; all bytes not received because of parity error. Check FIFO Flags/ Counter.
0 0 1	0 0 0 1 0 0 0 1	Same as above; ATN on
0 1 0	0 0 0 0 0 0 0 1	Command Phase completed. Check Transfer Complete bit in Status Register.
0 1 0	0 0 0 1 0 0 0 1	Same as above; ATN on

Select with ATN (Initiator)

Select with ATN (Initiator)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00100000	Arbitration completed; selection timeout; Disconnected
000	00011000	Arbitration and select completed. Not Message Out Phase; ATN on
010	00011000	Arbitration and Select completed; Message Out Phase; sent one byte. ATN off; Not Command Phase
011	00011000	Command Phase. All command bytes not transferred because of premature phase change
100	00011000	Select with ATN completed, not Command Phase

Select with ATN and Stop (Initiator)

Select with ATN and Stop (Initiator)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00100000	Arbitration completed; Selection timeout; disconnected.
000	00011000	Arbitration and Select completed. Not Message Out Phase; ATN on
001	00011000	Sent one message byte, ATN on

Select without ATN (Initiator)

Select without ATN (Initiator)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00100000	Arbitration complete; Select timeout. Disconnected
010	00011000	Arbitrate and Select completed. Not Command Phase.
011	00011000	Aribtrate and Select completed. Command Phase; all command bytes not transferred because of Phase change.
100	00011000	Arbitrate, Select, and Command Phase completed.

Receive Command Sequence (Target)

Receive Command Sequence (Target)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
001	00001000	Command Phase set; all bytes not received because of parity error. Check FIFO Flags/ Counter. Check Transfer Complete bit.
001	00011000	Same as above; ATN on.
010	00001000	Command Sequence completed. Check Transfer Complete bit.
010	00011000	Same as above; ATN on.

Command Complete Sequence (Target)

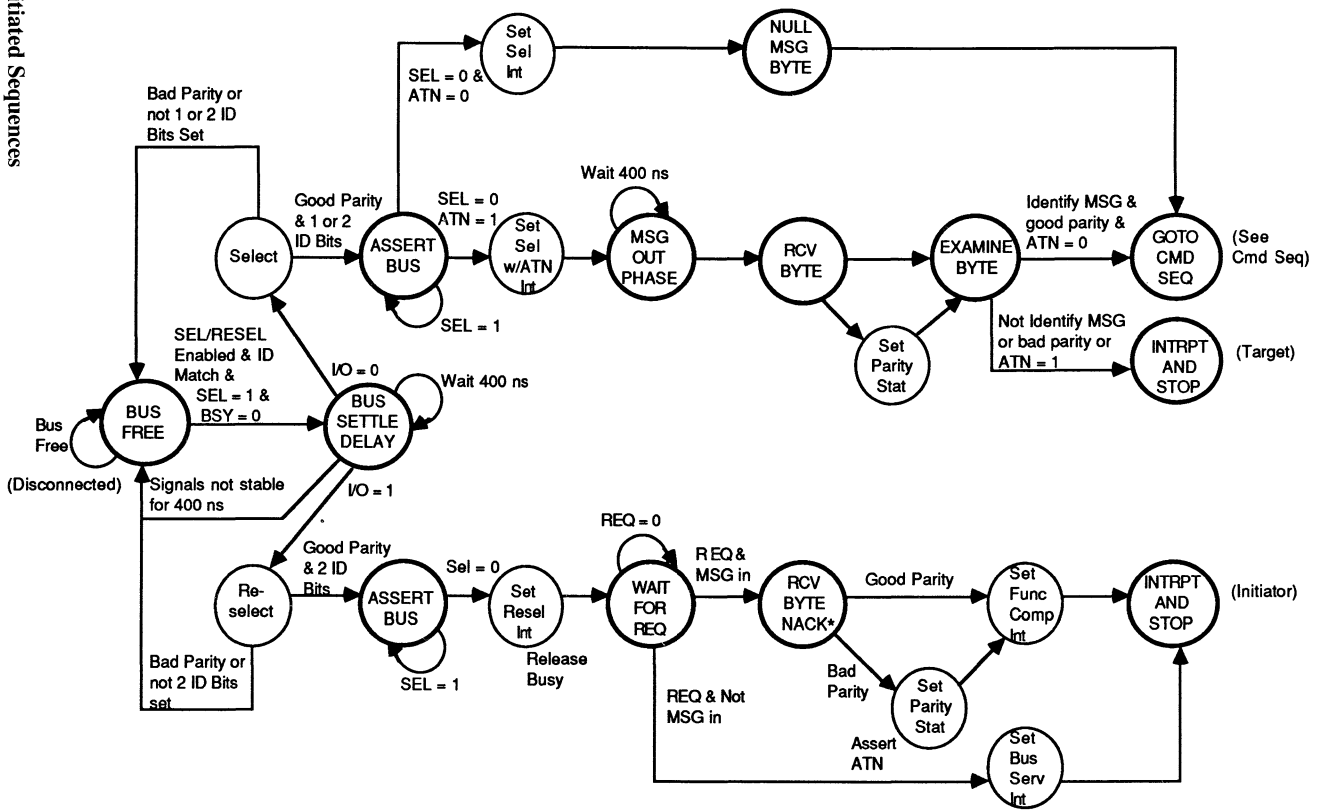
Command Complete Sequence (Target)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00011000	Status phase, sent one byte; ATN on.
001	00011000	Status Phase completed. Message In Phase; sent one byte; ATN on.
010	00001000	Command Complete Sequence completed.

Disconnect Sequence (Target)

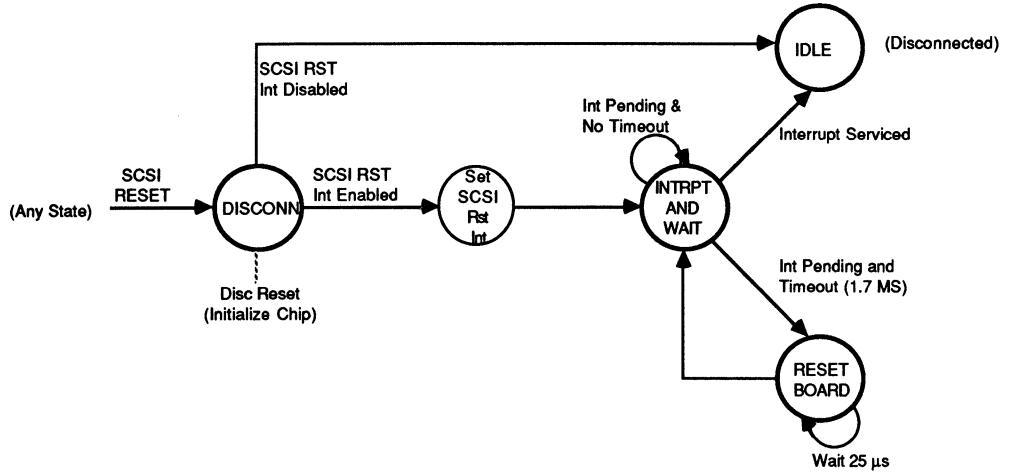
Disconnect Sequence (Target)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00011000	Message In Phase; sent one byte; ATN on.
001	00011000	Message In Phase; sent second byte; ATN on.
010	00101000	Disconnect Sequence completed.

Terminate Sequence (Target)

Terminate Sequence (Target)		
SEQ STEP 210	INT REG 76543210	DESCRIPTION
000	00011000	Status Phase, sent one byte; ATN on.
001	00011000	Status complete. Message In Phase, sent one byte; ATN on.
010	00101000	Terminate Sequence Completed.



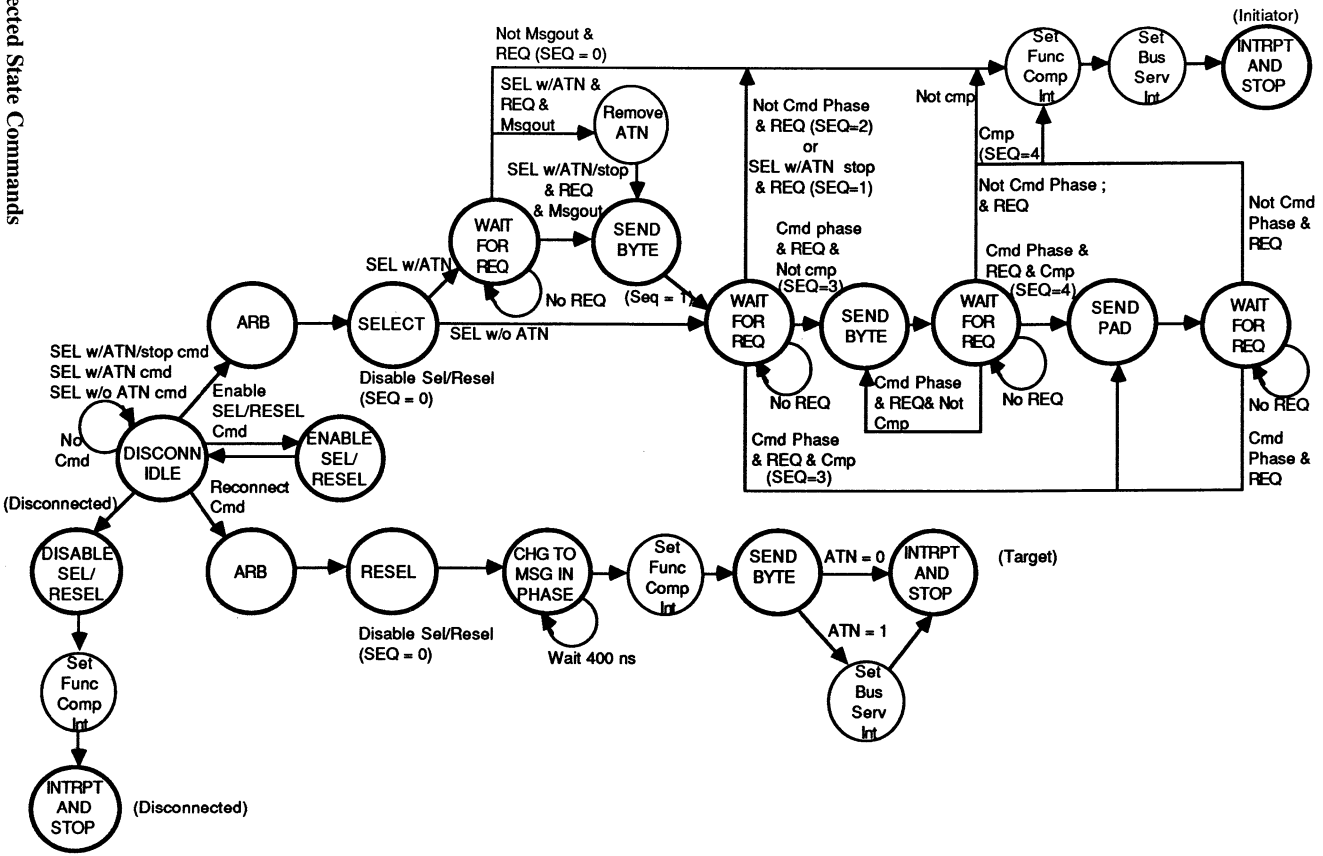
*Note: NACK = ACK remains asserted until Message Accepted issued.

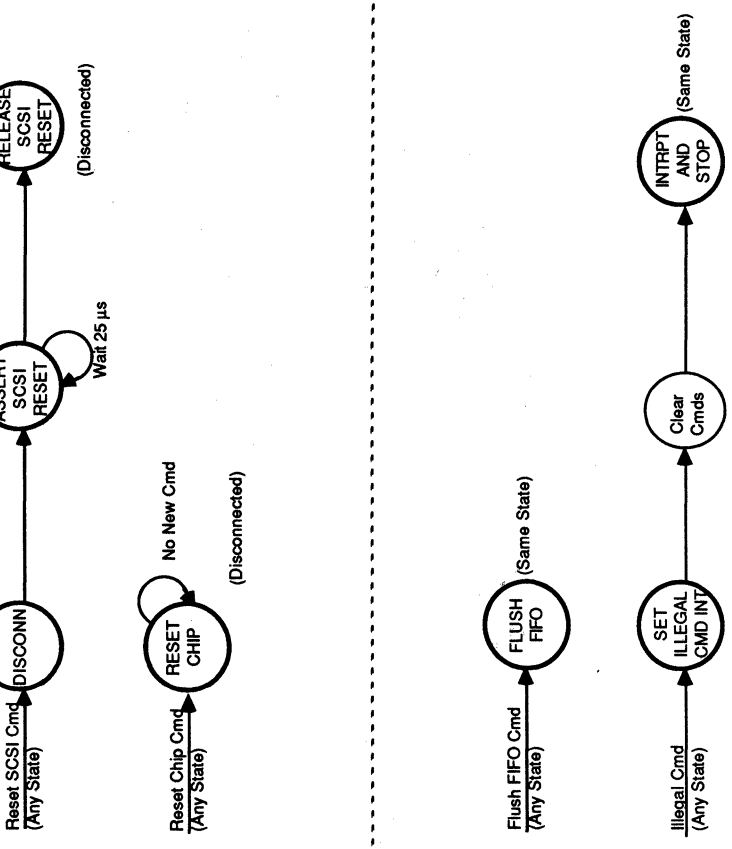


SCSI Reset

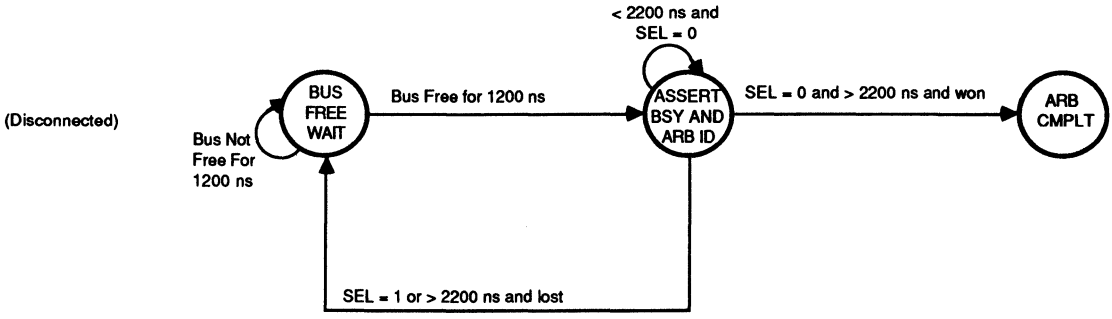
Bus Initiated Sequences

Disconnected State Commands

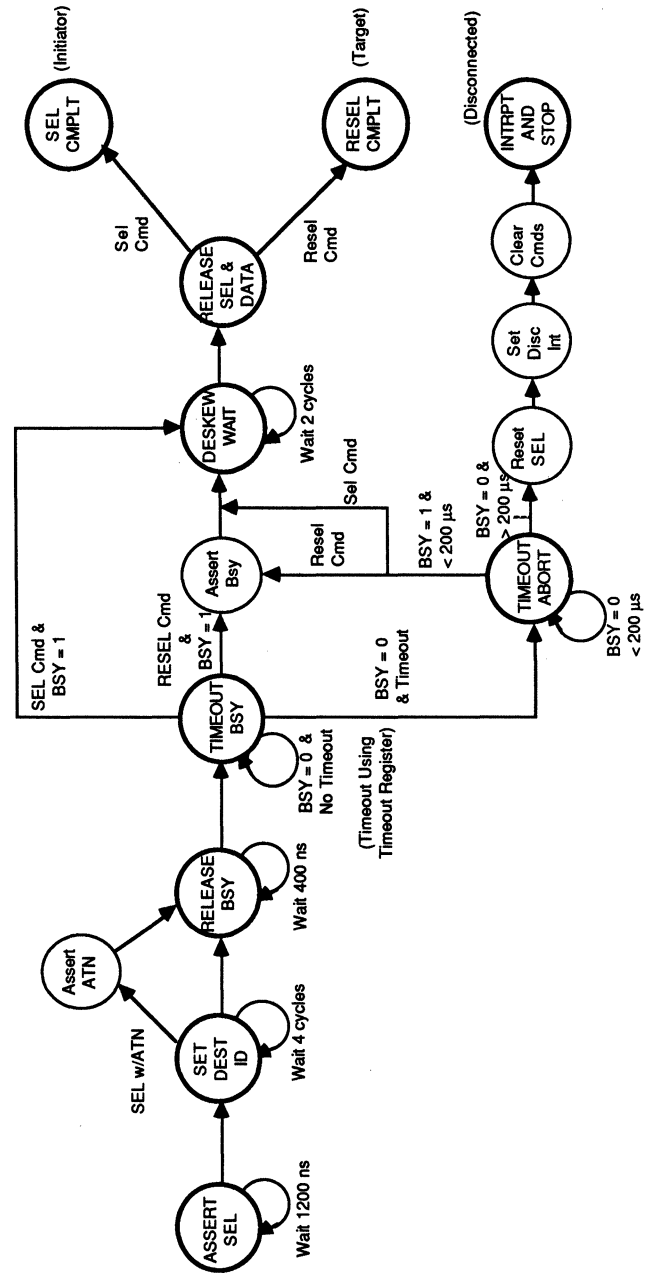




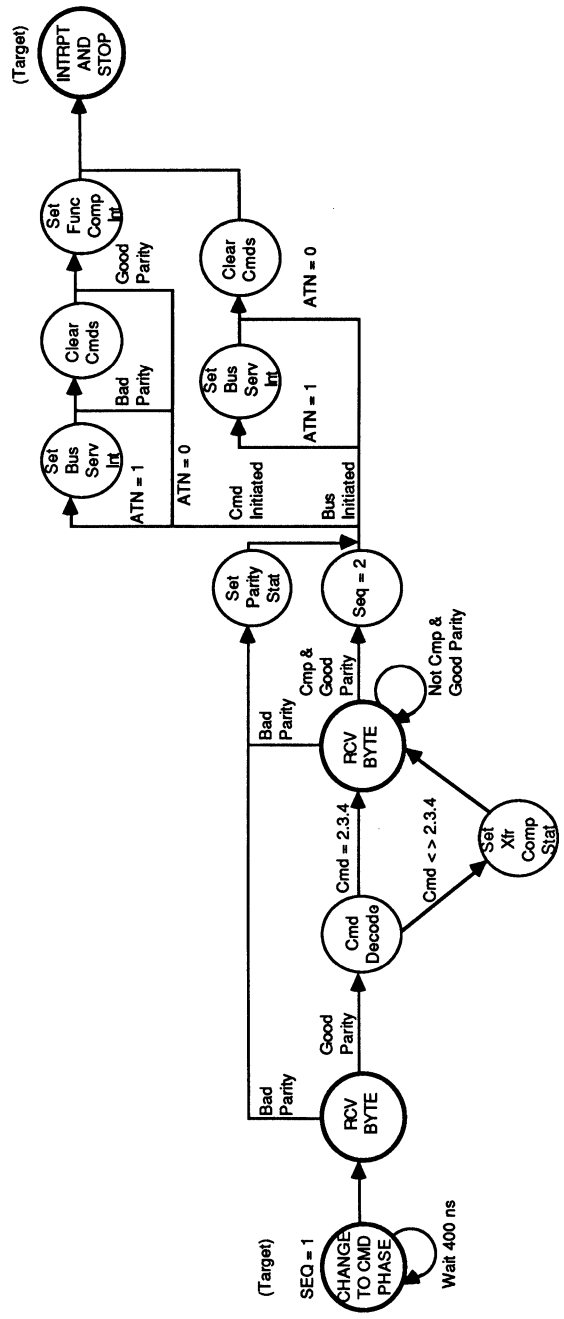
Disconnected State Commands (cont.)


(Disconnected)
BUS FREE WAIT
Bus Not Free For 1200 ns
Bus Free for 1200 ns
< 2200 ns and SEL = 0
ASSERT BSY AND ARB ID
SEL = 0 and > 2200 ns and won
ARB CMPLT
SEL = 1 or > 2200 ns and lost

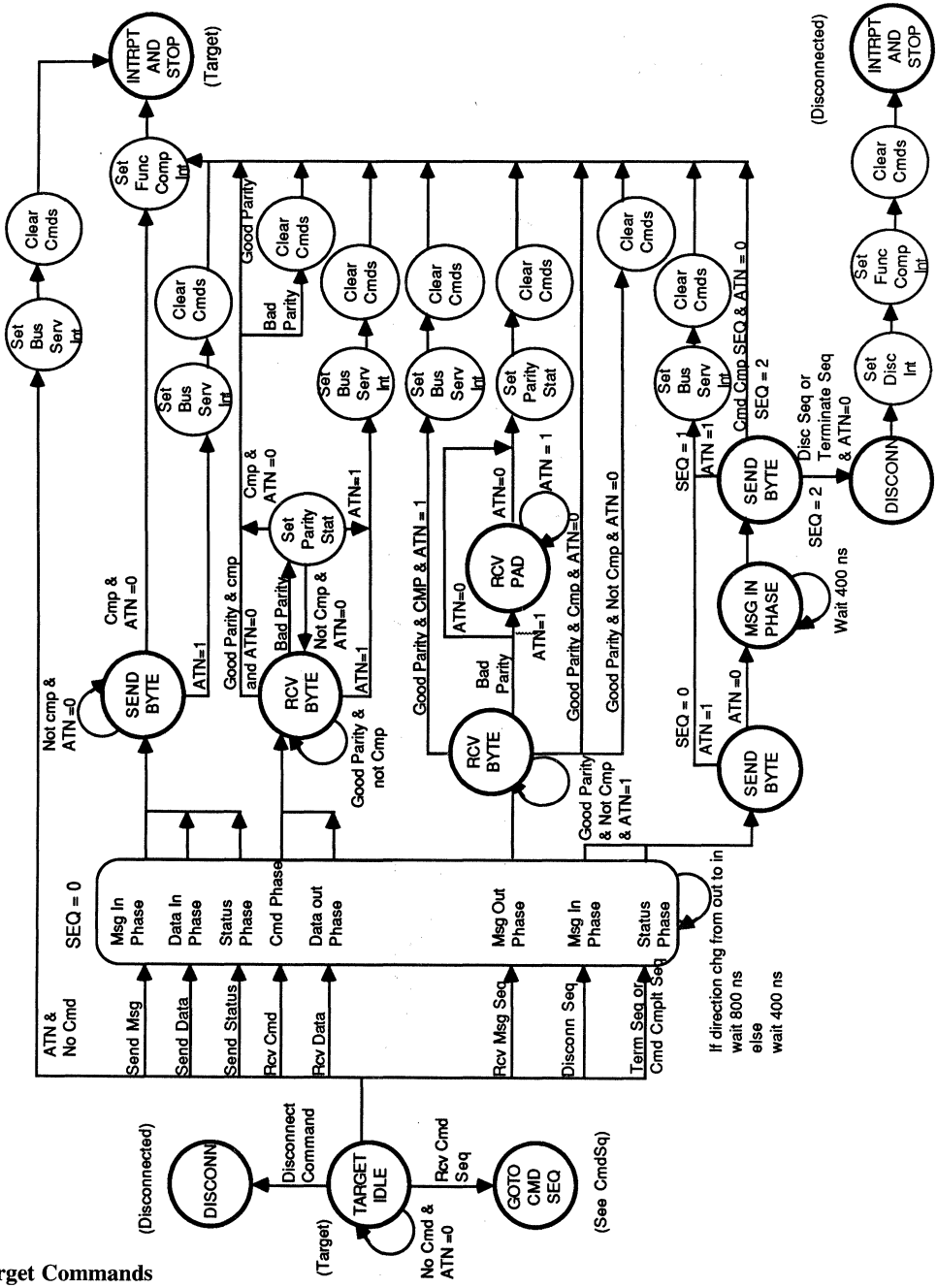
Arbitrate



Select/Reselect

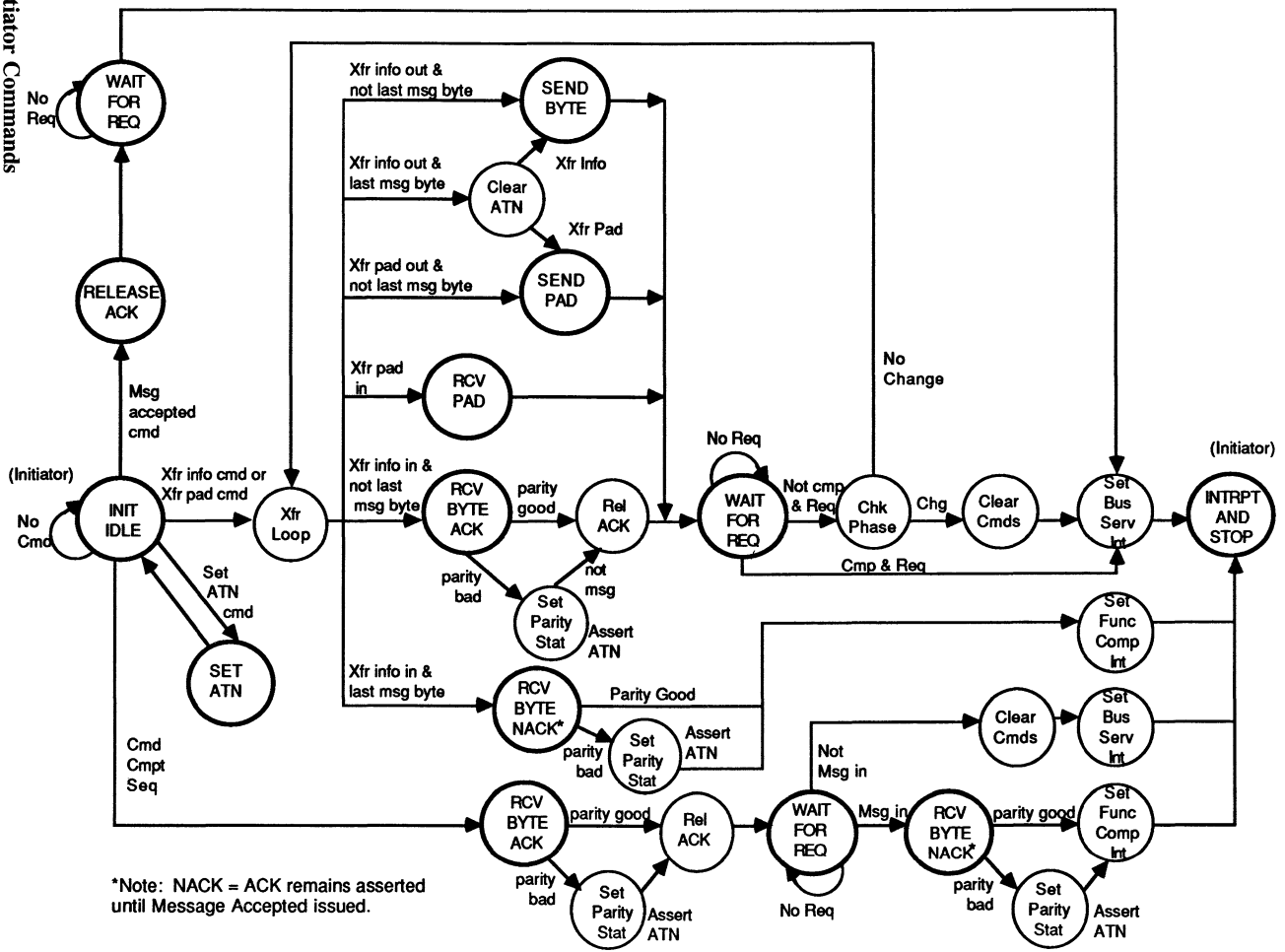


Command Sequence



Target Commands

Initiator Commands



Electrical Characteristics

D.C. Characteristics

Absolute Maximum Stress Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
Tstg	Storage Temperature	−55	150	°C
VDD	Supply Voltage	−0.5	7.0	V
VIN	Input Voltage	VSS − 0.5	VDD + 0.5	V
ESD*	Electrostatic Discharge (All except SCSI pins)	−4000	4000	V
ESD*	Electrostatic Discharge (SCSI pins)	−7000	7000	V

*Test using the human body model—100pF at 1.5kΩ

Operating Conditions

SYMBOL	PARAMETER	MIN	MAX	UNIT
VDD	Supply Voltage	4.75	5.25	V
IDD*	Supply Current (Static IDD)	0	10	mA
IDD	Supply Current (Dynamic IDD)	0	50	mA
Ta	Operating Free-Air	0	70	°C

* Static IDD refers to all inputs at VSS, all outputs floating, and all bidirectional pins configured as inputs.

Data Bus Lines DB0 - DB7

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VIL	Input Low Voltage	VSS − 0.5	0.8	V	
IIL	Input Leakage Current	−400	10	μA	0 < VIN < VDD
VOH	Output High Voltage	2.4	VDD	V	IOH = −400 μA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA

CLK, RESET, A3-A0

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VIL	Input Low Voltage	VSS - 0.5	0.8	V	
IIL	Input Leakage Current	−10	10	μA	0 < VIN < VDD

CS/, RD/, WR/, ACK/, DIFFM

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIH	Input High Voltage	2.2	VDD + 0.5	V	
VIL	Input Low Voltage	VSS − 0.5	0.8	V	
IIL	Input Leakage Current	−10	10	μA	0 < VIN < VDD

DREQ, TGS, IGS

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOH	Output High Voltage	2.4	VDD	V	IOH = −400 μA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA
IOZ	Tri-State Leakage Current	−10	10	μA	0 < VOUT < VDD

INT/

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA
IOZ	Tri-State Leakage Current	−10	10	μA	0 < VOUT < VDD

RSTI/, SELI/, BSYI/, ATNI/, MSGI/, CDI/, IOI/, REQI/, ACKI/

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIL	Input Low Voltage	VSS − 0.5	0.8	V	
VIH	Input High Voltage	2.0	VDD + 0.5	V	
IIL	Input Leakage Current	−10	10	μA	0 < VIN < VDD
HST	Hysteresis	200		mV	

RSTO/, SELO/, BSYO/, ATNO/, MSGO/, CDO/, IOO/, REQO/, ACKO/, SDO0-SDO7/, SDOP/

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOL	Output Low Voltage	VSS	0.5	V	IOL = 48 mA
IOZ	Tri-State Leakage Current	−10	10	μA	0 < VOUT < VDD
SFT	Signal Fall Time	8		ns	IOL = 48 mA

RESETO

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOH	Output High Voltage	2.4	VDD	V	IOH = −4 mA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA
IOZ	Tri-State Leakage	−10	10	μA	0 < VOUT < VDD

SDI0-SDI7/

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIL	Input Low Voltage	2.0	VDD + 0.5	V	
VIH	Input High Voltage	VSS - 0.5	0.8	V	
HST	Hysteresis	200		mV	
IIL	Input Leakage Current	−10	10	μA	0 < VIN < VDD
VOH	Output High Voltage	2.4	VDD	V	IOH= −400 μA
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4 mA

A.C. Characteristics

The A.C. characteristics described in this section apply over the voltage range Vdd equal to 4.75 to 5.25 volts and the temperature range 0 to 7 °C. Chip output timing is based on simulation under worst case conditions (4.75 V, 70 °C) and the following pad termination:

SYMBOLS	OUTPUT LOADS	
RESETO, DREQ	50 pf	
DB7...0	85 pf	
TGS, IGS	50 pf	
SDIP/, SDI7/..0	50 pf	
INT/	50 pf	1K pullup
RSTO/, SELO/, BSYO/, ATNO/, MSGO/, CDO/ IOO/, REQO/, ACKO/, SDO7...0/, SDOP/	200 pf	110 pullup 165 pulldown

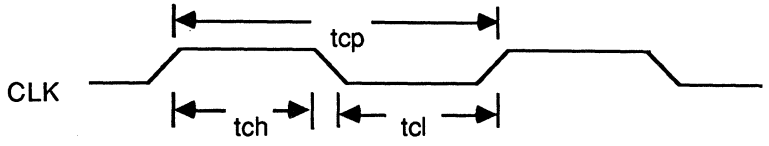
System Interface

All timings in this specification are taken from the 10% & 90% points with respect to the specified VOL & VOH of the waveforms.

Clock Interface

Symbol	Description	Min	Max	Unit
tcp	Clock period	40	100	ns
fcpa	Clock frequency, async	10*	25	MHz
fcps	Clock frequency, sync	13*	25	MHz
tch	Clock high	.45 tcp	.55 tcp	ns
tcl	Clock low	.45 tcp	.55 tcp	ns

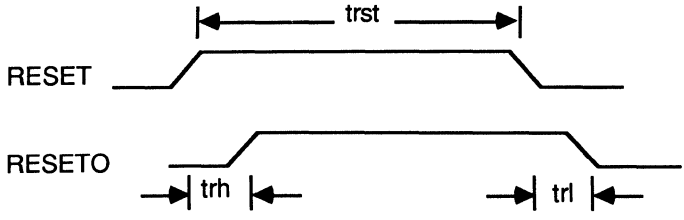
*Note: These minimum numbers required to comply with ANSI SCSI specification.



Reset Input

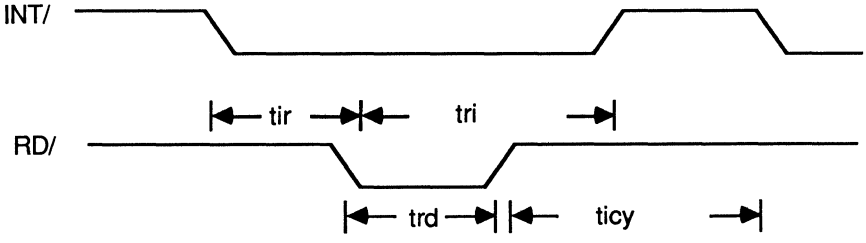
Symbol	Description	Min	Max	Unit
trst	Reset pulse width	*2tcp or 200		ns
trh	Reset high to RESETO high		50	ns
trl	Reset low to RESETO low		50	ns

* Whichever is longer



Interrupt Output

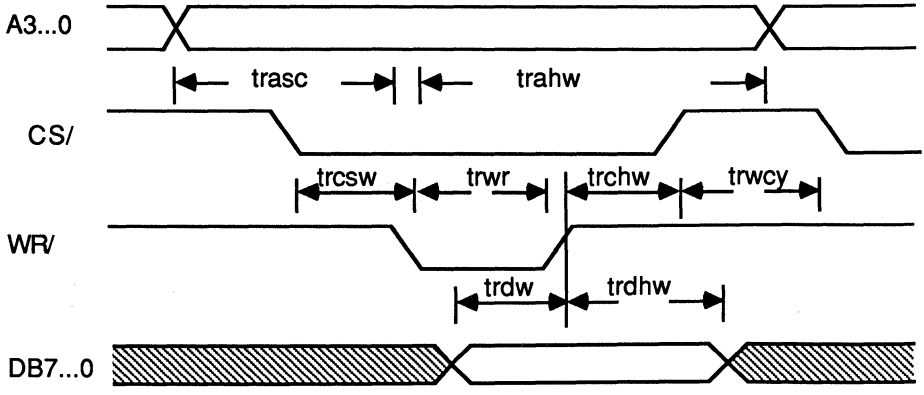
Symbol	Description	Min	Max	Unit
tir	INT/ to RD/	2tcp		ns
trd	RD/ pulse width	50		ns
tri	RD/ low to INT/ high		70 +3.6tcp	ns
ticy	RD/ high to INT/ low	1.4 tcp		ns



Register Interface

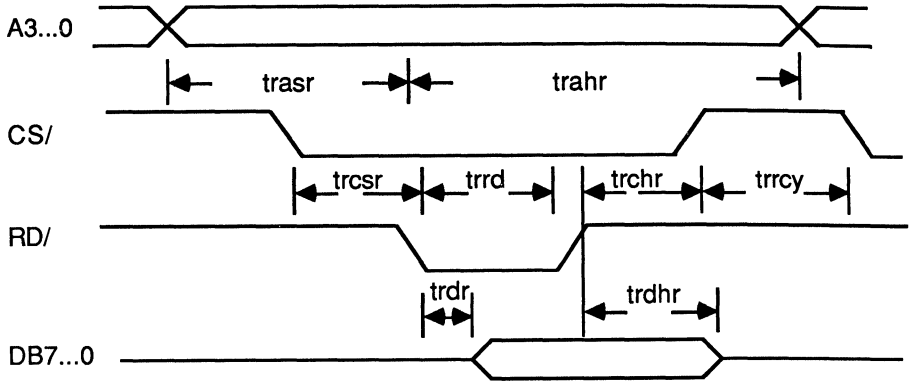
Register Write

Symbol	Description	Min	Max	Unit
trasc	Address setup to WR/	12		ns
trcsw	CS/ setup to WR/	12		ns
trwr	WR/ pulse width	40		ns
trdw	Data to WR/ high	20		ns
trahw	Address hold time from WR/	30		ns
trdhw	Data hold time	10		ns
trchw	WR/ high to CS/ high	12		ns
trwcy	CS/ high to CS/ low	60		ns



Register Read

Symbol	Description	Min	Max	Unit
trasr	Address setup to RD/	12		ns
trcsr	CS/ setup to RD/	12		ns
trrd	RD/ pulse width	2tcp		ns
trdr	RD/ to data		50	ns
trahr	Address hold time from RD/	30		ns
trdhr	Data hold time	2	50	ns
trchr	RD/ high to CS/ high	12		ns
trrcy	CS/ high to CS/ low	30		ns

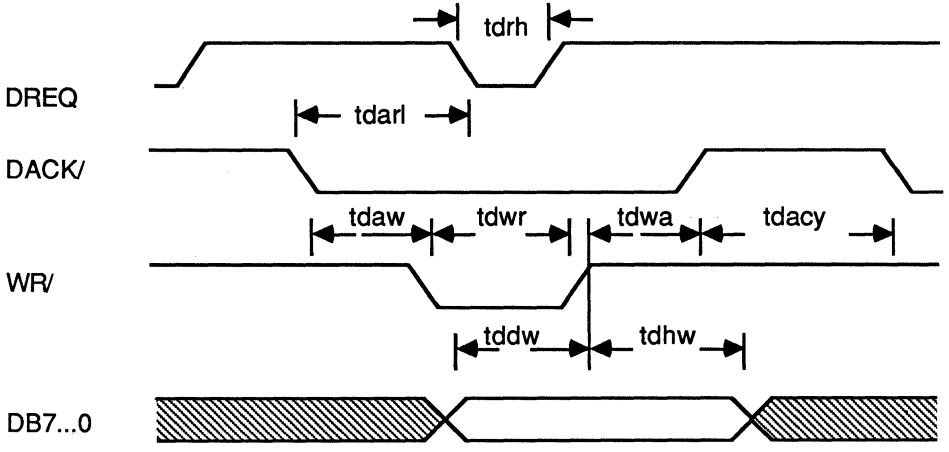


DMA Interface

DMA Write

Symbol	Description	Min	Max	Unit
tdarl *(1)	DACK/ low to DREQ low	0	65	ns
tdaw	DACK/ low to WR/ low	0		ns
tdwr	WR/ pulse width	40		ns
tddw	Data to WR/ high	20		ns
tdwa *(1,2)	WR/ high to DACK/ high	12		ns
tdhw	Data hold time	10		ns
tdacy *(1,2)	DACK/ high to DACK/ low	12		ns
tdrh	DREQ low to DREQ high	0		ns

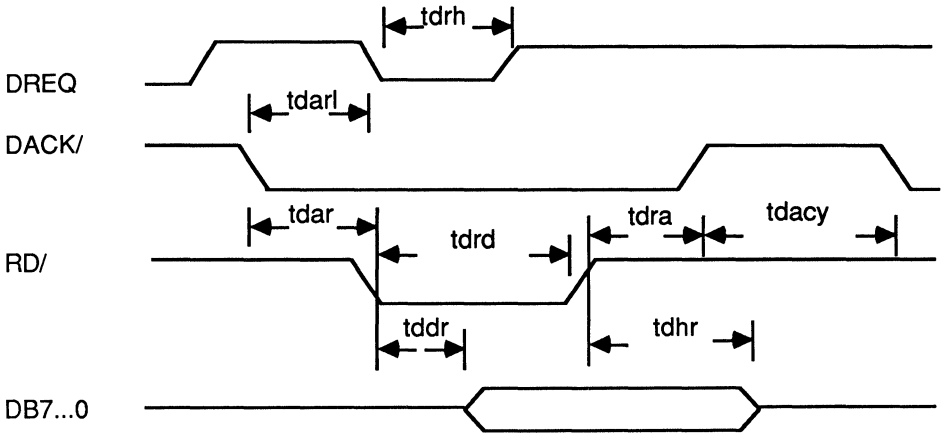
*Notes: 1. If tdacy + tdwa ≥ 30 ns, tdarl = 60 ns maximum
2. If tdacy ≥ 20 ns, tdwa = 0 ns minimum



DMA Read

Symbol	Description	Min	Max	Unit
tdarl	DACK/ low to DREQ low	0	65	ns
tdar *(1,2)	DACK/ low to RD/ low	12		ns
tdrd	RD/ pulse width	50		ns
tddr *(1)	RD/ to data		50	ns
tdra	RD/ high to DACK/ high	0		ns
tdhr	Data hold time	2	50	ns
tdacy *(1,2)	DACK/ high to DACK/low	12		ns
tdrh	DREQ low to DREQ high	0		ns

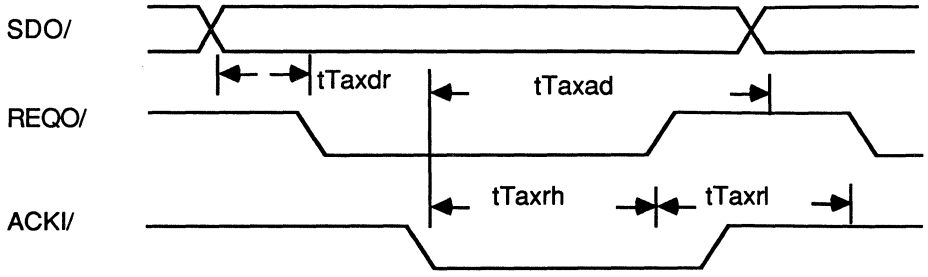
*Notes: 1. If tdacy + tdar ≥ 30 ns, tddr = 45 ns maximum
2. If tdacy ≥ 20 ns, tdar = 0 ns minimum



Target Asynchronous Transfers

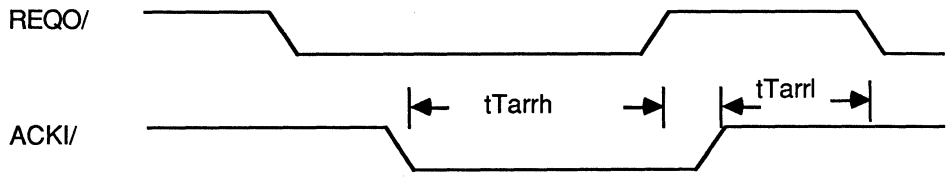
Target Asynchronous Send

Symbol	Description	Min	Max	Unit
tTaxdr	Data to REQO/ low	55		ns
tTaxrh	ACKI/ low toREQO/ high		49	ns
tTaxad	ACKI/ low to data (FIFO bottom full)		83	ns
tTaxrl	ACKI/ high to REQO/ low (data already set up)		54	ns



Target Asynchronous Receive

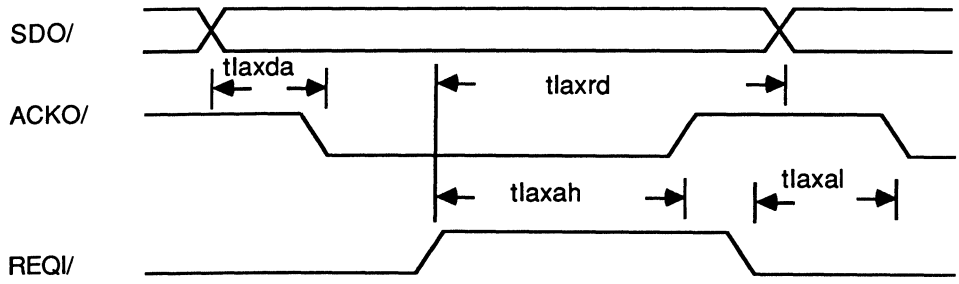
Symbol	Description	Min	Max	Unit	
tTarrh	ACKI/ low to REQO/ high			50	ns
tTarrl	ACKI/ high to REQO/ low (FIFO not full)			55	ns



Initiator Asynchronous Transfers

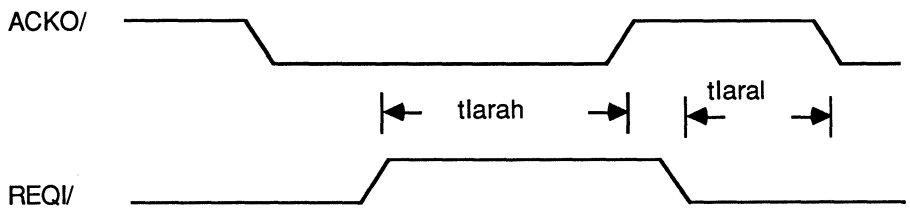
Initiator Asynchronous Send

Symbol	Description	Min	Max	Unit
tlaxda	Data to ACKO/ low	65		ns
tlaxah	REQI/ high to ACKO/ high		46	ns
tlaxrd	REQI/ high to data (FIFO bottom full)		80	ns
tlaxal	REQI/ low to ACKO/ low (data already setup)		55	ns



Initiator Asynchronous Receive

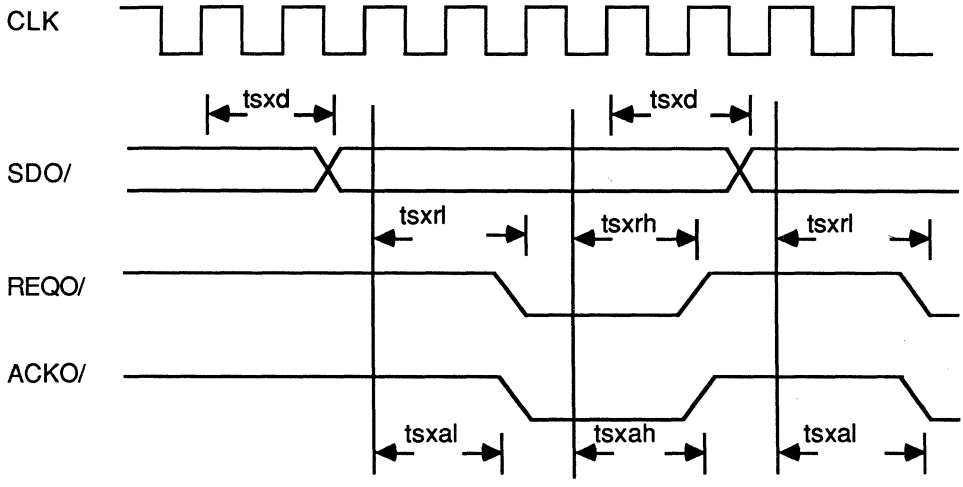
Symbol	Description	Min	Max	Unit
tlarah	REQI/ high to ACKO/ high		50	ns
tlaral	REQI/ low to ACKO/ low (FIFO not full)		68	ns



Synchronous Transfers

Target and Initiator Synchronous Transmit

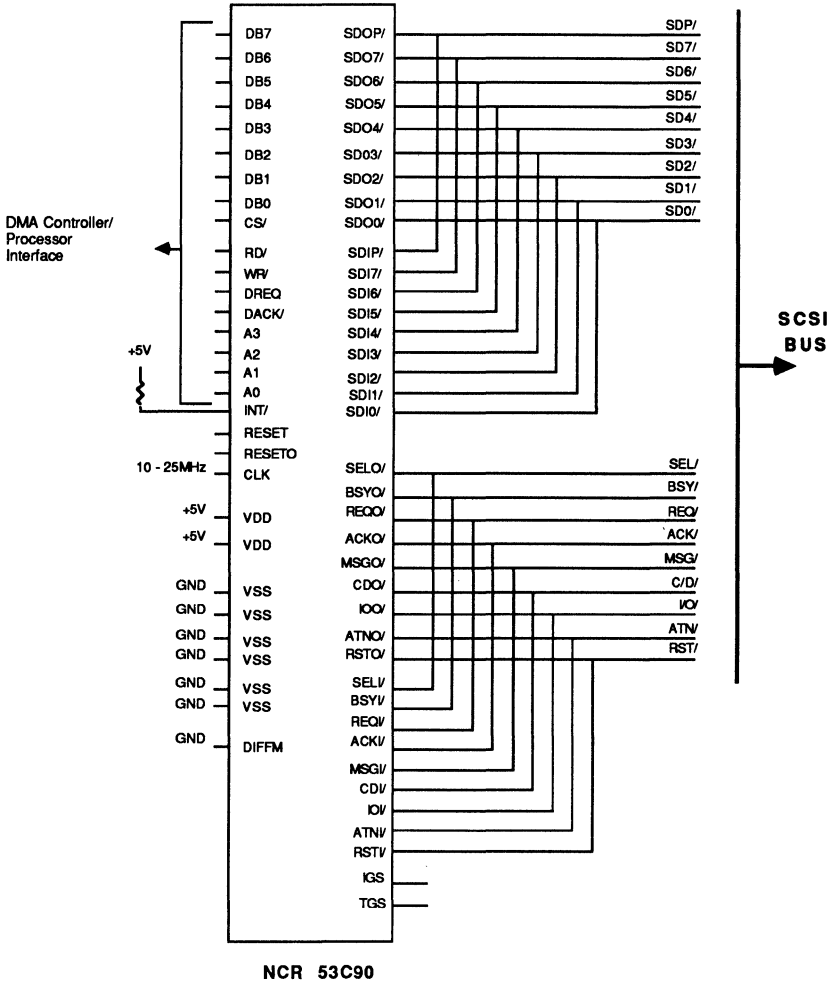
Symbol	Description	Min	Max	Unit
tsxd	Data from CLK high	20	84	ns
tsxrl	REQO/ low from CLK high	15	64	ns
tsxrh	REQO/ high from CLK low	17	69	ns
tsxal	ACKO/ low from CLK high	15	64	ns
tsxah	ACKO/ high from CLK low	17	70	ns



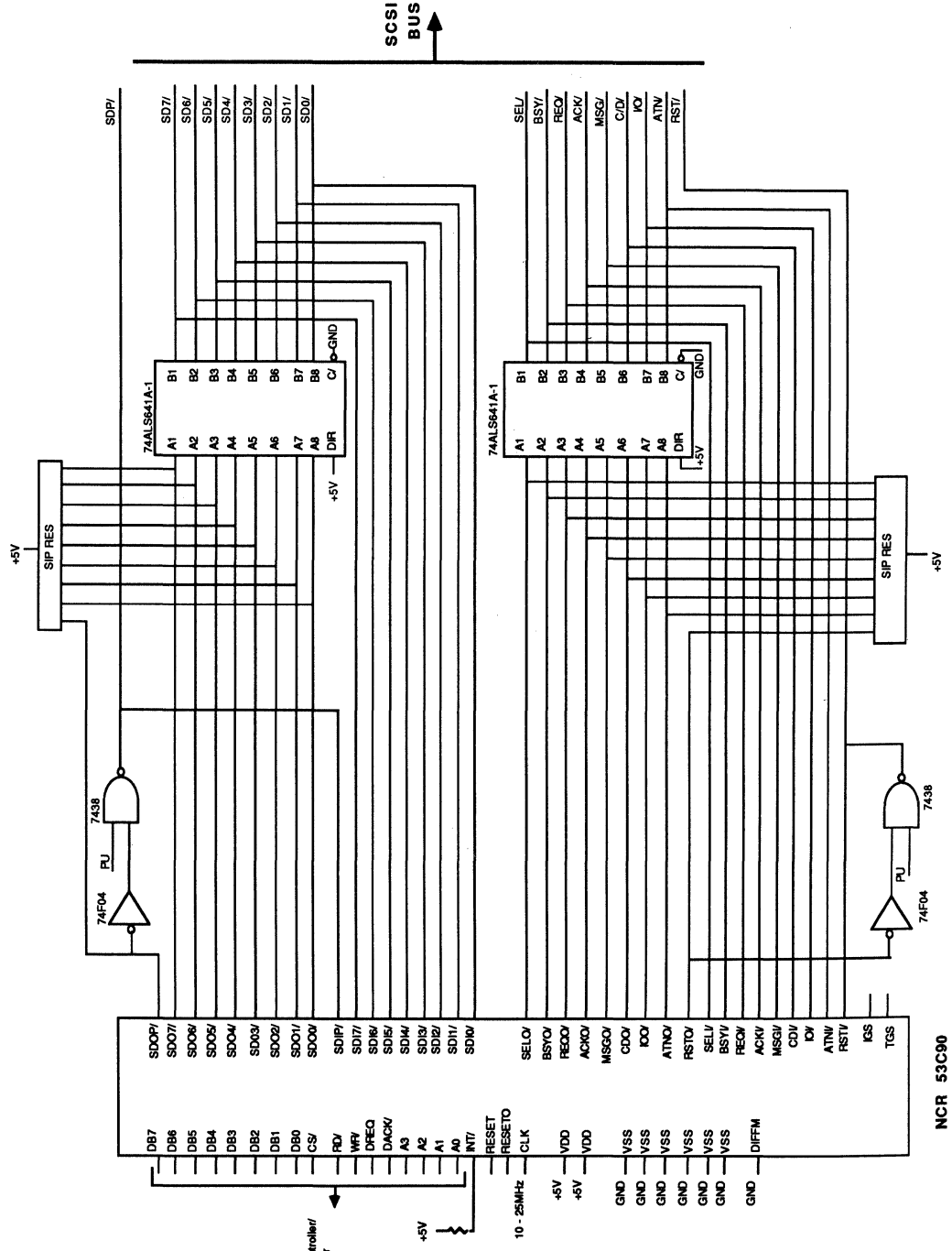
Applications

Single-ended Mode

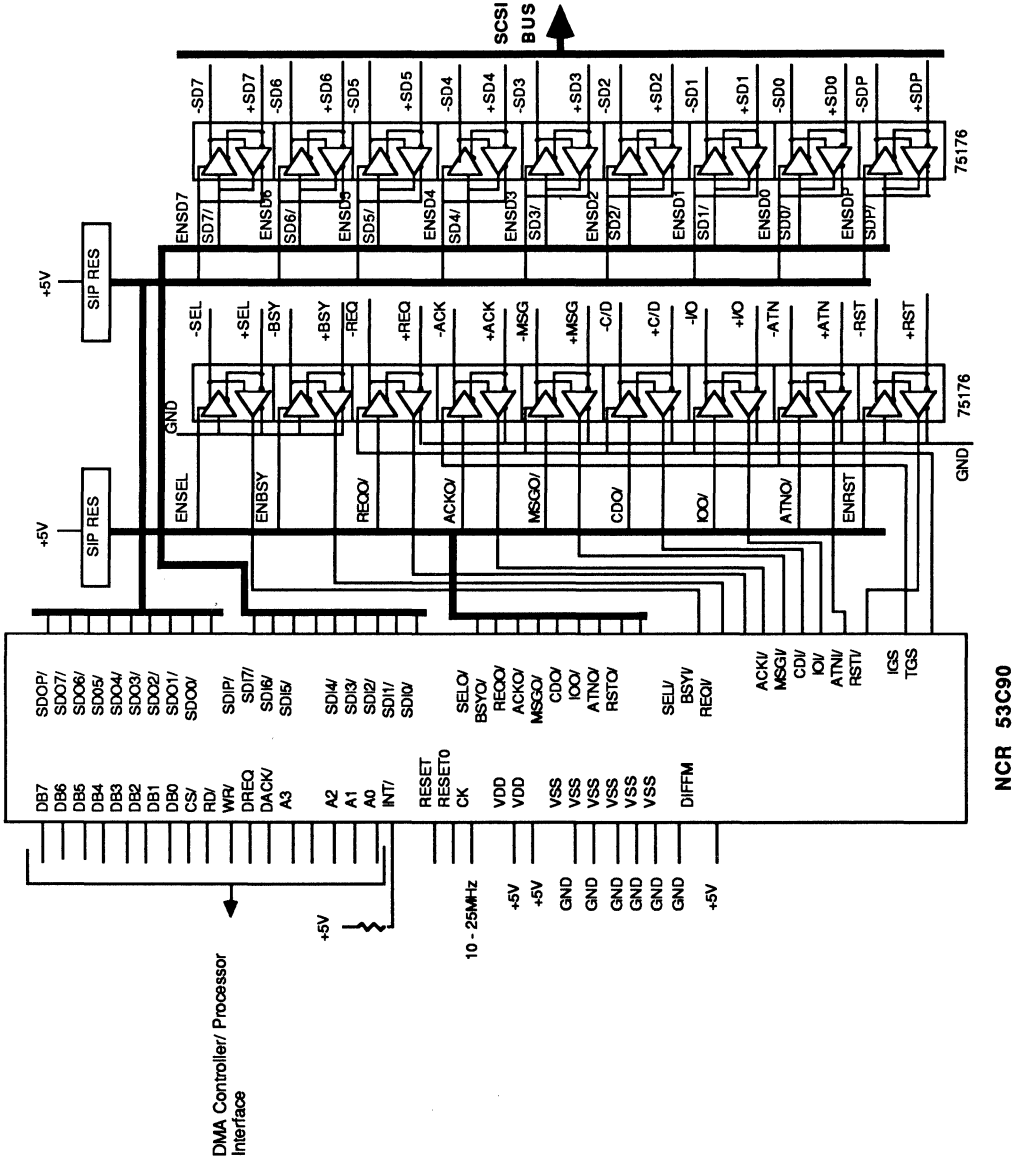
Single-ended Mode without External Drivers



Single-ended Mode with External Drivers



Differential Mode



Appendices

Appendix A - Register Summary

#	ESP Read Registers	#	ESP Write Registers
0	Transfer counter lo(LSB)	0	Transfer count lo (LSB)
1	Transfer counter hi (MSB)	1	Transfer count hi (MSB)
2	FIFO	2	FIFO
3	Command	3	Command
4	Status	4	S/R bus ID
5	Interrupt	5	S/R timeout
6	Sequence step	6	Sync period
7	FIFO flags	7	Sync offset
8	Configuration	8	Configuration
9	NCR Reserved	9	Clock factor
A	NCR Reserved	A	Test

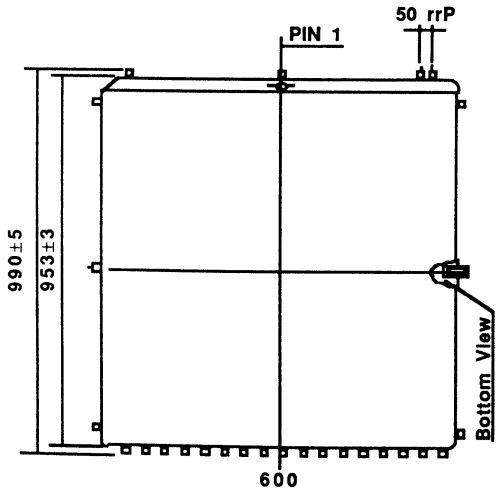
Command Register (RW3) (Bit 7 Set = DMA Mode)

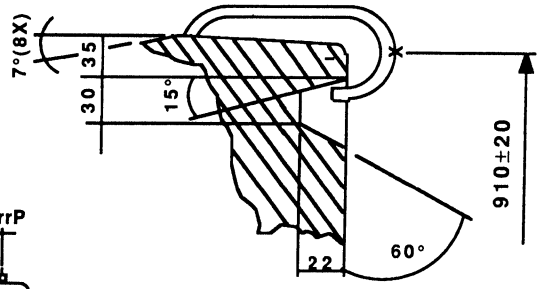
Misc Cmds	Initiator Cmds	Target Cmds
00 80 NOP 01 81 Flush FIFO 02 82 Reset chip 03 83 Reset SCSI	10 90 Transfer info 11 91 Cmd comp seq 12 92 Accept msg 18 98 Transfer pad 1A 9A Set ATN	20 A0 Send msg 21 A1 Send status 22 A2 Send data 23 A3 Disconnect seq 24 A4 Terminate seq 25 A5 Cmd comp seq 27 A7 Disconnect 28 A8 Rcv msg seq 29 A9 Rcv cmd 2A AA Rcv data 2B AB Rcv cmd seq
Disconnected Cmds		
40 C0 Reconnect 41 C1 Sel w/o ATN 42 C2 Sel w/ATN 43 C3 Sel w/ATN/stop 44 C4 Enable S/R 45 C5 Disable S/R		

STATUS (RO4)	INT (RO5)	CONFIG (RW8)
0 I/O 1 C/D 2 MSG 3 Xfr complete 4 Xfr count 0 5 Parity error 6 Gross error 7 NCR Reserved	0 Selected 1 Selected w/ATN 2 Reselected 3 Func complete 4 Bus service 5 Disconnect 6 Illegal cmd 7 SCSI reset	0 Bus ID 0 1 Bus ID 1 2 Bus ID 2 3 Chip Test Mode 4 Parity enable 5 Parity test mode 6 SCSI rst int dis 7 Slow cable
SEQ STEP (RO6)	**TEST (WOA)**	
0 Seq step 0 1 Seq step 1 2 Seq step 2	0 Target Mode 1 Initiator Mode 2 Tri-state Mode	

Appendix B - 68 Pin PLCC Mechanical Drawing

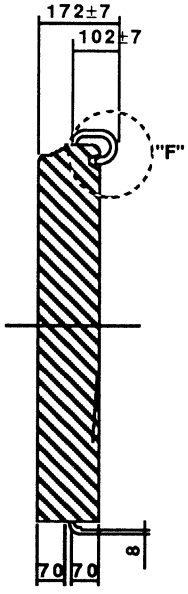
The Units are 1/1000 inches.

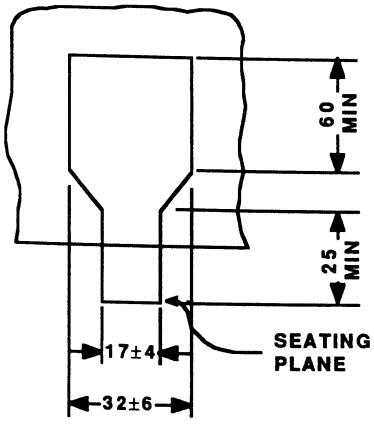




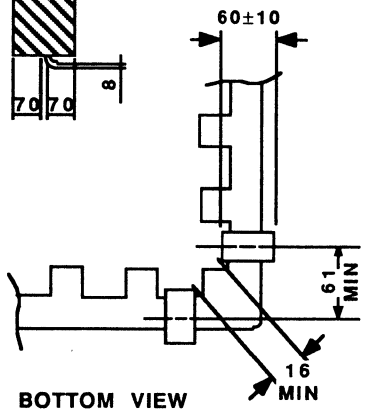
DETAIL OF "F"

X: SURFACE MOUNT POINT





SIDE VIEW



BOTTOM VIEW

5380/C80 SCSI INTERFACE

General Description

The NCR 5380 SCSI interface device is a 40 pin NMOS device designed to accommodate the Small Computer Systems Interface (SCSI) as defined by the ANSI X3T9.2 committee. The NCR 5380 operates in both the initiator and target roles and can therefore be used in host adapter, host port and formatter designs. This device supports arbitration, including reselection. Special high-current open collector output drivers, capable of sinking 48mA at 0.5V, allow for direct connection to the SCSI bus. Differential pair operation is supported using a 48 pin version of this part, designated the NCR 5381 (refer to Appendix A4).

SCSI Interface

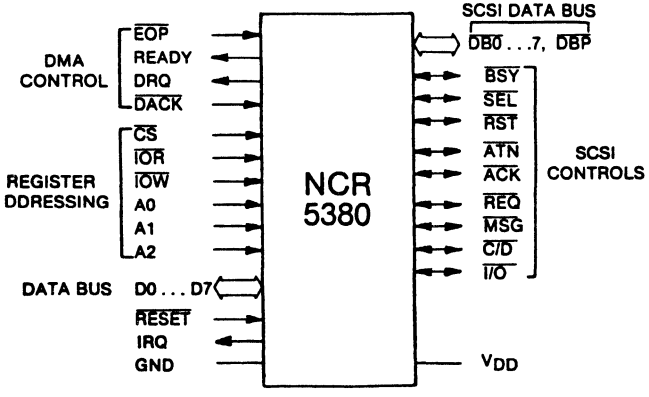
- Asynchronous, interface to 1.5 MBPS
- Supports initiator and target roles
- Parity generation w/optional checking
- Supports arbitration
- Direct control of all bus signals
- High current outputs drive SCSI bus directly

The NCR 5380 communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory mapped I/O. Minimal processor intervention is required for DMA transfers because the 5380 controls the necessary handshake signals. The NCR 5380 interrupts the MPU when it detects a bus condition that requires attention. Normal and block mode DMA is provided to match many popular DMA controllers. **A CMOS version of the NCR 5380, the NCR 53C80 is available in a 44-pin PLCC or a 48-pin DIP. Refer to Appendix A5 for information on the NCR 53C80.**

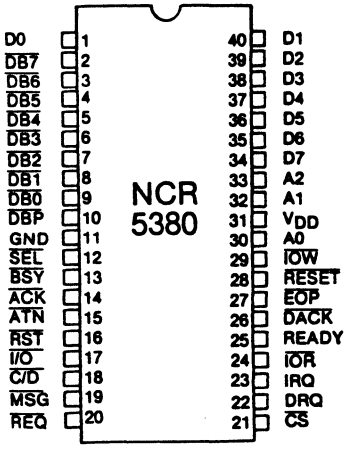
MPU Interface

- Memory or I/O mapped interface
- DMA or programmed I/O
- Normal or block mode DMA
- Optional MPU interrupts

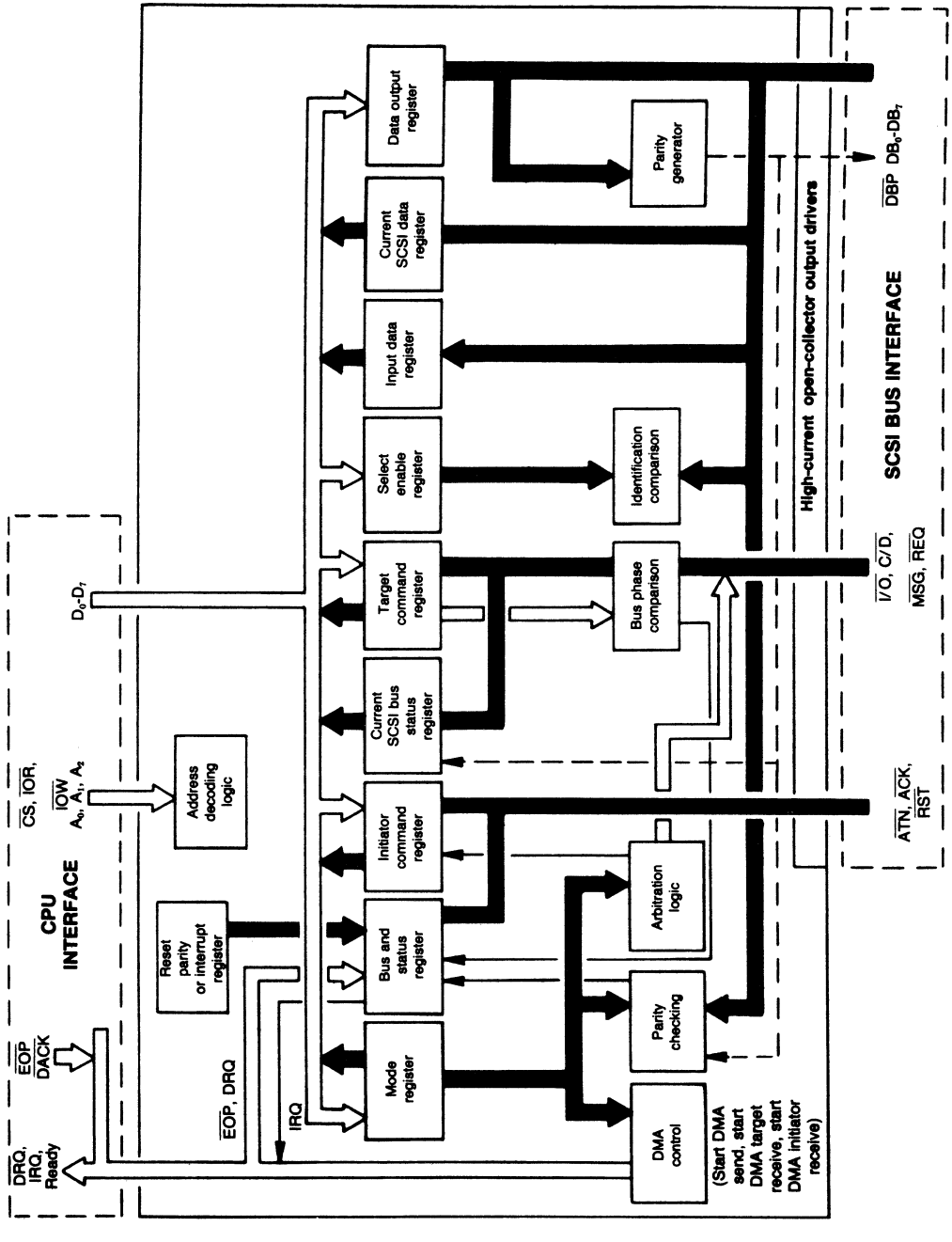
Functional Pin Grouping



Pinout



Block Diagram



Pin Description

Microprocessor Interface Signals

Pin Name	Pin #	Description
A0, A1, A2	30, 32, 33	INPUTS These signals are used with $\overline{CS}$, $\overline{IOR}$ or $\overline{IOW}$ to address all internal registers.
$\overline{CS}$	21	INPUT Chip Select enables a read or write of the internal register selected by A0, A1 and A2. $\overline{CS}$ is an active low signal.
$\overline{DACK}$	26	INPUT DMA Acknowledge resets DRQ and selects the data register for input or output data transfers. $\overline{DACK}$ is an active low signal.
DRQ	22	OUTPUT DMA Request indicates that the data register is ready to be read or written. DRQ occurs only if DMA mode is true in the Command Register. It is cleared by $\overline{DACK}$.
D0...D7	1,40...34	BI-DIRECTIONAL, TRI-STATE Microprocessor data bus active high
$\overline{EOP}$	27	INPUT The End of Process signal is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte will be transferred but no additional bytes will be requested.
$\overline{IOR}$	24	INPUT I/O Read is used to read an internal register selected by $\overline{CS}$ and A0, A1 and A2. It also selects the Input Data Register when used with $\overline{DACK}$. $\overline{IOR}$ is active low.
$\overline{IOW}$	29	INPUT I/O Write is used to write an internal register selected by $\overline{CS}$ and A0, A1 and A2. It also selects the Output Data Register when used with $\overline{DACK}$. $\overline{IOW}$ is active low.
IRQ	23	OUTPUT Interrupt Request alerts a microprocessor of an error condition or an event completion.
READY	25	OUTPUT Ready can be used to control the speed of block mode DMA transfers. This signal goes active to indicate the chip is ready to send/receive data and remains false after a transfer until the last byte is sent or until the DMA Mode bit is reset.
$\overline{RESET}$	28	INPUT Reset clears all registers. It does not force the SCSI signal $\overline{RST}$ to the active state. $\overline{RESET}$ is an active low signal.

Power Signals

Pin Name	Pin #	Description
VDD	31	+5 VOLTS
GND	11	GROUND

SCSI Interface Signals

The following signals are all bi-directional, active low, open collector signals. With 48 mA sink capability, all pins interface directly with the SCSI bus.

Pin Name	Pin #	Description
$\overline{ACK}$	14	Driven by an initiator, $\overline{ACK}$ indicates an acknowledgment for a REQ/ACK data transfer handshake. In the target role, $\overline{ACK}$ is received as a response to the $\overline{REQ}$ signal.
$\overline{ATN}$	15	Driven by an initiator, $\overline{ATN}$ indicates an attention condition. This signal is received in the target role.
$\overline{BSY}$	13	This signal indicates that the SCSI bus is being used and can be driven by both the initiator and the target device.
$\overline{CD}$	18	A signal driven by the target, $\overline{CD}$ indicates Control or Data information is on the data bus. This signal is received by the initiator.
$\overline{IO}$	17	$\overline{IO}$ is a signal driven by a target which controls the direction of data movement on the SCSI bus. True indicates input to the initiator. This signal is also used to distinguish between Selection and Reselection phases.
$\overline{MSG}$	19	$\overline{MSG}$ is a signal driven by the target during the Message phase. This signal is received by the initiator.
$\overline{REQ}$	20	Driven by a target, $\overline{REQ}$ indicates a request for a REQ/ACK data transfer handshake. This signal is received by the initiator.
$\overline{RST}$	16	The $\overline{RST}$ signal indicates an SCSI bus RESET condition.
$\overline{DB0}$...$\overline{DB7}$ $\overline{DBP}$	9...2 10	These eight data bits ($\overline{DB0}$-$\overline{DB7}$) plus a parity bit ($\overline{DBP}$) form the data bus. $\overline{DB7}$ is the most significant bit and has the highest priority during the Arbitration phase. Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during arbitration.
$\overline{SEL}$	12	$\overline{SEL}$ is used by an initiator to select a target or by a target to reselect an initiator.

Electrical Characteristics

OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	MAX	UNITS
Supply Voltage	V_{DD}	4.75	5.25	Volts
Supply Current	I_{DD}		145	mA.
Ambient Temperature	T_A	0	70	°C

INPUT SIGNAL REQUIREMENTS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
High-level, Input V_{IH}		2.0	5.25	Volts
Low-level, Input V_{IL}		-0.3	0.8	Volts
SCSI BUS pins 2 . . . 20				
High-level Input Current, I_{IH}	V_{IH} = 5.25 V		50	µa.
Low-level Input Current, I_{IL}	V_{IL} = 0 Volts		-50	µa.
All other pins				
High-level Input Current, I_{IH}	V_{IH} = 5.25 V		10	µa.
Low-level Input Current, I_{IL}	V_{IL} = 0 Volts		-10	µa.

OUTPUT SIGNAL REQUIREMENTS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
SCSI BUS pins 2 . . . 20				
Low-level Output V_{OL}	V_{DD} = 4.75 V I_{OL} = 48.0mA.		0.5	Volts
All other pins				
High-level Output V_{OH}	V_{DD} = 4.75 V I_{OH} = -3.0mA.	2.4		Volts
Low-level Output V_{OL}	V_{DD} = 4.75 V I_{OL} = 7.0 mA.		0.5	Volts

PRELIMINARY

Notice: This is not a final specification.
Some parametric limits are subject to change.

Internal Registers

General

The NCR 5380 SCSI Interface Device appears as a set of eight registers to the controlling CPU. By reading and writing the appropriate registers, the CPU may initiate any SCSI bus activity or may sample and assert any signal on the SCSI bus. This allows the user to implement all or portions of the SCSI protocol in software. These registers are read (written) by activating $\overline{CS}$ with an address on A2-A0 and then issuing an $\overline{IOR}$ ($\overline{IOW}$) pulse. This section describes the operation of the internal registers.

SCSI signal names are used to describe the contents of these internal registers. Even though the bus is active low a one (1) is used to indicate signal assertion and a zero (0) is used to indicate the non-asserted or inactive state.

Address A2	A1	A0	R/W	Register Name
0	0	0	R	Current SCSI Data
0	0	0	W	Output Data
0	0	1	R/W	Initiator Command
0	1	0	R/W	Mode
0	1	1	R/W	Target Command
1	0	0	R	Current SCSI Bus Status
1	0	0	W	Select Enable
1	0	1	R	Bus and Status
1	0	1	W	Start DMA Send
1	1	0	R	Input Data
1	1	0	W	Start DMA Target Receive
1	1	1	R	Reset Parity/Interrupts
1	1	1	W	Start DMA Initiator Receive

Register Summary

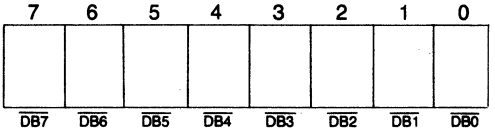
Data Registers

The data registers are used to transfer SCSI commands, data, status, and message bytes between the microprocessor data bus and the SCSI bus. The NCR 5380 does not interpret any information that passes through the data registers. The data registers consist of the transparent Current SCSI Data Register, the Output Data Register, and the Input Data Register.

Current SCSI Data Regsiter - Address 0 (Read-only)

The Current SCSI Data Register is a read-only register which allows the microprocessor to read the active SCSI data bus. This is accomplished by activating $\overline{CS}$ with an address on A2-A0 of 000 and issuing an $\overline{IOR}$ pulse. If parity checking is enabled, the SCSI bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during arbitration.

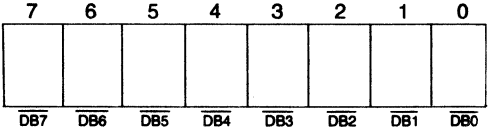
Current SCSI Data Register



Ouptut Data Register - Address 0 (write-only)

The Output Data Register is a write-only register that is used to send data to the SCSI bus. This is accomplished by either using a normal MPU write, or under DMA control, by using $\overline{IOW}$ and $\overline{DACK}$. This register is also used to assert the proper ID bits or the SCSI bus during the arbitration and selection phases.

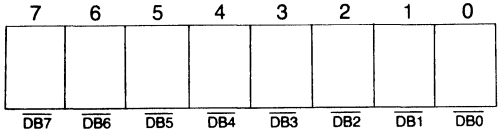
Output Data Register



Input Data Register - Address 6 (Read-only)

The Input Data Register is a read-only register that is used to read latched data from the SCSI bus. Data is latched either during a DMA target receive operation when ACK (pin 14) goes active or during a DMA Initiator receive when REQ (pin 20) goes active. The DMA Mode bit (port 2, bit 1) must be set before data can be latched in the Input Data Register. This register may be read under DMA control using IOR and DACK. Parity is optionally checked when the Input Data Register is loaded.

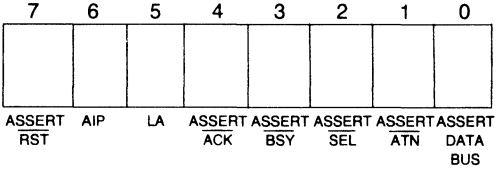
Input Data Register



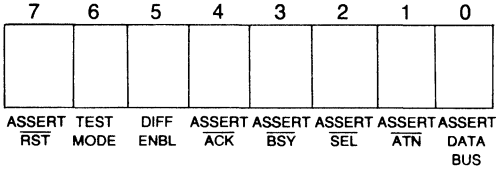
Initiator Command Register - Address 1 (Read/Write)

The Initiator Command Register is a read/write register which is used to assert certain SCSI bus signals, to monitor those signals, and to monitor the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator; however, most can be used during target role operation.

Initiator Command Register (Register Read)



Initiator Command Register (Register Write)



The following describes the operation of all bits in the Initiator Command Register.

BIT 7—ASSERT RST

Whenever a one (1) is written to bit 7 of the Initiator Command Register, the RST signal (pin 16) is asserted on the SCSI bus. The RST signal will remain asserted until this bit is reset or until an external RESET (pin 28) occurs. After this bit is set (1), IRQ (pin 23) goes active and all internal logic and control registers are reset (except for the interrupt latch and the ASSERT RST bit). Writing a zero (0) to bit 7 of the Initiator Command Register de-asserts the RST signal. Reading this register simply reflects the status of this bit.

BIT 6—AIP (Arbitration in Progress—read bit)

This bit is used to determine if arbitration is in progress. For this bit to be active, the ARBITRATE bit (port 2, bit 0) must have been set previously. It indicates that a bus free condition has been detected and that the chip has asserted BSY (pin 13) and the contents of the Output Data Register (port 0) onto the SCSI bus. AIP will remain active until the ARBITRATE bit is reset.

BIT 6—TEST MODE (write bit)

This bit may be written during a test environment to disable all output drivers, effectively removing the NCR 5380 from the circuit. Resetting this bit returns the part to normal operation.

BIT 5—LA (Lost Arbitration—read bit)

This bit, when active, indicates that the NCR 5380 detected a bus free condition, arbitrated for use of the bus by asserting BSY (pin 13) and its ID on the data bus and lost arbitration due to SEL (pin 12) being asserted by another bus device. For this bit to be active the ARBITRATE bit (port 2, bit 0) must be active.

BIT 5—DIFF ENBL (Differential Enable—write bit)

This bit is not used in the NCR 5380 and is only meaningful in the NCR 5381, a 48 pin device which supports external differential pair transceivers. DIFF ENBL should only be asserted if the device is physically connected as either an Initiator or as a Target. If enabled, the signal TGS (pin 14—NCR 5381) is asserted if the TARGET MODE bit (port 2, bit 6) is set (1) or the signal IGS (pin 12—NCR 5381) is asserted if the TARGET MODE bit is reset (0).

BIT 4—ASSERT ACK

This bit is used by the bus initiator to assert ACK (pin 14) on the SCSI bus. In order to assert ACK the TARGET MODE bit (port 2, bit 6) must be false. Writing a zero to this bit resets ACK on the SCSI bus. Reading this register simply reflects the status of this bit.

BIT 3—ASSERT BSY

Writing a one (1) into this bit position asserts BSY (pin 13) onto the SCSI bus. Conversely, a zero (0) resets the BSY signal. Asserting BSY indicates a successful selection or reselection and resetting this bit creates a bus disconnect condition. Reading this register simply reflects the status of this bit.

BIT 2—ASSERT SEL

Writing a one (1) into this bit position asserts SEL (pin 12) onto the SCSI bus. SEL is normally asserted after arbitration has been successfully completed. SEL may be de-asserted by resetting this bit to a zero. A read of this register simply reflects the status of this bit.

BIT 1—ASSERT ATN

ATN (pin 15) may be asserted on the SCSI bus by setting this bit to a one (1) if the TARGET MODE bit (port 2, bit 6) is false. ATN is normally asserted by the initiator to request a Message Out bus phase. Note that since ASSERT SEL and ASSERT ATN are in the same register, a select with ATN may be implemented with one MPU write. ATN may be de-asserted by resetting this bit to a zero (0). A read of th is register simply reflects the status of this bit.

BIT 0—ASSERT DATA BUS

The ASSERT DATA BUS bit, when set, allows the contents of the Output Data Register to be enabled as chip outputs on the signals DB0-DB7. Parity is also generated and asserted on DBP. In the NCR 5381, this bit asserts the DBEN signal (pin 36). Resetting this bit disables the output data bus or the DBEN signal.

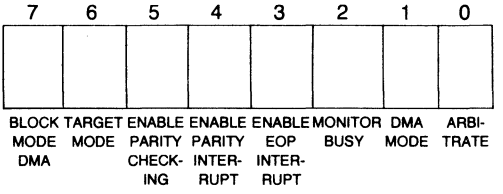
When connected as an Initiator, the outputs are only enabled if the TARGET MODE bit (port 2, bit 6) is false, the received signal I/O (pin 17) is false, and the phase signals (C/D, I/O, and MSG) match the contents of the ASSERT C/D ASSERT I/O, and ASSERT MSG in the Target Command Register.

This bit should also be set during DMA send operations.

Mode Register—Address 2 (Read/Write)

The Mode Register is used to control the operation of the chip. This register determines whether the NCR 5380 operates as an initiator or a target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register may be read to check the value of these internal control bits. The following describes the operation of these control bits.

Mode Register



BIT 7—BLOCK MODE DMA

The BLOCK MODE DMA bit controls the characteristics of the DMA DRQ-DACK handshake. When this bit is reset (0) and the DMA MODE bit is active (1), the DMA handshake uses the normal interlocked handshake and the rising edge of DACK (pin 26) indicates the end of each byte being transferred. In block mode operation, BLOCK MODE DMA bit set (1) and DMA MODE bit set (1), the end of IOR (pin 24) or IOW (pin 29) signifies the end of each byte transferred and DACK is allowed to remain active throughout the DMA operation. READY (pin 25) can then be used to request the next transfer.

BIT 6—TARGET MODE

The TARGET MODE bit allows the NCR 5380 to operate as either an SCSI bus initiator, bit reset (0), or as an SCSI bus target device, bit set (1). In order for the signals ATN (pin 15) and ACK (pin 14) to be asserted on the SCSI bus, the TARGET MODE bit must be reset (0). In order for the signals C/D, I/O, MSG and REQ to be asserted on the SCSI bus, the TARGET MODE bit must be set (1).

BIT 5—ENABLE PARITY CHECKING

The ENABLE PARITY CHECKING bit determines whether parity errors will be ignored or saved in the parity error latch. If this bit is reset (0), parity will be ignored. Conversely, if this bit is set (1) parity errors will be saved.

BIT 4—ENABLE PARITY INTERRUPT

The ENABLE PARITY INTERRUPT bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the ENABLE PARITY CHECKING bit (bit 5) is also enabled (1).

BIT 3—ENABLE EOP INTERRUPT

The ENABLE EOP INTERRUPT, when set (1), causes an interrupt to occur when an $\overline{EOP}$ (End of Process) signal (pin 27) is received from the DMA controller logic.

BIT 2—MONITOR BUSY

The MONITOR BUSY bit, when true (1), causes an interrupt to be generated for an unexpected loss of $\overline{BSY}$ (pin 13). When the interrupt is generated due to loss of $\overline{BSY}$, the lower 6 bits of the Initiator Command Register are reset (0) and all signals are removed from the SCSI bus.

BIT 1—DMA MODE

The DMA MODE bit is normally used to enable a DMA transfer and must be set (1) prior to writing ports 5 through 7. Ports 5 through 7 are used to start DMA transfers. The TARGET MODE bit (port 2, bit 6) must be consistent with writes to port 6 and 7 [i.e. set (1) for a write to port 6 and reset (0) for a write to port 7]. The control bit ASSERT DATA BUS (port 1, bit 0) must be true (1) for all DMA send operations. In the DMA mode, $\overline{REQ}$ (pin 20) and $\overline{ACK}$ (pin 14) are automatically controlled.

The DMA MODE bit is not reset upon the receipt of an $\overline{EOP}$ signal. Any DMA transfer may be stopped by writing a zero into this bit location, however care must be taken not to cause $\overline{CS}$ and $\overline{DACK}$ to be active simultaneously.

Note: $\overline{BSY}$ must be active in order to set the DMA Mode bit.

BIT 0—ARBITRATE

The ARBITRATE bit is set (1) to start the arbitration process. Prior to setting this bit the Output Data Register should contain the proper SCSI device ID value. Only one data bit should be active for SCSI bus arbitration. The NCR 5380 will wait for a bus free condition before entering the arbitration phase. The results of the arbitration phase may be determined by reading the status bits LA and AIP (port 1, bits 5 & 6 respectively).

Target Command Register - Address 3 (Read/Write)

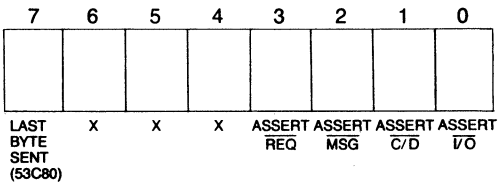
When connected as a target device, the Target Command Register allows the MPU to control the SCSI bus information transfer phase and/or to assert $\overline{REQ}$ (pin 20) simply by writing this register. The TARGET MODE bit (port 2, bit 6) must be true (1) for bus assertion to occur. The SCSI bus phases are described in the following table.

SCSI Information Transfer Phases

Bus Phase	ASSERT $\overline{I/O}$	ASSERT $\overline{C/D}$	ASSERT $\overline{MSG}$
Data Out	0	0	0
Unspecified	0	0	1
Command	0	1	0
Message Out	0	1	1
Data In	1	0	0
Unspecified	1	0	1
Status	1	1	0
Message In	1	1	1

When connected as an Initiator with DMA Mode true, if the phase lines (I/O, C/D, and MSG) do not match the phase bits in the Target Command Register, a phase mismatch interrupt is generated when $\overline{REQ}$ (pin 20) goes active. In order to send data as an Initiator, the ASSERT $\overline{I/O}$, ASSERT $\overline{C/D}$, and ASSERT $\overline{MSG}$ bits must match the corresponding bits in the Current SCSI Bus Status Register (port 4). The ASSERT $\overline{REQ}$ bit (bit 3) has no meaning when operating as an Initiator.

Target Command Register

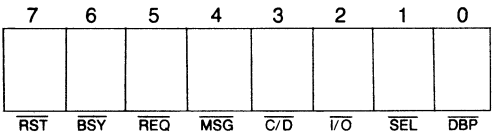


The NCR 53C80 uses bit 7 of this register to determine when the last byte of a DMA transfer is sent to the SCSI bus. This flag is necessary since the End of DMA bit in the Bus and Status Register only reflects when the last byte was received from the DMA.

Current SCSI Bus Status Register - Address 4 (Read-only)

The Current SCSI Bus Status register is a read-only register which is used to monitor seven SCSI bus control signals plus the data bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll REQ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. The following describes the Current SCSI Bus Status Register.

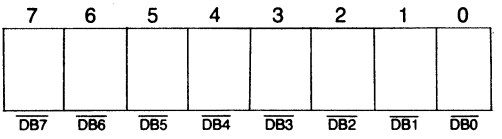
Current SCSI Bus Status Register



Select Enable Register - Address 4 (Write-only)

The Select Enable Register is a write-only register which is used as a mask to monitor a single ID during a selection attempt. The simultaneous occurrence of the correct ID bit, BSY false, and SEL true will cause an interrupt. This interrupt can be disabled by resetting all bits in this register. If the ENABLE PARITY CHECKING bit (port 2, bit 5) is active (1), parity will be checked during selection.

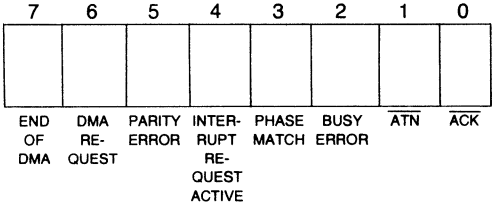
Select Enable Register



Bus and Status Register - Address 5 (Read-only)

The Bus and Status Register is a read-only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status Register (ATN & ACK) as well as six other status bits. The following describes each bit of the Bus and Status Register individually.

Bus and Status Register



BIT 7—END OF DMA TRANSFER

The END OF DMA TRANSFER bit is set if EOP (pin 27), DACK (pin 26), and either IOR (pin 24), or IOW (pin 29) are simultaneously active for at least 100 nsec. Since the EOP signal can occur during the last byte sent to the Output Data Register (port 0), the REQ and ACK signals should be monitored to insure that the last byte has been transferred. This bit is reset when the DMA MODE bit is reset (0) in the Mode Register (port 2).

The NCR 53C80 contains a true End of DMA Status bit (last byte sent) in bit 7 of the Target Command Register.

BIT 6—DMA REQUEST

The DMA REQUEST bit allows the MPU to sample the output pin DRQ (pin 22). DRQ can be cleared by asserting $\overline{\text{DACK}}$ (pin 26) or by resetting the DMA MODE bit (bit 1) in the Mode Register (port 2). The DRQ signal does not reset when a phase mismatch interrupt occurs.

BIT 5—PARITY ERROR

This bit is set if a parity error occurs during a data receive or a device selection. The PARITY ERROR bit can only be set (1) if the ENABLE PARITY CHECK bit (port 2, bit 5) is active (1). This bit may be cleared by reading the Reset Parity/Interrupt Register (port 7).

BIT 4—INTERRUPT REQUEST ACTIVE

This bit is set if an enabled interrupt condition occurs. It reflects the current state of the IRQ (pin 23) output and can be cleared by reading the Reset Parity/ Interrupt Register (port 7).

BIT 3—PHASE MATCH

The SCSI signals $\overline{\text{MSG}}$, $\overline{\text{C}}/\overline{\text{D}}$ and $\overline{\text{I}}/\overline{\text{O}}$ (pins 19, 18 and 17) represent the current information transfer phase. The PHASE MATCH bit indicates whether the current SCSI bus phase matches the lower 3 bits of the Target Command Register. PHASE MATCH is continuously updated and is only significant when operating as a bus initiator. A Phase Match is required for data transfers to occur on the SCSI bus.

BIT 2—BUSY ERROR

The BUSY ERROR bit is active if an unexpected loss of $\overline{\text{BSY}}$ signal (pin 13) has occurred. This level-sensitive latch is set whenever the MONITOR BUSY bit (port 2, bit 2) is true and $\overline{\text{BSY}}$ is false. An unexpected loss of $\overline{\text{BSY}}$ will disable any SCSI outputs and will reset the DMA MODE bit (port 2, bit 1).

BIT 1—ATN

This bit reflects the condition of the SCSI bus control signal $\overline{\text{ATN}}$ (pin 15). The signal is normally monitored by the target device.

BIT 0—ACK

This bit reflects the condition of the SCSI bus control signal $\overline{\text{ACK}}$ (pin 14). This signal is normally monitored by the target device.

DMA Registers

Three write-only registers are used to initiate all DMA activity. They are Start DMA Send (port 5), Start DMA Target Receive (port 6) and Start DMA Initiator Receive (port 7). Simply writing these registers starts the DMA transfers. Data presented to the NCR 5380 on signals D0-D7 during the register write is meaningless and has no effect on the operation. Prior to writing these registers the BLOCK MODE DMA bit (bit 7), the DMA MODE bit (bit 1) and the TARGET MODE bit (bit 6) in the Mode Register (port 2) must be appropriately set. The individual registers are briefly described below.

Start DMA Send - Address 5 (Write-only)

This register is written to initiate a DMA send, from the DMA to the SCSI bus, for either initiator or target role operations. The DMA MODE bit (port 2, bit 1) must be set prior to writing this register.

Start DMA Target Receive - Address 6 (Write-only)

This register is written to initiate a DMA receive, from the SCSI bus to the DMA, for target operation only. The DMA MODE bit (bit 1) and the TARGET MODE bit (bit 6) in the Mode Register (port 2) must both be set (1) prior to writing this register.

Start DMA Initiator Receive - Address 7 (Write-only)

This register is written to initiate a DMA receive, from the SCSI bus to the DMA, for initiator operation only. The DMA MODE bit (bit 1) must be true (1) and the TARGET MODE bit (bit 6) must be false (0) in the Mode Register (port 2) prior to writing this register.

Reset Parity/Interrupt - Address 7 (Read-only)

Reading this register resets the PARITY ERROR bit (bit 5), the ITNERRUPT REQUEST bit (bit 4) and the BUSY ERROR bit (bit 2) in the Bus and Status Register (port 5).

On-Chip SCSI Hardware Support

The NCR 5380 is easy to use because of its simple architecture. The chip allows direct control and monitoring of the SCSI bus by providing a latch for each signal. However, portions of the protocol define timings which are much too quick for traditional micorprocessors to control. Therefore, hardware support has been provided for DMA transfers, bus arbitration, phase change monitoring, bus disconnection, bus reset, parity generation, parity checking, and device selection/reselection.

Arbitration is accomplished using a bus-free filter to continuously monitor $\overline{BSY}$. If $\overline{BSY}$ remains inactive for at least 400 nsec then the SCSI bus is considered free and arbitration may begin. Arbitration will begin if the bus is free, $\overline{SEL}$ is inactive and the ARBITRATION bit (port 2, bit 0) is active. Once arbitration has begun ($\overline{BSY}$ asserted), an arbitration delay of 2.2 μsec must elapse before the data bus can be examined to determine if arbitration has been won. This delay must be implemented in the controlling software driver.

The NCR 5380 is a clockless device. Delays such as bus free delay, bus set delay and bus settle delay are implemented using gate delays. These delays may differ between devices because of inherent process variations, but are well within the proposed ANSI X3T9.2 specification (Revision 17).

Interrupts

The NCR 5380 provides an interrupt output (IRQ) to indicate a task completion or an abnormal bus occurrence. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode Register (port 2) or the Select Enable Register (port 4).

When an interrupt occurs, the Bus and Status Register and the Current SCSI Bus Status Register must be read to determine which condition created the interrupt. IRQ (pin 23) can be reset simply by reading the Reset Parity/Interrupt Register (port 7) or by an external chip reset ($\overline{RESET}$ active for 200 nsec).

Assuming the NCR 5380 has been properly initialized, an interrupt will be generated if the chip is selected or reselected, if an $\overline{EOP}$ signal occurs during a DMA transfer, if an SCSI bus reset occurs, if a parity error occurs during a data transfer, if a bus phase mismatch occurs, or if an SCSI bit disconnection occurs.

Selection/Reselection

The NCR 5380 can generate a select interrupt if $\overline{SEL}$ (pin 12) is true (1), its device ID is true (1) and $\overline{BSY}$ (pin 13) is false for at least a bus settle delay (400 ns). If $\overline{I/O}$ (pin 17) is active this should be considered a reselect interrupt. The correct ID bit is determined by a match in the Select Enable Register (port 4). Only a single bit match is requried to generate an interrupt. This interrupt may be disabled by writing zeros into all bits of the Select Enable Register.

If parity is supported, parity should also be good during the selection phase. Therefore, if the ENABLE PARITY BIT (port 2, bit 5) is active, then the PARITY ERROR bit should be checked to insure that a proper selection has occurred. The ENABLE PARITY INTERRUPT bit need not be set for this interrupt to be generated.

The proposed SCSI specification also requires that no more than two device IDs be active during the selection process. To insure this, the Current SCSI Data Register (port 0) should be read.

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed below.

Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	1	X	0	X	0
END OF DMA	DMA REQUEST	PARTIY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
0	0	0	X	X	X	1	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

End of Process (EOP) Interrupt

An End of Process signal (EOP, pin 27) which occurs during a DMA transfer (DMA MODE true) will set the END OF DMA status bit (port 5, bit 7) and will optionally generate an interrupt if ENABLE EOP INTERRUPT bit (port 2, bit 3) is true. The EOP pulse will not be recognized (EDN OF DMA bit set) unless EOP, DACK and either IOR or IOW are concurrently active for at least 100 nsec. DMA transfers can still occur if EOP/ was not asserted at the correct time. This interrupt can be disabled by resetting the ENABLE EOP INTERRUPT bit.

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) for this interrupt are displayed below.

Bus and Status Register

7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	X
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	ATN	ACK

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
0	1	X	X	X	X	0	X
RST	BSY	REQ	MSG	C/D	I/O	SEL	DBP

The END OF DMA bit is used to determine when a block transfer is complete. Receive operations are complete when there is no data left in the chip and no additional handshakes occurring. The only exception to this is receiving data as an initiator and the target opts to send additional data for the same phase. In this case, REQ goes active and the new data is present in the Input Data Register. Since a phase mismatch interrupt will not occur, REQ and ACK need to be sampled to determine that the Target is attempting to send more data.

For send operations, the END OF DMA bit is set when the DMA finishes its transfer, but the SCSI transfer may still be in progress. If connected as a Target, REQ and ACK should be sampled until both are false. If connected as an Initiator, a phase change interrupt can be used to signal the completion of the previous phase. It is possible for the Target to request additional data for the same phase. In this case, a phase change will not occur and both REQ and ACK must be sampled to determine when the last byte was transferred.

If using the NCR 53C80, Last Byte Sent (bit 7 of the Target Command Regsiter) may be sampled to determine when the last byte has been transferred.

SCSI Bus Reset

The NCR 5380 generates an interrupt when the RST signal (pin 16) transitions to true. The device releases all bus signals within a bus clear delay (800 nsec) of this transition. This interrupt also occurs after setting the ASSERT RST bit (port 1, bit 7). This interrupt cannot be disabled. (Note: The RST signal is not latched in bit 7 of the Current SCSI Bus Status Register and may not be active when this port is read. For this case, the Bus Reset interrupt may be determined by default.)

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed below.

Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	1	X	0	0	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	ATN	ACK

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
X	0	0	0	0	0	0	0
RST	BSY	REQ	MSG	C/D	I/O	SEL	DBP

Parity Error

An interrupt is generated for a received parity error if the ENABLE PARITY CHECK (bit 5) and the ENABLE PARITY INTERRUPT (bit 4) bits are set (1) in the Mode Register (port 2). Parity is checked during a read of the Current SCSI Data Register (port 0) and during a DMA receive operation. A parity error can be detected without generating an interrupt by disabling the ENABLE PARITY INTERRUPT bit and checking the PARITY ERROR flag (port 5, bit 5).

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed below.

Bus and Status Register

7	6	5	4	3	2	1	0
0	X	1	1	1	0	X	X
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
0	1	X	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Bus Phase Mismatch

The SCSI phase lines are comprised of the signals $\overline{I/O}$, $\overline{C/D}$ and $\overline{MSG}$. These signals are compared with the corresponding bits in the Target Command Register: ASSERT $\overline{I/O}$ (bit 0), ASSERT $\overline{C/D}$ (bit 1) and ASSERT $\overline{MSG}$ (bit 2). The comparison occurs continually and is reflected in the PHASE MATCH bit (bit 3) of the Bus and Status Register (port 5). If the DMA MODE bit (port 2, bit 1) is active and a phase mismatch occurs when $\overline{REQ}$ (pin 20) transitions from false to true, an interrupt (IRQ) is generated.

A phase mismatch prevents the recognition of $\overline{REQ}$ and removes the chip from the bus during an initiator send operation. ($\overline{DB0}$-$\overline{DB7}$, $\overline{DBP}$ will not be driven even though the ASSERT DATA BUS bit (port 1, bit 0) is active.) This interrupt is only significant when connected as an Initiator and may be disabled by resetting the DMA MODE bit. (Note: It is possible for this interrupt to occur when connected as a Target if another device is driving the phase lines to a different state.)

The proper values for the Bus and Status Register (port 5) and the Current SCSI Bus Status Register (port 4) are displayed below.

Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	1	0	0	X	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
0	1	1	X	X	X	0	X
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Loss of $\overline{BSY}$

If the MONITOR BUSY bit (bit 2) in the Mode Register (port 2) is active, an interrupt will be generated if the $\overline{BSY}$ signal (pin 13) goes false for at least a bus settle delay (400 nsec). This interrupt may be disabled by resetting the MONITOR BUSY bit. Register values are as follows.

Bus and Status Register

7	6	5	4	3	2	1	0
0	0	0	1	X	1	0	0
END OF DMA	DMA REQUEST	PARITY ERROR	INTERRUPT REQUEST ACTIVE	PHASE MATCH	BUSY ERROR	$\overline{ATN}$	$\overline{ACK}$

Current SCSI Bus Status Register

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
$\overline{RST}$	$\overline{BSY}$	$\overline{REQ}$	$\overline{MSG}$	$\overline{C/D}$	$\overline{I/O}$	$\overline{SEL}$	$\overline{DBP}$

Reset Conditions

Three possible reset situations exist with the NCR 5380, as follows:

Hardware Chip Reset

When the signal RESET/ (pin 28) is active for at least 200 nsec, the NCR 5380 device is re-initialized and all internal logic and control registers are cleared. This is a chip reset only and does not create an SCSI bus reset condition.

SCSI Bus Reset ($\overline{RST}$) Received

When an SCSI $\overline{RST}$ signal (pin 16) is received, an IRQ interrupt is generated and a chip reset is performed. All internal logic and registers are cleared, except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1). (Note: The $\overline{RST}$ signal may be sampled by reading the Current SCSI Bus Status Register (port 4); however, this signal is not latched and may not be present when this port is read.)

SCSI Bus Reset ($\overline{RST}$) Issued

If the CPU sets the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1), the $\overline{RST}$ signal (pin 16) goes active on the SCSI bus and an internal reset is performed. Again, all internal logic and registers are cleared except for the IRQ interrupt latch and the ASSERT $\overline{RST}$ bit (bit 7) in the Initiator Command Register (port 1). The $\overline{RST}$ signal will continue to be active until the ASSERT $\overline{RST}$ bit is reset or until a hardware reset occurs.

Data Transfers

Data may be transferred between SCSI bus devices in one of four modes: Programmed I/O: Normal DMA; Block Mode DMA; or Pseudo DMA. The following sections describe these modes in detail. (Note: For all data transfers operations $\overline{DACK}$ and $\overline{CS}$ should never be active simultaneously.)

Programmed I/O Transfers

Programmed I/O is the most primitive form of data transfer. The $\overline{REQ}$ (pin 20) and $\overline{ACK}$ (pin 14) handshake signals are individually monitored and asserted by reading and writing the appropriate register bits. This type of transfer is normally used when transferring small blocks of data such as command blocks or message and status bytes.

An Initiator send operation would begin by setting the $\overline{C/D}$, $\overline{I/O}$, and $\overline{MSG}$ bits in the Target Command Register to the correct state so that a phase match exists. In addition to the phase match condition, it is necessary for the ASSERT DATA BUS bit (port 1, bit 0) to be true and the received $\overline{I/O}$ signal to be false for the 5380 to send data.

For each transfer, the data is loaded into the Output Data Register (port 0). The MPU then waits for the REQ bit (port 4, bit 5) to become active. Once $\overline{REQ}$ goes active the PHASE MATCH bit (port 5, bit 3) is checked and the ASSERT $\overline{ACK}$ bit (port 1, bit 4) is set. The $\overline{REQ}$ bit is sampled until it becomes false and the MPU resets the ASSERT $\overline{ACK}$ bit to complete the transfer.

Normal DMA Mode

DMA transfers are normally used for large block transfers. The SCSI chip outputs a DMA request (DRQ - pin 22) whenever it is ready for a byte transfer. External DMA logic uses this DRQ signal to generate DACK and an IOR or an IOW pulse to the NCR 5380. DRQ goes inactive when DACK is asserted and DACK goes inactive sometime after the minimum read or write pulse width. This process is repeated for every byte. For this mode, DACK should not be allowed to cycle unless a transfer is taking place.

Refer to next column for information on halting a DMA transfer.

Block Mode DMA

Block Mode allows an external DMA device (Intel 8237-type DMA) to perform sequential DMA transfer without relinquishing the data bus to the CPU. Holding DACK active prevents Intel-type CPUs from gaining access to the sytem bus. The handshake itself does not increase the transfer rate. Preventing the CPU from sharing the system bus increases the DMA transfer rate but also halts the CPU operation.

Block Mode DMA transfers are supported for both Initiator and target role operation. When using this mode of operation, DRQ is asserted to signal the beginning of the DMA transfer. In response to DRQ, DACK is asserted and remains asserted throughout the transfer. READY goes active after the IOW or IOR pulse goes inactive, effectively replacing the DRQ signal.

Care must be taken when using this mode due to the operation of READY. If, for example, a phase mismatch interrupt occurs, READY will remain in the inactive state and INT will be active. For this condition the DMA chip must return control of the bus to the CPU so that the 5380 interrupt can be serviced. READY also does not return to the active state when an EOP pulse is received. Therefore, you might want to use EOP to insure that the CPU regains bus control after the last DMA byte has been transferred. As in the non-block DMA mode, the EOP signal does not de-assert ACK on the SCSI bus. To successfully complete a DMA send operation, either an additional byte of data must be written to the NCR 5380 to allow ACK to go inactive, or the CPU must reset the DMA Mode bit in the Mode Register.

Non-block mode DMA transfers end when DACK goes false, whereas block mode transfers end when IOR or IOW becomes inactive after each byte. Since this is the case, DMA transfers may be started sooner in a block mode transfer.

To obtain optimum performance in block mode operation, the DMA logic may optionally use the normal DMA mode interlocking handshake. READY is still available to throttle the DMA transfer, but DRQ is 30 to 40 nsec faster than READY and may be used to start the cycle sooner.

The methods described in "Halting A DMA Operation" apply for all DMA operations.

Pseudo DMA Mode

To avoid the tedium of monitoring and asserting the request/acknowledge handshake signals for programmed I/O transfers, the system may be designed to implement a pseudo DMA mode. This mode is implemented by programming the NCR 5380 to operate in the DMA MODE, but using the MPU to emulate the DMA handshake. DRQ (pin 22) may be detected by polling the DMA REQ bit (bit 6) in the Bus and Status Register (port 5), by sampling the signal through an external port or by using it to generate an MPU interrupt. Once DRQ is detected, the MPU can perform a DMA port read or write data transfer. This MPU read/write is externally decoded to generate the appropriate DACK and IOR or IOW signals.

Often, external decoding logic is necessary to generate the NCR 5380 CS signal. This same logic may be used to generate DACK at no extra system cost and provide an increased performance in programmed IO transfers.

Halting a DMA Operation

The EOP signal is not the only way to halt a DMA transfer. A bus phase mismatch or a reset of the DMA MODE bit (port 2, bit 1) can also terminate a DMA cycle for the current bus phase.

Using the $\overline{EOP}$ Signal

If $\overline{EOP}$ is used, it should be asserted for at least 100 nsec while $\overline{DACK}$ and $\overline{IOR}$ or $\overline{IOW}$ are simultaneously active. Note, however, that if $\overline{IOR}$ or $\overline{IOW}$ is not active an interrupt will be generated, but the DMA activity will continue. The $\overline{EOP}$ signal does not reset the DMA MODE bit. Since the $\overline{EOP}$ signal can occur during the last byte sent to the Output Data Register (port 0), the $\overline{REQ}$ and $\overline{ACK}$ signals should be monitored to insure that the last byte has transferred.

Bus Phase Mismatch Interrupt

A bus phase mismatch interrupt may be used to halt the transfer if operating as an Initiator. Using this method frees the host from maintaining a data length counter and frees the DMA logic from providing the $\overline{EOP}$ signal. If performing an initiator send operation, the NCR 5380 requires $\overline{DACK}$ to cycle before $\overline{ACK}$ goes inactive. Since phase changes cannot occur if $\overline{ACK}$ is active, either $\overline{DACK}$ must be cycled after the last byte is sent or the DMA MODE bit must be reset in order to receive the phase mismatch interrupt.

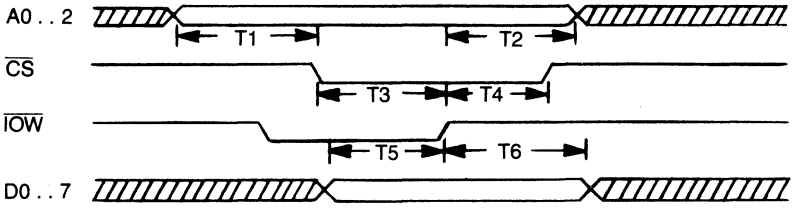
Resetting the DMA MODE Bit

A DMA operation may be halted at any time simply by resetting the DMA MODE bit. It is recommended that the DMA MODE bit be reset after receiving an $\overline{EOP}$ or bus phase mismatch interrupt. The DMA MODE bit must then be set before writing any of the start DMA registers for subsequent bus phases.

If resetting the DMA MODE bit is used instead of $\overline{EOP}$ for target role operation, then care must be taken to reset this bit at the proper time. If receiving data as a target device, the DMA MODE bit must be reset once the last DRQ is received and before $\overline{DACK}$ is asserted to prevent an additional $\overline{REQ}$ from occurring. Resetting this bit causes DRQ to go inactive. However, the last byte received remains in the Input Data Register and may be obtained either by performing a normal MPU read or by cycling $\overline{DACK}$ and $\overline{IOR}$. In most cases $\overline{EOP}$ is easier to use when operating as a Target device.

External Timing Diagrams

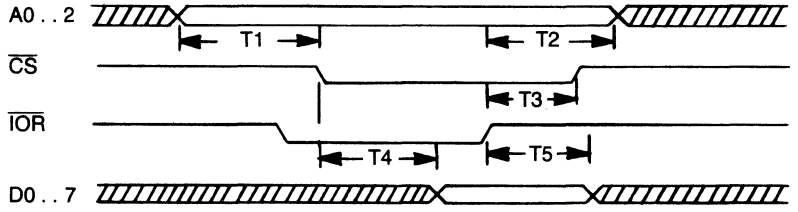
CPU Write



NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	Address setup to write enable *	20			ns.
T2	Address hold from end write enable *	20			ns.
T3	Write enable width *	70			ns.
T4	Chip select hold from end of $\overline{IOW}$	0			ns.
T5	Data setup to end of write enable *	50			ns.
T6	Data hold time from end of $\overline{IOW}$	30			ns.

* Write enable is the occurrence of $\overline{IOW}$ and $\overline{CS}$

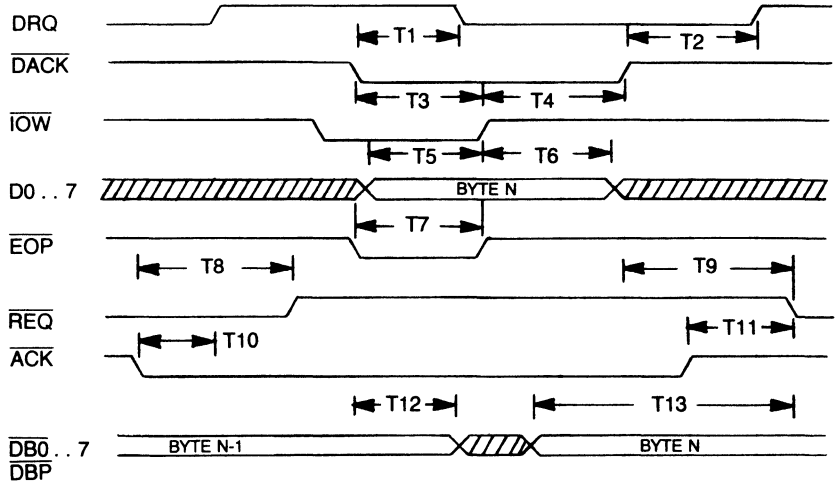
CPU Read



NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	Address setup to read enable *	20			ns.
T2	Address hold from end read enable *	20			ns.
T3	Chip select hold from end of $\overline{IOR}$	0			ns.
T4	Data access time from read enable *			130	ns.
T5	Data hold time from end of $\overline{IOR}$	20			ns.

* Read enable is the occurrence of $\overline{IOR}$ and $\overline{CS}$

DMA Write (Non-Block Mode) Target Send

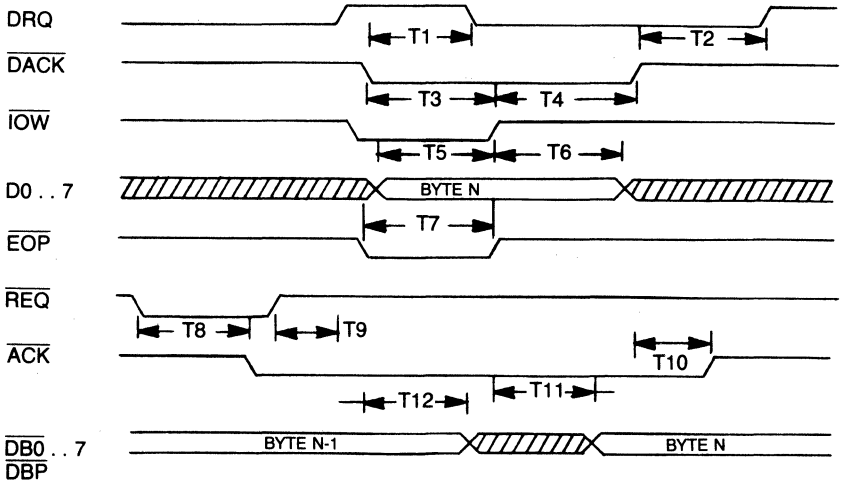


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from $\overline{\text{DACK}}$ true			130	ns.
T2	$\overline{\text{DACK}}$ false to DRQ true	30			ns.
T3	Write enable width *	100			ns.
T4	DACK hold from end of $\overline{\text{IOW}}$	0			ns.
T5	Data setup to end of write enable *	50			ns.
T6	Data hold time from end of IOW	40			ns.
T7	Width of $\overline{\text{EOP}}$ pulse (note 1)	100			ns.
T8	$\overline{\text{ACK}}$ true to $\overline{\text{REQ}}$ false	25	110	125	ns.
T9	$\overline{\text{REQ}}$ from end of $\overline{\text{DACK}}$ ($\overline{\text{ACK}}$ false)	30	140	150	ns.
T10	$\overline{\text{ACK}}$ true to DRQ true (target)	15	100	110	ns.
T11	$\overline{\text{REQ}}$ from end of $\overline{\text{ACK}}$ ($\overline{\text{DACK}}$ false)	20	140	150	ns.
T12	DATA hold from write enable	15			ns.
T13	Data setup to $\overline{\text{REQ}}$ true (target)	60			ns.

* Write enable is the occurrence of $\overline{\text{IOW}}$ and $\overline{\text{DACK}}$

Note 1: $\overline{\text{EOP}}$, $\overline{\text{IOW}}$, and $\overline{\text{DACK}}$ must be concurrently true for at least T7 for proper recognition of the $\overline{\text{EOP}}$ pulse.

DMA Write (Non-Block Mode) Initiator Send

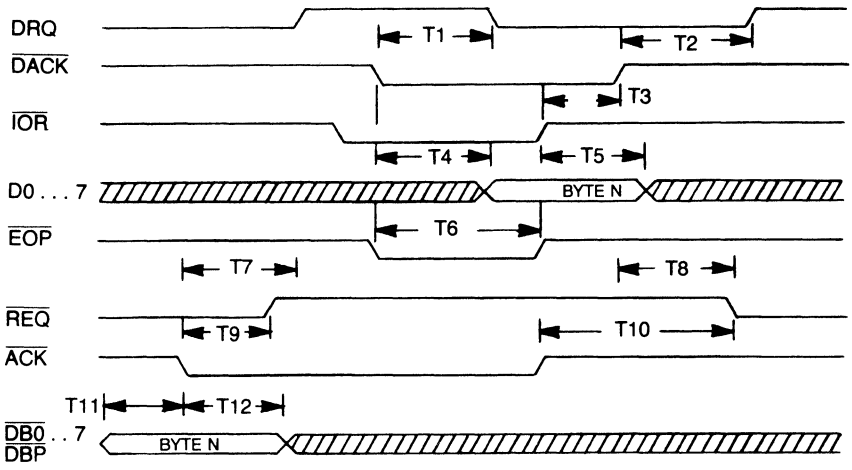


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from DACK true			130	ns.
T2	DACK false to DRQ true	30			ns.
T3	Write enable width *	100			ns.
T4	DACK hold from end of IOW	0			ns.
T5	Data setup to end of write enable *	50			ns.
T6	Data hold time from end of IOW	40			ns.
T7	Width of EOP pulse (note 1)	100			ns.
T8	REQ true to ACK true	20	150	160	ns.
T9	REQ false to DRQ true	20	100	110	ns.
T10	DACK false to ACK false	25	140	150	ns.
T11	IOW false to valid SCSI data			100	ns.
T12	DATA hold from write enable	15			ns.

* Write enable is the occurrence of IOW and DACK

Note 1: EOP, IOW, and DACK must be concurrently true for at least T7 for proper recognition of the EOP pulse.

DMA Read (Non-Block Mode) Target Receive

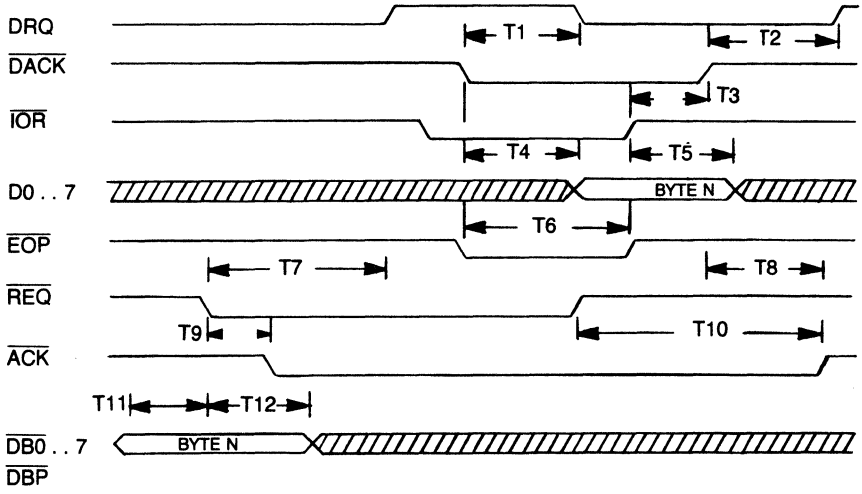


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from $\overline{DACK}$ true			130	ns.
T2	$\overline{DACK}$ false to DRQ true	30			ns.
T3	$\overline{DACK}$ hold time from end of $\overline{IOR}$	0			ns.
T4	Data access time from read enable *			115	ns.
T5	Data hold time from end of $\overline{IOR}$	20			ns.
T6	Width of $\overline{EOP}$ pulse (note 1)	100			ns.
T7	$\overline{ACK}$ true to DRQ true	15	100	110	ns.
T8	$\overline{DACK}$ false to $\overline{REQ}$ true ($\overline{ACK}$ false)	30		150	ns.
T9	$\overline{ACK}$ true to $\overline{REQ}$ false	25	110	125	ns.
T10	$\overline{ACK}$ false to $\overline{REQ}$ true ($\overline{DACK}$ false)	20	140	150	ns.
T11	DATA setup time to $\overline{ACK}$	20			ns.
T12	DATA hold time from $\overline{ACK}$	50			ns.

* Read enable is the occurrence of $\overline{IOR}$ and $\overline{DACK}$

Note 1: $\overline{EOP}$, $\overline{IOR}$, and $\overline{DACK}$ must be concurrently true for at least T6 for proper recognition of the $\overline{EOP}$ pulse.

DMA Read (Non-Block Mode) Initiator Receive

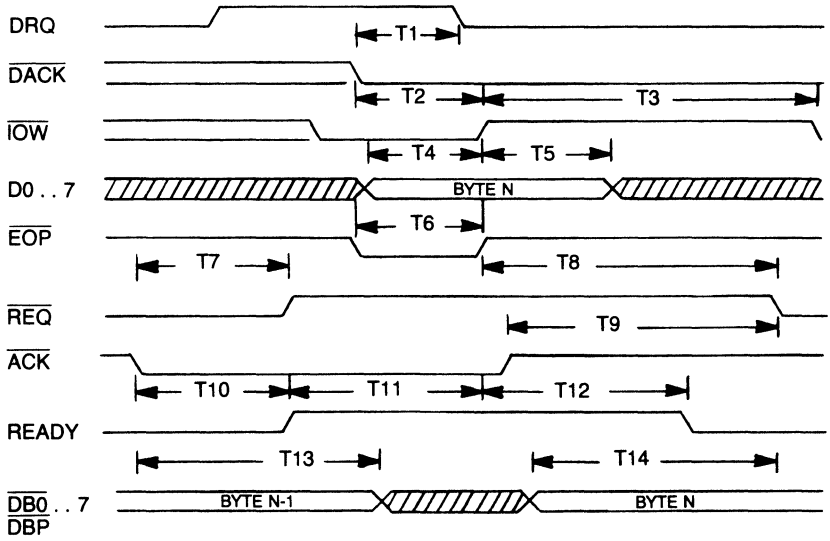


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from DACK true			130	ns.
T2	DACK false to DRQ true	30			ns.
T3	DACK hold time from end of IOR	0			ns.
T4	Data access time from read enable *			115	ns.
T5	Data hold time from end of IOR	20			ns.
T6	Width of EOP pulse (note 1)	100			ns.
T7	REQ true to DRQ true	20	140	150	ns.
T8	DACK false to ACK false (REQ false)	25	140	160	ns.
T9	REQ true to ACK true	20	150	160	ns.
T10	REQ false to ACK false (DACK false)	15	120	140	ns.
T11	DATA setup time to REQ	20			ns.
T12	DATA hold time from REQ	50			ns.

*Read enable is the occurrence of IOR and DACK

Note 1: EOP, IOR, and DACK must be concurrently true for at least T6 for proper recognition of the EOP pulse.

DMA Write (Block Mode) Target Send

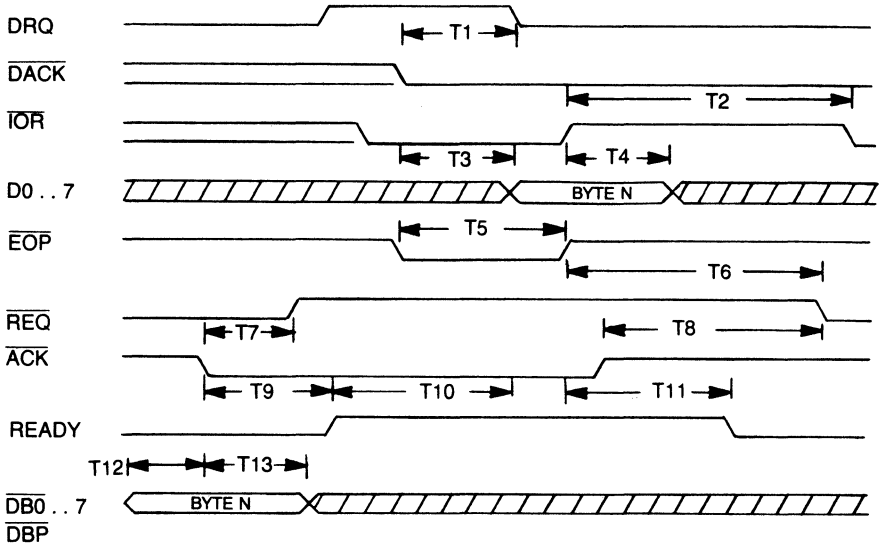


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from DACK true			130	ns.
T2	Write enable width *	100			ns.
T3	Write recovery time	120			ns.
T4	Data setup to end of write enable *	50			ns.
T5	Data hold time from end of IOW	40			ns.
T6	Width of EOP pulse (note 1)	100			ns.
T7	ACK true to REQ false	25	110	125	ns.
T8	REQ from end of IOW (ACK false)	40		180	ns.
T9	REQ from end of ACK (IOW false)	20	160	170	ns.
T10	ACK true to READY true	20	130	140	ns.
T11	READY true to IOW false	70			ns.
T12	IOW false to READY false	20	130	140	ns.
T13	DATA hold from ACK true	40			ns.
T14	Data setup to REQ true	60			ns.

* Write enable is the occurrence of IOW and DACK

Note 1: EOP, IOW, and DACK must be concurrently true for at least T6 for proper recognition of the EOP pulse.

DMA Read (Block Mode) Target Receive

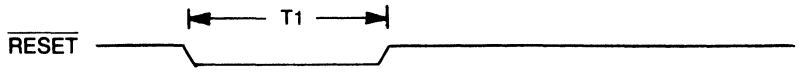


NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	DRQ false from DACK true			130	ns.
T2	IOR recovery time	120			ns.
T3	Data access time from read enable *		100	110	ns.
T4	Data hold time from end of IOR	20			ns.
T5	Width of EOP pulse (note 1)	100			ns.
T6	IOR false to REQ true (ACK false)	30	180	190	ns.
T7	ACK true to REQ false	25	110	125	ns.
T8	ACK false to REQ true (IOR false)	20	160	170	ns.
T9	ACK true to READY true	20	130	140	ns.
T10	READY true to valid data			50	ns.
T11	IOR false to READY false	20	125	140	ns.
T12	DATA setup time to ACK	20			ns.
T13	DATA hold time from ACK	50			ns.

* Read enable is the occurrence of IOR and DACK

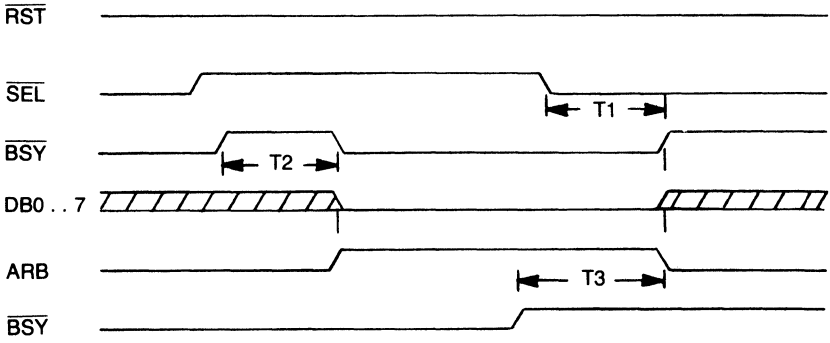
Note 1: EOP, IOR, and DACK must be concurrently true for at least T5 for proper recognition of the EOP pulse.

Reset



NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	Minimum width of reset	200			ns.

Arbitration



NAME	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
T1	Bus clear from SEL true			600	ns.
T2	ARBITRATE start from BSY false	1200		2200	ns.
T3	Bus clear from BSY false			1100	ns.

A1. NCR 5380 vs. NCR 5385/86

The NCR 5380 was designed to provide a low-cost SCSI interface using a minimum number of parts. Much of the intelligence and some of the features included in the NCR 5385E/86 have been removed. In some instances, such as arbitration, this causes the controlling CPU to provide more of the protocol control. The NCR 5385/86 remains appropriate for many applications and will continue to be strongly supported.

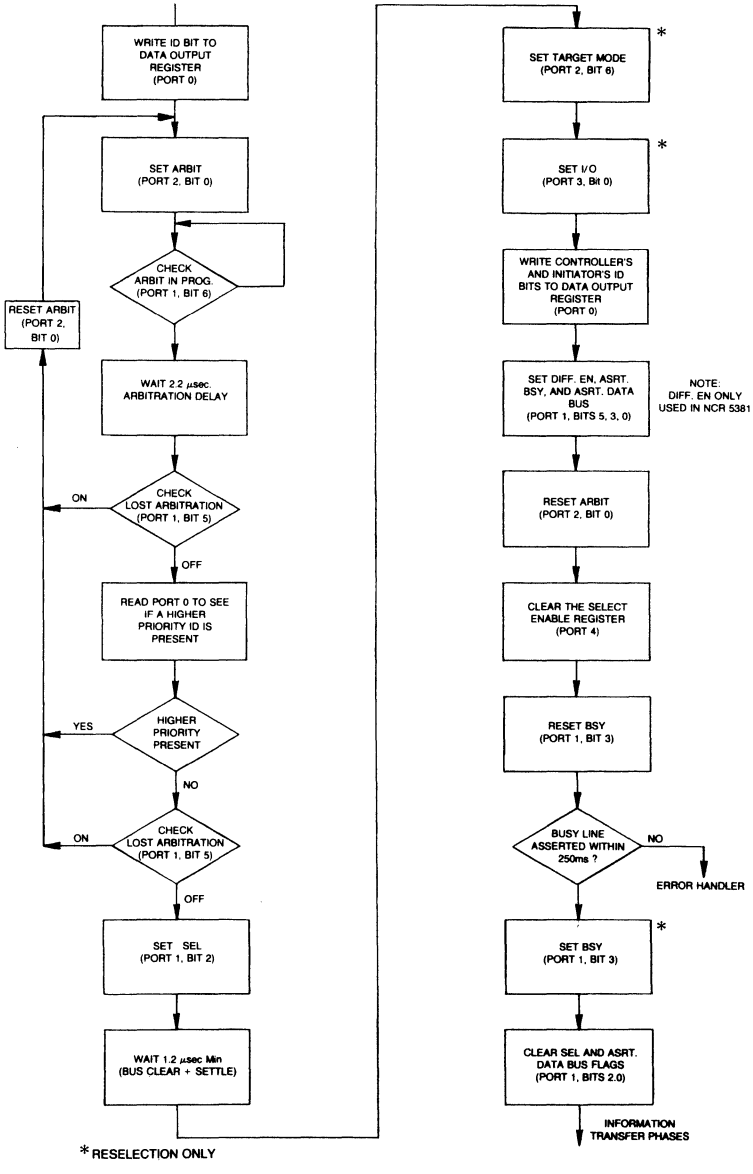
The main differences between the NCR 5380 and the NCR 5385E/86 are shown in the following table.

Functional Areas	5380	5385E/86
Arbitration	Optional, Firmware Dependant	Automatically Invoked
Maximum Transfer Rate	1.5 MBPS	2.0 MBPS
Transfer Counter	None	24 bits
Data Buffering	Single	Double
Clock Circuitry	None Req'd	5-10 MHz
Single-Ended Transceivers	On-chip	External
Differential Pair	External (NCR 5381)	External
Synchronous Mode	No Firm Plans	

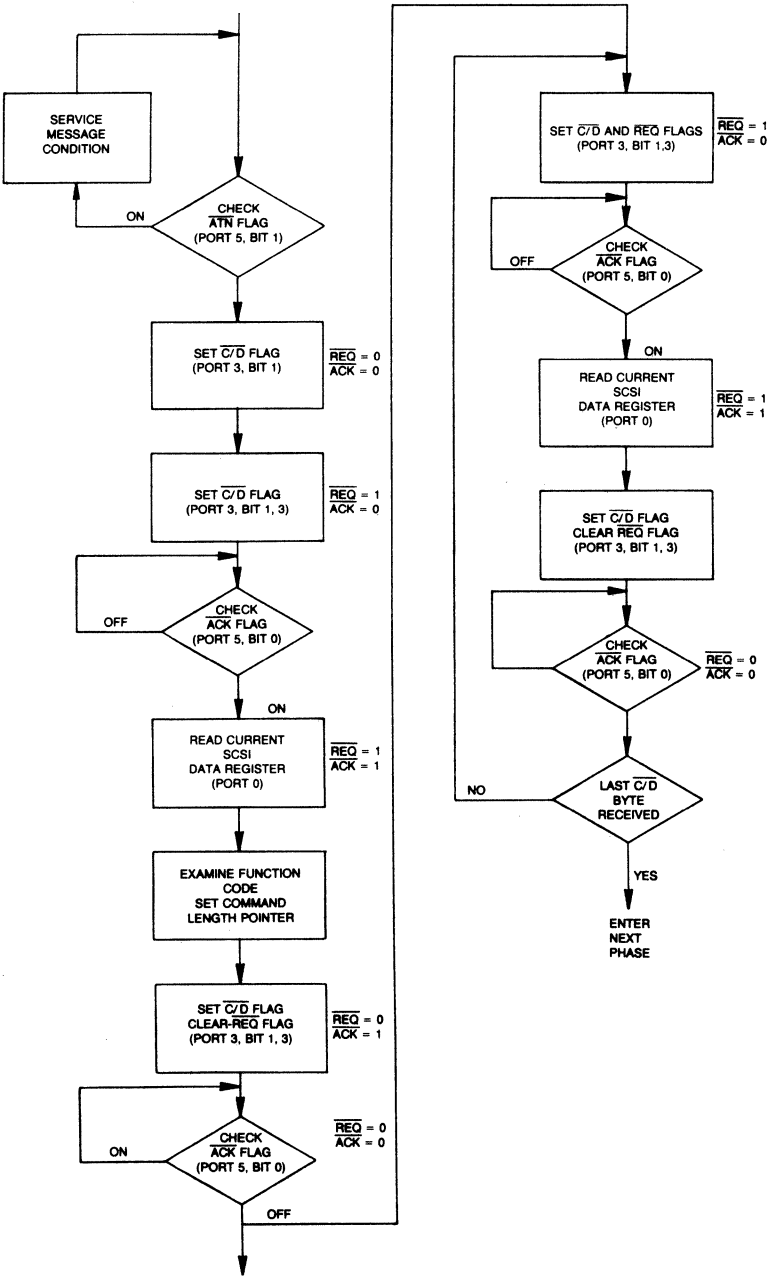
A2. Flowcharts/Software

Flowcharts and sample software drivers are provided as a guideline to facilitate your firmware development. Firmware will vary depending on the application and the level of the SCSI protocol being supported. In accordance with register definitions, a one (1) designates signal assertion and a zero (0) designates signal non-assertion, or the inactive state.

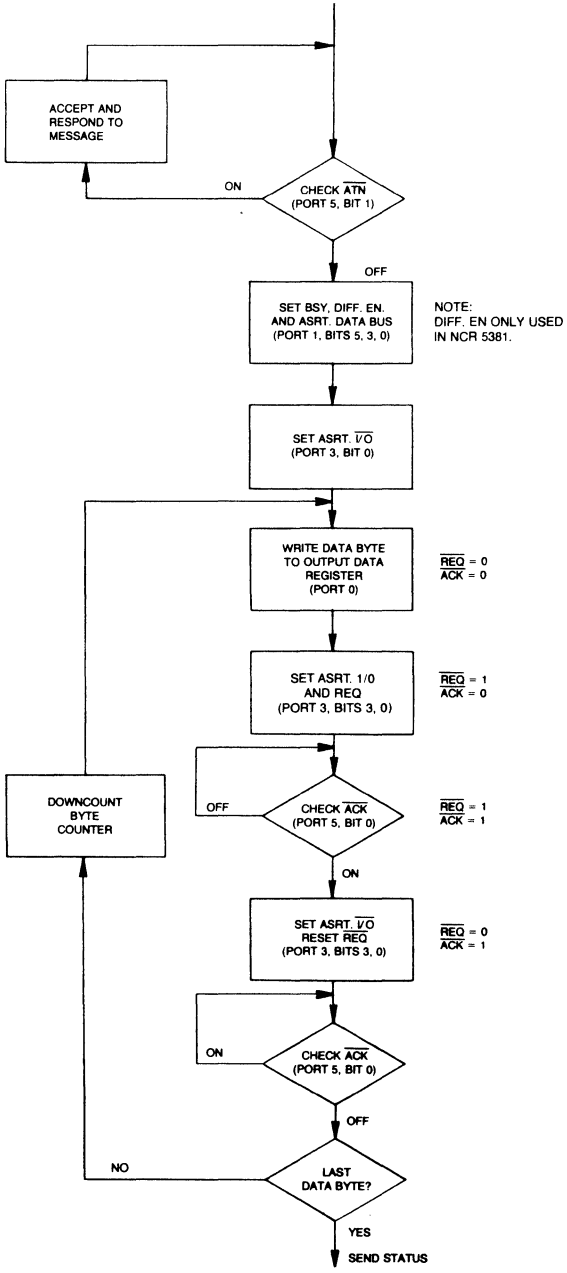
ARBITRATION AND (RE) SELECTION



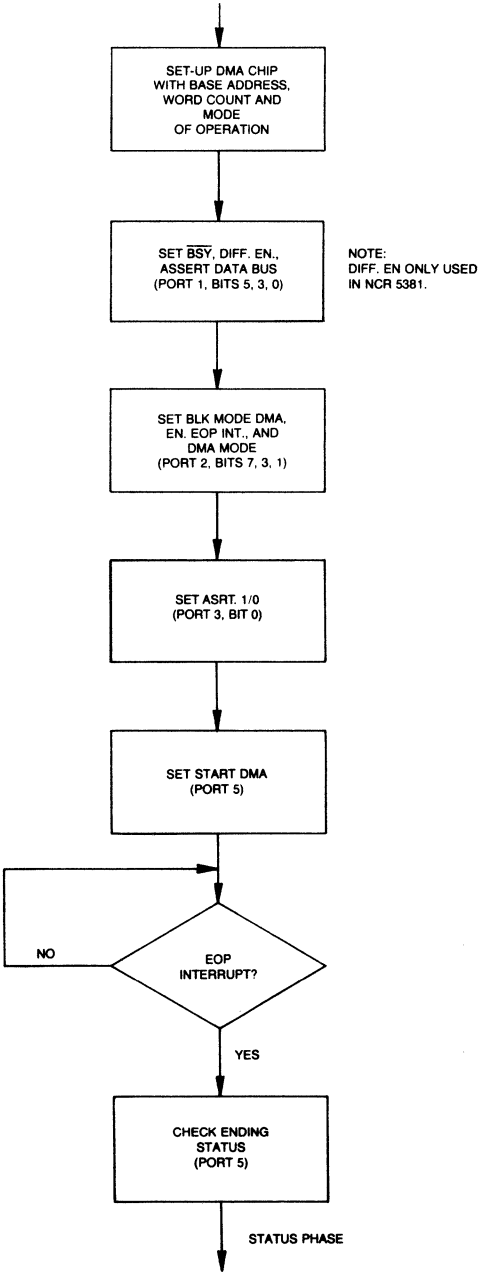
COMMAND TRANSFER PHASE (TARGET)



DATA TRANSFER TO HOST VIA PROGRAMMED I/O



DATA TRANSFER VIA DMA



```
LINE# LOC   CODE        LINE
00001  0000             :  NCR 5380 SCSI PROTOCOL DRIVER
00002  0000             :
00003  0000             ;  SUPPORTS BOTH INITIATOR AND TARGET ROLES
00004  0000             ;
00005  0000             ;  ASSUMES THAT THE COMMAND BLOCK (CDB),
00006  0000             ;  DATA BLOCK(DBLK), AND THE EXPECTED PHASE
00007  0000             ;  TABLE HAVE BEEN SPECIFIED IN MEMORY
00008  0000             ;
00009  0000             SLFAIL=$01              ; SELECTION FAILED STATUS
00010  0000             DISCNT=$02              ; DISCONNECTED STATUS
00011  0000             PRTYER=$03              ; PARITY ERROR STATUS
00012  0000             BUSRST=$04              ; SCSI BUS RESET STATUS
00013  0000             CHIPFL=$05              ; CHIP FAILURE STATUS
00014  0000             MESSAG=$06              ; MESSAGE IN BYTE BEING RETND
00015  0000             DIFFPH=$07              ; UNEXPECTED PHASE REQUESTED
00016  0000             ;
00017  0000             CMDCPL=$00              ; COMMAND COMPLETE MESSAGE
00018  0000             ;
00019  0000             DATA0=$00               ; DATA OUT PHASE
00020  0000             CMD=$08                 ; COMMAND PHASE
00021  0000             STATUS=$0C              ; STATUS PHASE
00022  0000             DATAI=$04               ; DATA IN PHASE
00023  0000             MESSO=$18               ; MESSAGE OUT PHASE
00024  0000             MESSI=$1C               ; MESSAGE IN PHASE
00025  0000             DISCON=$80              ; FLAG TO DISCONNECT
00026  0000             SELECT=$40              ; FLAG TOWAIT FOR SELECTION
00027  0000             ;
00028  0000             S5380=$DE00             ; 5380 ADDRESS SPACE
00029  0000             SDMA=$DE0C              ; PSEUDO DMA ADDRESS
00030  0000             IID=$DE08               ; INIT. ID EXT. LATCH
00031  0000             SRST=$DF00              ; NCR5380 DEVICE RESET
00032  0000             BPNTR=$FB               ; DATA BLOCK POINTER
00033  0000             ;
00034  0000             ;*=BPNTR                   ;
00035  0000             ;
00036  0000             ;DATAB .WORD DBLK           ;
00037  0000             ;
00038  0000             ;
00039  0000             *=$C000                 ; PROGRAM SPACE  ORIGIN
00040  C000             TID     *=*+1               ; TARGET ID SPACE
00041  C001             ICRVAL  *=*+1             ; INIT. CMD REG. STORAGE
00042  C002             INITFL  *=*+1              ; INITIATOR FLAG
00043  C003             OCFLAG  *=*+1              ; OPEN COLL. FLAG
00044  C004             PTYFLG  *=*+1              ; PARITY FLAG
00045  C005             ATNFLG  *=*+1              ; ATN FLAG
00046  C006             ;
00047  C006             PHSIDX  *=*+1              ; EXPECTED PHASE INDEX
00048  C007             XPTPHS  *=*+30             ; EXPECTED PHASE TABLE
00049  C025             ;
00050  C025             COUNT   *=*+1              ; BYTE COUNT
00051  C026             XCNT    *=*+1             ; BYTE COUNT MULTIPLIER
00052  C027             ;
00053  C027             CDB     *=*+12             ; CMD BLOCK STORAGE
00054  C033             DBLK    *=*+512            ; DATA BLOCK
00055  C233             STAT    *=*+2              ; STATUS BYTES
```

```
LINE# LOC   CODE        LINE

00056  C235  27 C0       CDBS   .WORD CDB,DBLK,STAT
00056  C237  33 C0
00056  C239  33 C2
00057  C23B              ;
00058  C23B              ; INITIALIZATION
00059  C23B              ;
00060  C23B  A9 00       START  LDA #00          ; ZERO ACCUM
00061  C23D  8D 06 C0          STA PHSIDX       ; INITIALIZE PHASE INDEX
00062  C240  AD 00 DF          LDA SRST         ; RESET 5380NUMBER
00063  C243  A9 04             LDA #%00000100   ; ENABLE MONITOR BSY INT.
00064  C245  AE 04 C0          LDX PTYFLG       ; LOAD PARITY FLAG
00065  C248  F0 02             BEQ NOPTY        ; IF ZERO, NO PARITY
00066  C24A  09 30             ORA #%00110000   ; OR IN CHECK PARITY BITS
00067  C24C  8D 02 DE  NOPTY  STA S5380+2      ; STORE IN MODE REGISTER
00068  C24F              ;
00069  C24F  AE 06 C0          LDX PHSIDX       ; LOAD VALUE OF PHASE INDEX
00070  C252  A9 40             LDA #SELECT      ; GET VALUE OF SEL CMD
00071  C254  DD 07 C0          CMP XPTPHS,X     ; COMPARE W/CURRENT PHASE
00072  C257  D0 03             BNE INIT         ; IF NOT = BEGIN ARBITRATION
00073  C259  4C 8D C3          JMP TARSEL      ; ELSE,WAIT FOR TARGET SELECT
00074  C25C              ;
00075  C25C              ; BEGIN SCSI BUS ARBITRATION
00076  C25C              ;
00077  C25C  AD 02 DE  INIT   LDA S5380+2     ; READ MODE REG.
00078  C25F  29 FE             AND #%11111110  ; MASK ARB BIT
00079  C261  8D 02 DE          STA S5380+2      ;RESET ARBITRATION BIT
00080  C264  AD 08 DE  ARB    LDA IID          ;BEGIN ARBITRATION
00081  C267  8D 00 DE          STA S5380+0      ;LOAD ID INTO ODR
00082  C26A  AD 02 DE          LDA S5380+2      ;READ MODE REG.
00083  C26D  09 01             ORA #%00000001   ;SET ARBITRATION BIT
00084  C26F  8D 82 53          STA $5380+2      ;STORE IN MODE REG.
00085  C272              ;
00086  C272              ; HAS BUS GONE FREE?
00087  C272              ;
00088  C272  2C 01 DE  NFREE  BIT S5380+1      ;BUS FREE?
00089  C275  50 FB             BVC NFREE        ;NO LOOP UNTIL FREE
00090  C277              ;
00091  C277  EA                NOP              ;YES, WAIT AN ARB DELAY (2.2USE
00092  C278  AD 01 DE          LDA S5380+1      ;LOAD INIT CMD REG.
00093  C27B  29 20             AND #%00100000   ;MASK ALL BUT LA BIT
00094  C27D  D0 DD             BNE INIT         ;IF LOST ARB, RESTART
00095  C27F              ;
00096  C27F              ; CHECK FOR HIGHER PRIORITY ID?
00097  C27F              ;
00098  C27F  AD 00 DE          LDA S5380+0      ;LOAD CURRENT DATA REG.
00099  C282  38                SEC              ;SET CARRY BIT
00100  C283  ED 08 DE          SBC IID          ;SUB YOUR ID FROM DATA REG.
00101  C286  F0 08             BEQ WIN          ;IF EQUAL TO ZERO, WIN ARB
00102  C288  38                SEC              ;NOT=, SOMEONE ELSE IS ARB-ING
00103  C289  ED 08 DE          SBC IID          ;SUBTRACT YOUR ID AGAIN
00104  C28C  30 02             BMI WIN          ;IF NEG, YOUR ID WAS HIGHER
00105  C28E  D0 CC             BNE INIT         ;OTHERWISE, RESTART
00106  C290              ;
00107  C290              ; RECHECK LOST ARBITRATION
00108  C290              ;
```

```
LINE# LOC   CODE        LINE

00109  C290  AD 01 DE   WIN    LDA S5380+1        ;LOAD INIT. CMD REG.
00110  C293  29 20              AND #%00100000    ;MASK ALL BUT LA BIT
00111  C295  D0 C5              BNE INIT          ;IF LOST ARB, RESTART
00112  C297                     ;
00113  C297  A9 0C              LDA #%00001100    ;LOAD VALUE TO SET SEL SIGNAL
00114  C299  AE 05 C0           LDX ATNFLG        ;LOAD ATN FLAG
00115  C29C  F0 02              BEQ WOATN         ; IF ZERO, SEL W/O ATN
00116  C29E  09 02              ORA #%00000010    ; OR IN ATN BIT
00117  C2A0  8D 01 DE   WOATN   STA S5380+1       ;TURN ON SEL LINE
00118  C2A3                     ;
00119  C2A3                     ;  WAIT 1.2 USEC
00120  C2A3                     ;
00121  C2A3  EA                 NOP
00122  C2A4  AD 02 C0           LDA INITFL        ;LOAD IN A SOFTWARE FLAG
00123  C2A7  D0 0D              BNE SEL           ;IF FLAG SET, PERFORM INIT. SE
00124  C2A9                     ;
00125  C2A9                     ; ELSE, TARGET RESELECTION
00126  C2A9                     ;
00127  C2A9                     ;
00128  C2A9  AD 02 DE           LDA S5380+2       ;READ MODE REG.
00129  C2AC  09 40              ORA #%01000000     ;ENABLE TARGET MODE
00130  C2AE  8D 02 DE           STA S5380+2       ;SET TARGET MODE
00131  C2B1  A9 01              LDA #$01          ;ENABLE ASSERT I/O
00132  C2B3  8D 03 DE           STA S5380+3       ;SET ASSERT I/O
00133  C2B6                     ;
00134  C2B6  AD 08 DE   SEL     LDA IID          ; LOAD INITIATOR ID
00135  C2B9  0D 00 C0           ORA TID            ; OR IN TARGET ID
00136  C2BC  8D 00 DE           STA S5380+0       ;LOAD INT & TAR ID'S INTO ODR
00137  C2BF                     ;
00138  C2BF                     ; TEST FOR DIFFERENTIAL PAIR
00139  C2BF                     ;
00140  C2BF  A9 05              LDA #%00000101    ;SEL & DATA BUS BITS
00141  C2C1  AE 03 C0           LDX OCFLAG        ;LOAD IN A SOFTWARE FLAG
00142  C2C4  D0 02              BNE OPNCOL        ;IF FLAG SET, OPEN COLLECTOR
00143  C2C6                     ;
00144  C2C6                     ; DIFFERENTIAL PAIR
00145  C2C6                     ;
00146  C2C6  09 20              ORA #%00100000    ; OR IN DIFF. ENBL BITS
00147  C2C8  8D 01 DE   OPNCOL  STA S5380+1      ;SET SEL, DATA BUS, & (DIFF. PA
00148  C2CB  8D 01 C0           STA ICRVAL         ; RETAIN VALUE OF INIT CMD REG
00149  C2CE                     ;
00150  C2CE                     ; RESET ARBITRATION BIT
00151  C2CE                     ;
00152  C2CE  AD 02 DE           LDA S5380+2       ;READ MODE REGISTER
00153  C2D1  29 FE              AND #%11111110    ;MASK ARB BIT
00154  C2D3  8D 02 DE           STA S5380+2       ;RESET ARB BIT
00155  C2D6                                      ;
00156  C2D6                     ; DISABLE THE SEL EN REGISTER TO AVOID A SEL INT.
00157  C2D6                     ;
00158  C2D6  A9 00              LDA #$00          ; ZERO ACCUM.
00159  C2D8  8D 04 DE           STA S5380+4       ; ZERO SELECT ENABLE REG.
00160  C2DB                     ;
00161  C2DB                     ; RELEASE BUSY
00162  C2DB                     ;
00163  C2DB  AD 01 C0           LDA ICRVAL        ; GET INIT CMD REG VALUE
```

```
LINE# LOC   CODE          LINE

00164  C2DE  29 F7          AND #%11110111     ; MASK OUT BSY BIT
00165  C2E0  8D 01 DE       STA S5380+1        ; RESET BSY
00166  C2E3  8D 01 C0       STA ICRVAL         ; RETAIN ICR VALUE
00167  C2E6                ;
00168  C2E6                ; NOW WAIT 400NSEC AND BEGIN LOOKING FOR BSY
00169  C2E6                ;
00170  C2E6  A0 60          LDY #$60           ; LOAD UP X REG FOR COUNTER
00171  C2E8  A2 FF       RELD   LDX #$FF       ; LOAD UP Y REG FOR COUNTER
00172  C2EA  2C 04 DE    STIM   BIT S5380+4    ; SAMPLE BSY BIT
00173  C2ED  70 18          BVS SLECT          ; IF BSY ACTIVE, SELECTED
00174  C2EF                ;
00175  C2EF                ;  WAIT 250 MSEC
00176  C2EF                ;
00177  C2EF  CA              DEX                ; DELAY
00178  C2F0  D0 F8           BNE STIM           ; IF NOT ZERO LOOP
00179  C2F2  88              DEY
00180  C2F3  D0 F3           BNE RELD           ; IF Y NOT ZERO RELOAD X
00181  C2F5                ;
00182  C2F5                ;    SELECTION TIMEOUT
00183  C2F5                ;
00184  C2F5  A9 00           LDA #$00           ;TAR. DID NOT RESPOND TO SEL
00185  C2F7  8D 00 DE       STA S5380+0        ; RESET ID BITS
00186  C2FA  A2 20           LDX #$20           ; LOAD 200 USEC COUNTER
00187  C2FC  2C 04 DE    CHK    BIT S5380+4    ; CHECK BSY AGAIN
00188  C2FF  70 06           BVS SLECT          ; IF SET SELECTION OK
00189  C301  CA              DEX                ;
00190  C302  D0 F8           BNE CHK            ;
00191  C304                ;
00192  C304                ; SELECTION FAILED
00193  C304                ;
00194  C304  A9 01           LDA #SLFAIL        ; LOAD STATUS IN ACCUM.
00195  C306  60              RTS                ; RETURN TO CALLING PRGM
00196  C307                ;
00197  C307                ; SUCCESSFUL (RE)SELECTION
00198  C307                ;
00199  C307  AD 01 C0    SLECT  LDA ICRVAL     ; GET VALUE OF INIT CMD REG.
00200  C30A  AE 02 C0           LDX INITFL     ; GET INIT FLAG
00201  C30D  D0 08              BNE IF         ; IF INITIATOR JUMP
00202  C30F  09 08              ORA #%00001000 ; AND SET BSY IF TARGET.
00203  C311  8D 01 DE           STA S5380+1    ; WRITE TO ICR
00204  C314  8D 01 C0           STA ICRVAL     ; UPDATE PRESENT ICR VALUE
00205  C317  A9 28       IF     LDA #%00101000 ;MASK TO RESET SEL & DATA BUS
00206  C319  2D 01 C0           AND ICRVAL     ; AND WITH ICR VALUE
00207  C31C  8D 01 DE           STA S5380+1    ; RESET SEL & DATA BUS
00208  C31F  8D 01 C0           STA ICRVAL     ; UPDATE NEW ICR VALUE
00209  C322                ;
00210  C322                ;  BEGIN TRANSFERS
00211  C322                ;
00212  C322  CA              DEX                ; DEC INITIATOR FLAG
00213  C323  F0 03           BEQ PDMA           ; IF ZERO, INITIATOR ROLE
00214  C325  4C 2D C4        JMP RES            ; ELSE, TARGET ROLE
00215  C328                ;
00216  C328                ;     INITIATOR  ROLE
00217  C328                ;
00218  C328                ; USE PSEUDO DMA MODE
```

```
LINE# LOC   CODE          LINE

00219  C328                ;
00220  C328  A9 00         PDMA    LDA #DATA0       ; LOAD TCR W/DATA OUT PHASE
00221  C32A  8D 03 DE              STA S5380+3      ;
00222  C32D  AD 02 DE      NXT     LDA S5380+2      ; GET MODE REGISTER
00223  C330  09 02                 ORA #%00000010;OR IN DMA MODE BIT
00224  C332  8D 02 DE              STA S5380+2      ; SET DMA MODE BIT
00225  C335  AE 06 C0              LDX PHSIDX       ; LOAD X W/PHASE INDEX
00226  C338  BD 08 C0              LDA XPTPHS+1,X; GET PHASE COUNT
00227  C33B  8D 25 C0              STA COUNT        ; STORE IN PHASE COUNT BYTE
00228  C33E  BD 09 C0              LDA XPTPHS+2,X; GET COUNT MULTIPLIER
00229  C341  8D 26 C0              STA XCNT         ; STORE IN MULTIPLIER
00230  C344                ;
00231  C344                ;    WAIT FOR PHASE MISMATCH INT.
00232  C344                ;
00233  C344  AD 05 DE      WAIT    LDA S5380+5      ; SAMPLE BUS&STATUS REG.
00234  C347  29 10                 AND #%00010000   ; LOOK FOR INT. REQ.
00235  C349  F0 F9                 BEQ WAIT         ; IF NOT SET, WAIT
00236  C34B                ;
00237  C34B                ; IRQ IS ACTIVE
00238  C34B                ;
00239  C34B  AD 02 DE              LDA S5380+2      ; GET MODE REG.
00240  C34E  29 FD                 AND #%11111101   ; RESET DMA MASK
00241  C350  8D 02 DE              STA S5380+2      ; RESET DMA MODE BIT
00242  C353  AD 05 DE              LDA S5380+5      ; GET BUS & STATUS REG
00243  C356  AD 05 DE              LSR              ; SHIFT RIGHT 3 TIMES
00244  C359  AD 05 DE              LSR              ;
00245  C35C  AD 05 DE              LSR              ;
00246  C35F  B0 15                 BCS EBUSY        ; LOSS OF BUSY ERROR
00247  C361  B0 15 DE              LSR              ; SHIFT
00248  C364  90 76                 BCC PHSMM        ; IF CARRY CLEAR, MISMATCH
00249  C366  90 76 DE              LSR              ; SHIFT TWICE
00250  C369  90 76 DE              LSR              ;
00251  C36C  B0 0B                 BCS EPRTY        ; IF SET, PARITY ERROR
00252  C36E  2C 04 DE              BIT S5380+4      ; GET CURRENT SCSI BUS STATUS
00253  C371  30 14                 BMI BRST         ; IF BIT 7 SET, BUS RESET OCCURE
00254  C373  4C 8A C3              JMP FAIL         ; SHOULD NOT GET HERE
00255  C376                ;
00256  C376                ;  RETURN ERROR STATUS TO CALLING PROGRAM
00257  C376                ;
00258  C376  A9 02         EBUSY   LDA #DISCNT      ; SET DISCONNECT FLAG
00259  C378  60                    RTS
00260  C379  AD 01 C0      EPRTY   LDA ICRVAL       ; GET INIT. CMD REG. VALUE
00261  C37C  09 02                 ORA #%00000010   ; TURN ON ATN SIGNAL
00262  C37E  8D 01 DE              STA S5380+1      ; SET ATN
00263  C381  8D 01 C0              STA ICRVAL
00264  C384  A9 03                 LDA #PRTYER      ; SET PARITY ERROR
00265  C386  60                    RTS
00266  C387  A9 04         BRST    LDA #BUSRST      ; SET BUS RESET ERROR
00267  C389  60                    RTS
00268  C38A  A9 05         FAIL    LDA #CHIPFL      ; SET CHIP FAIL ERROR
00269  C38C  60                    RTS              ; RETURN TO CALLING PRGM
00270  C38D                ;
00271  C38D                ; WAIT FOR TARGET SELECTION
00272  C38D                ;
00273  C38D  AD 02 DE      TARSEL  LDA S5380+2      ; GET MODE REG.
```

```
LINE# LOC   CODE        LINE

00274  C390  09 40           ORA #%01000000  ; SET TARGET MODE MASK
00275  C392  8D 02 DE        STA S5380+2     ; SET TARGET MODE BIT
00276  C395  AD 08 DE        LDA IID         ; GET TARGET ID
00277  C398  8D 04 DE        STA S5380+4     ; STORE IN SELECT ENABLE REG.
00278  C39B  AD 05 DE  LOOK  LDA S5380+5     ; SAMPLE BUS&STATUS REG.
00279  C39E  29 10           AND #%00010000  ; LOOK FOR INT REQ
00280  C3A0  F0 F9           BEQ LOOK        ; KEEP WAITING
00281  C3A2                  ;
00282  C3A2                  ;
00283  C3A2                  ;   CHECK FOR MORE THAN TWO ID'S ACTIVE
00284  C3A2                  ;
00285  C3A2  AD 00 DE        LDA S5380+0     ; READ SCSI DATA BUS
00286  C3A5  A2 09           LDX #$09        ; SHIFT COUNT
00287  C3A7  A0 00           LDY #$00        ; INITIALIZE BIT COUNT
00288  C3A9  A0 00 DE  UP    LSR             ; SHIFT BIT INTO CARRY BIT
00289  C3AC  CA              DEX             ; DECR. SHIFT COUNT
00290  C3AD  F0 05           BEQ OUT         ; IF ZERO, DONE COUNTING
00291  C3AF  90 F8           BCC UP          ; IF CARRY NOT SET, DO NEXT
00292  C3B1  C8              INY             ; IF CARRY SET BUMP BIT CNT
00293  C3B2  B0 F5           BCS UP          ; GET NEXT BIT
00294  C3B4  38        OUT   SEC             ; SET CARRY BIT
00295  C3B5  98              TYA             ; PUT Y IN ACCUM
00296  C3B6  E9 03           SBC #$03        ; SUBTRACT 3 FROM BIT COUNT
00297  C3B8  30 06           BMI CI          ; IF MINUS, OK
00298  C3BA  AD 07 DE        LDA S5380+7     ; NOT MINUS, RESET IRQ.
00299  C3BD  4C 9B C3        JMP LOOK        ; WAIT FOR GOOD SELECTION
00300  C3C0                  ;
00301  C3C0                  ; CHECK INTERRUPT
00302  C3C0                  ;
00303  C3C0  AD 05 DE  CI    LDA S5380+5     ; SAMPLE AGAIN
00304  C3C3  29 20           AND #%00100000  ; MASK PARITY BIT
00305  C3C5  F0 B2           BEQ EPRTY       ; PARITY SELECTION ERROR
00306  C3C7  AD 04 DE        LDA S5380+4     ; GET CURRENT SCSI BUS ST.
00307  C3CA  29 02           AND #%00000010  ; CHECK SEL
00308  C3CC  F0 BC           BEQ FAIL        ; IF NOT SET, FAILURE
00309  C3CE  A9 08           LDA #%00001000  ; SET BSY MASK
00310  C3D0  8D 01 DE        STA S5380+1     ; SET BSY SEL COMPLETE
00311  C3D3  8D 01 C0        STA ICRVAL        ; RETAIN ICR VALUE
00312  C3D6  AD 07 DE        LDA S5380+7     ; RESET INTERRUPT
00313  C3D9  4C 2D C4        JMP RES           ;
00314  C3DC                  ;
00315  C3DC                  ; PHASE MISMATCH CONDITION
00316  C3DC                  ;
00317  C3DC  AE 06 C0  PHSMM LDX PHSIDX      ; LOAD X WITH PHASE POINTER
00318  C3DF  AD 04 DE        LDA S5380+4     ; LOAD CURRENT SCSI BUS STATUS
00319  C3E2  29 1C           AND #%00011100  ; MASK ALL BUT PHASE BITS
00320  C3E4  DD 07 C0        CMP XPTPHS,X    ; COMPARE TO XPTED PHASE
00321  C3E7  F0 03           BEQ PHSMTH      ; YES, PHASE MATCHES
00322  C3E9  4C B4 C4        JMP DP          ; ELSE, DIFFERENT PHASE
00323  C3EC                  ;
00324  C3EC                  ; PHASE MATCHES EXPECTED PHASE
00325  C3EC                  ;
00326  C3EC  4C B4 C4  PHSMTH LSR             ; SHIFT TO TCR REG. FORMAT
00327  C3EF  4C B4 C4        LSR             ;
00328  C3F2  8D 03 DE        STA S5380+3     ; STORE IN TCR
```

```
LINE# LOC   CODE        LINE

00329  C3F5  AD 07 DE          LDA S5380+7      ; RESET INTERRUPT
00330  C3F8  A9 18             LDA #MESSO       ; LOAD MESSOUT VALUE
00331  C3FA  DD 07 C0          CMP XPTPHS,X     ; WAS PHASE MATCH MESS. OUT
00332  C3FD  D0 0B             BNE GMR          ; IF NOT MESS. OUT, CONTINUE
00333  C3FF                ;
00334  C3FF                ; MESSAGE OUT,    RESET ATN
00335  C3FF                ;
00336  C3FF  AD 01 C0          LDA ICRVAL       ; GET INITIATOR CMD. REG
00337  C402  29 FD             AND #%11111101   ; MASK OFF ATN
00338  C404  8D 01 DE          STA S5380+1      ; TURN OFF ATN
00339  C407  8D 01 C0          STA ICRVAL       ; UPDATE ICR VALUE
00340  C40A  AD 02 DE     GMR  LDA S5380+2      ; GET MODE REG
00341  C40D  09 02             ORA #%00000010   ; SET DMA MODE BIT
00342  C40F  8D 02 DE          STA S5380+2      ; STORE IN TCR
00343  C412  AD 03 DE          LDA S5380+3      ; GET PHASE AGAIN
00344  C415  29 01             AND #%00000001   ; SET I/O MASK
00345  C417  F0 0B             BEQ IDMAO        ; IF ZERO, DMA OUTPUT
00346  C419                ;
00347  C419                ; INITIATOR DMA INPUT
00348  C419                ;
00349  C419  8D 07 DE     IDMAI  STA S5380+7    ; START INIT. RCV.
00350  C41C  A0 00             LDY #00          ; INITIALIZE Y
00351  C41E  20 FE C4          JSR DMAIN        ; PERFORM DMA INPUT
00352  C421  4C 2D C3          JMP NXT          ; PREPARE FOR NEXT PHASE
00353  C424                ;
00354  C424                ; INITIATOR DMA OUTPUT
00355  C424                ;
00356  C424  20 27 C5     IDMAO  JSR DMAOUT     ; PERFORM DMA OUTPUT
00357  C427  8D 0C DE          STA SDMA         ; EXTRA WRITE FOR ACK TO GO OFF
00358  C42A  4C 2D C3          JMP NXT          ; PREPARE FOR NEXT PHASE
00359  C42D                ;
00360  C42D                ; TARGET OPERATION
00361  C42D                ;
00362  C42D  AD 02 DE     RES  LDA S5380+2      ; GET MODE REGISTER
00363  C430  09 02             ORA #%00000010;OR IN DMA MODE BIT
00364  C432  8D 02 DE          STA S5380+2      ; SET DMA MODE BIT
00365  C435  AE 06 C0          LDX PHSIDX       ; LOAD X W/PHASE INDEX
00366  C438  BD 08 C0          LDA XPTPHS+1,X; GET PHASE COUNT
00367  C43B  8D 25 C0          STA COUNT        ; STORE IN PHASE COUNT BYTE
00368  C43E  BD 09 C0          LDA XPTPHS+2,X; GET COUNT MULTIPLIER
00369  C441  8D 26 C0          STA XCNT         ; STORE IN MULTIPLIER
00370  C444                ;
00371  C444  AD 05 DE          LDA S5380+5      ; GET BUS & STATUS REG.
00372  C447  29 02             AND #%00000010   ; MASK ATN BIT
00373  C449  D0 4D             BNE MESSOT       ; ATN ACTIVE DO MESS OUT PHASE
00374  C44B  AE 06 C0          LDX PHSIDX       ; GET CURRENT PHASE INDEX
00375  C44E  A9 80             LDA #DISCON      ; GET DISCONNECTED VALUE
00376  C450  DD 07 C0          CMP XPTPHS,X     ; COMPARE W/PHASE VALUE
00377  C453  F0 3B             BEQ DISCTD       ; IF =, TIME TO DISCONNECT
00378  C455  BD 07 C0          LDA XPTPHS,X     ; GET PHASE
00379  C458  BD 07 C0          LSR              ; SHIFT TO TCR FORMAT
00380  C45B  BD 07 C0          LSR              ;
00381  C45E  8D 03 DE          STA S5380+3      ; STORE IN TARGET COMMAND REG.
00382  C461  29 01             AND #%00000001   ; SAVE I/O BIT
00383  C463  F0 1D             BEQ TDMAO        ; IF ZERO, DMA OUTPUT
```

```
LINE# LOC   CODE        LINE

00384  C465                ;
00385  C465                ; TARGET DMA INPUT
00386  C465                ;
00387  C465  8D 06 DE     TDMAI  STA S5380+6      ; START DMA TARGET RCV
00388  C468  A0 01                LDY #01         ; SET Y TO ONE,SO NO EXTRA REQ
00389  C46A  20 FE C4             JSR DMAIN       ; PERFORM DMA INPUT
00390  C46D                ;
00391  C46D                ;      HANDLE LAST BYTE TO PREVENT EXTRA REQ
00392  C46D                ;
00393  C46D  2C 05 DE     LSTDRQ BIT S5380+5      ; LOOK FOR DRQ
00394  C470  50 FB                BVC LSTDRQ      ; LOOP TILL ON
00395  C472  AD 02 DE             LDA S5380+2     ; GET MODE REG.
00396  C475  29 FD                AND #%11111101  ; MASK DMA MODE BIT
00397  C477  8D 02 DE             STA S5380+2     ; RESET DMA MODE BIT
00398  C47A  AD 0C DE             LDA SDMA        ; GET LAST BYTE FROM CHIP
00399  C47D  91 FB                STA (BPNTR),Y   ; STORE LAST BYTE
00400  C47F  4C 2D C4             JMP RES         ; DO NEXT PHASE
00401  C482                ;
00402  C482                ; TARGET DMA OUTPUT
00403  C482                ;
00404  C482  20 27 C5     TDMAO  JSR DMAOUT      ; PERFORM DMA OUTPUT
00405  C485  AD 02 DE             LDA S5380+2     ; GET DMA MODE
00406  C488  29 FD                AND #%11111101  ; MASK DMA MODE BIT
00407  C48A  8D 02 DE             STA S5380+2     ; RESET DMA MODE BIT
00408  C48D  4C 2D C4             JMP RES         ; DO NEXT PHASE
00409  C490                ;
00410  C490                ; TARGET DISCONNECT
00411  C490                ;
00412  C490  A9 00        DISCTD LDA #00         ; LOAD ACCUM W/ ZERO
00413  C492  8D 01 DE             STA S5380+1     ; RESET BSY & OTHER SIGNALS
00414  C495  A9 02                LDA #DISCNT     ; DISCONNECTED STATUS
00415  C497  60                   RTS             ; RETURN TO CALLING PRGM
00416  C498                ;
00417  C498                ; MESSOUT PHASE (TARGET)
00418  C498                ;
00419  C498  A9 18        MESSOT LDA #MESSO      ; GET VALUE OF MESSAGE OUT
00420  C49A  A9 18 DE             LSR             ; SHIFT TO TCR FORMAT
00421  C49D  A9 18 DE             LSR             ;
00422  C4A0  8D 03 DE             STA S5380+3     ; MESSOUT PHASE
00423  C4A3  A9 01                LDA #1          ; LOAD MULTPLIER/COUNTER VALUE
00424  C4A5  CA                   DEX             ; MOVE POINTER
00425  C4A6  9D 07 C0             STA XPTPHS,X    ; STORE MULTIPLIER
00426  C4A9  CA                   DEX             ; MOVE POINTER TO COUNT VALUE
00427  C4AA  9D 07 C0             STA XPTPHS,X    ; STORE COUNT
00428  C4AD  CA                   DEX             ; MOVE TO PHASE
00429  C4AE  8E 06 C0             STX PHSIDX      ; UPDATE MOVED PHASE INDEX
00430  C4B1  4C 82 C4             JMP TDMAO       ; DO DMA OUT
00431  C4B4                ;
00432  C4B4                ; DIFFERENT PHASE
00433  C4B4                ;
00434  C4B4  A9 1C        DP     LDA #MESSI      ; LOAD VALUE OF MESSAGE IN PHASE
00435  C4B6  DD 07 C0             CMP XPTPHS,X    ; IS THIS A MESSAGE IN PHASE
00436  C4B9  F0 03                BEQ MESSIN      ; IF=, READ MESSAGE
00437  C4BB  A9 07                LDA #DIFFPH     ; LOAD DIFFERENT PHASE ST.
00438  C4BD  60                   RTS             ; RETN W/UNEXPECTED PHASE STATUS
```

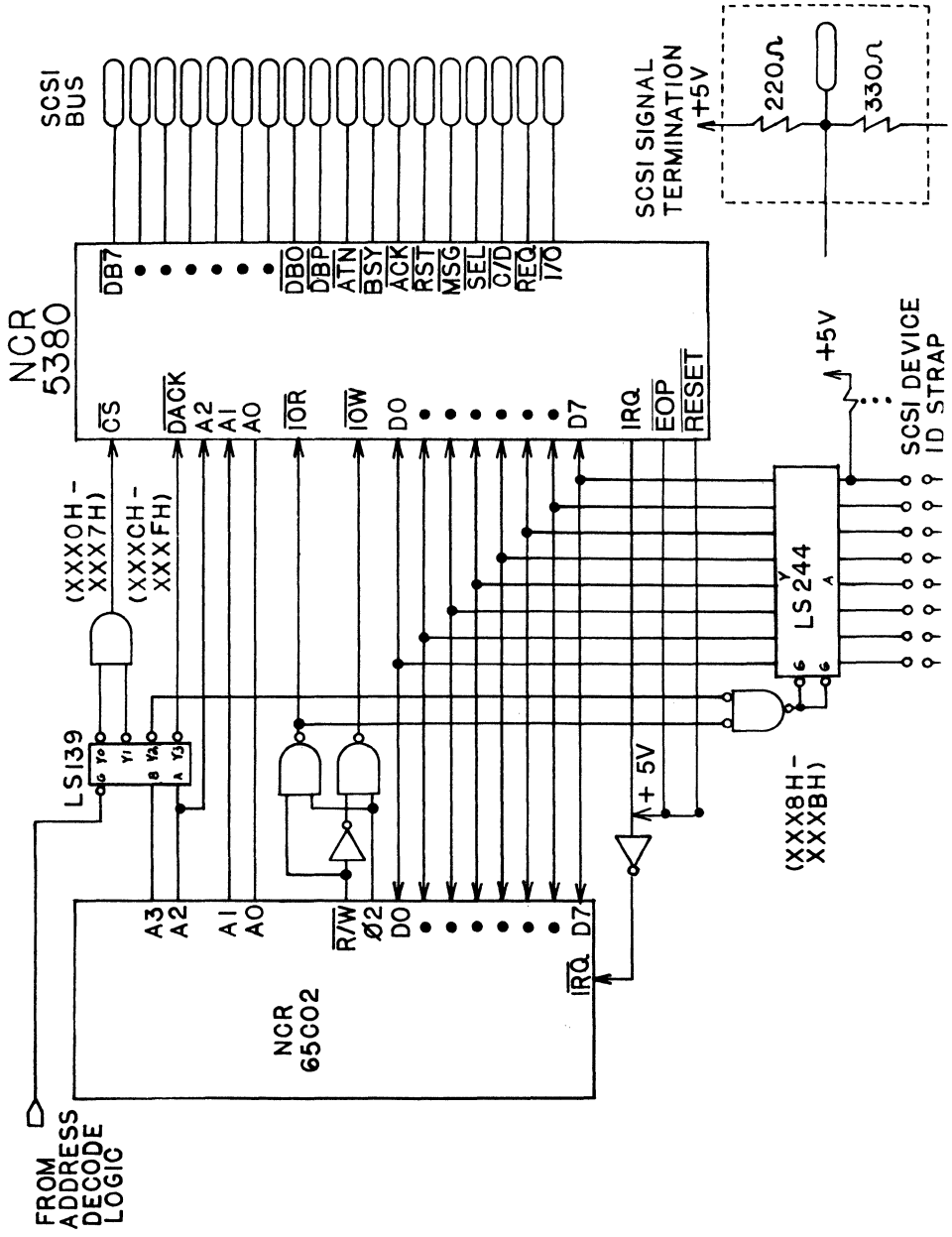
```
LINE# LOC   CODE        LINE

00439  C4BE             ;
00440  C4BE             ; MESSAGE IN PHASE
00441  C4BE             ;
00442  C4BE  60 07 C0   MESSIN LSR              ; SHIFT TO TCR FORMAT
00443  C4C1  60 07 C0          LSR              ;
00444  C4C4  8D 03 DE          STA S5380+3      ; LOAD TCR
00445  C4C7  AD 07 DE          LDA S5380+7      ; RESET INT.
00446  C4CA  AD 04 DE   POLL   LDA S5380+4      ; READ CURRENT BUS STATUS
00447  C4CD  29 20             AND #%00100000   ; LOOK FOR REQ.
00448  C4CF  F0 F9             BEQ POLL         ; IF ZERO, NO REQ.
00449  C4D1  AD 01 C0          LDA ICRVAL       ; GET CURRENT ICR VALUE
00450  C4D4  09 10             ORA #%00010000   ; OR IN ASSERT ACK
00451  C4D6  8D 01 DE          STA S5380+1      ; ASSERT ACK
00452  C4D9  8D 01 C0          STA ICRVAL       ; UPDATE ICR
00453  C4DC  AD 04 DE   STILON LDA S5380+4      ; READ CURRENT BUS STATUS
00454  C4DF  29 20             AND #%00100000   ; LOOK FOR NOT REQ
00455  C4E1  D0 F9             BNE STILON       ; IF NOT ZERO, STILL ON
00456  C4E3  A9 00             LDA #CMDCPL      ; LOAD COMMAND COMPLETE
00457  C4E5             ;
00458  C4E5             ; LEAVE ACK ACTIVE SO MESSAGE CAN BE REJECTED
00459  C4E5             ;
00460  C4E5  CD 06 DE          CMP S5380+6      ; COMPARE W/MESSAGE
00461  C4E8  D0 0E             BNE DIFMES       ; IF NOT CMD COMPLTE, DIFF.
00462  C4EA  AD 01 C0          LDA ICRVAL       ; GET ICR VAL
00463  C4ED  29 EF             AND #%11101111   ; MASK ACK BIT
00464  C4EF  8D 01 DE          STA S5380+1      ; RESET ACK
00465  C4F2  8D 01 C0          STA ICRVAL       ; UPDATE ICR
00466  C4F5  4C 2D C3          JMP NXT          ; GO TO NEXT PHASE
00467  C4F8             ;
00468  C4F8             ; NOT MESSAGE COMPLETE, RETURN FOR EVALUATION
00469  C4F8             ;
00470  C4F8  AE 06 DE   DIFMES LDX S5380+6      ; GET MESSAGE VALUE
00471  C4FB  A9 06             LDA #MESSAG      ; LOAD MESSAGE RETN STATUS
00472  C4FD  60                RTS              ; RETURN FOR MESSAGE EVALUATION
00473  C4FE             ;
00474  C4FE             ;
00475  C4FE             ; DMA INPUT
00476  C4FE             ;
00477  C4FE  A9 00      DMAIN  LDA #00          ; ZERO ACCUM.
00478  C500  AA                TAX              ; ZERO X
00479  C501             ;
00480  C501             ; RESET ASSERT DATA BUS
00481  C501             ;
00482  C501  AD 01 C0          LDA ICRVAL       ; GET ICR VALUE
00483  C504  29 FE             AND #%11111110   ; MASK ASSERT DATA BUS
00484  C506  8D 01 DE          STA S5380+1      ; RESET ASSERT DATA BUS BIT
00485  C509  8D 01 C0          STA ICRVAL       ; UPDATE ICR
00486  C50C             ;
00487  C50C             ; WAIT FOR DRQ
00488  C50C             ;
00489  C50C  2C 05 DE   REPT1  BIT S5380+5      ; TEST FOR DRQ
00490  C50F  50 FB             BVC REPT1        ; IF NOT THERE, LOOP
00491  C511             ;
00492  C511  AD 0C DE   GO1    LDA SDMA         ; READ DMA PORT
00493  C514  91 FB             STA (BPNTR),Y    ; STORE DATA IN BUFFER
```

```
LINE# LOC   CODE        LINE

00494  C516  C8              INY             ; INCR. POINTER
00495  C517  CC 25 C0        CPY COUNT       ; DONE?
00496  C51A  D0 F0           BNE REPT1       ; IF NOT ZERO, REPEAT
00497  C51C  E8              INX             ; ZERO, CHECK MULTIPLIER
00498  C51D  EC 26 C0        CPX XCNT        ; COMPARE X WITH MULTIPLIER
00499  C520  F0 32           BEQ NXTPHS      ; IF EQUAL, COUNT DONE
00500  C522  E6 FC     BUMP   INC BPNTR+1     ; GREATER THAN 256 BYTES BUMP MSB
00501  C524  4C 11 C5        JMP GO1         ; GET MORE BYTES
00502  C527                 ;
00503  C527                 ;  DMA OUTPUT
00504  C527                 ;
00505  C527  A9 01     DMAOUT LDA #%00000001  ; SET MASK
00506  C529  0D 01 C0        ORA ICRVAL      ; OR WITH ICR VALUE
00507  C52C  8D 01 DE        STA S5380+1     ; SET ASSERT DATA BUS BIT
00508  C52F  8D 01 C0        STA ICRVAL      ; UPDATE ICR VALUE
00509  C532  8D 05 DE        STA S5380+5     ; START DMA INIT SEND
00510  C535                 ;
00511  C535                 ; LOOK FOR DMA REQ (DRQ)
00512  C535                 ;
00513  C535  A9 00           LDA #00         ; ZERO ACCUM
00514  C537  A8              TAY             ; ZERO Y
00515  C538  AA              TAX             ; ZERO X
00516  C539  2C 05 DE  REPT   BIT S5380+5     ; SAMPLE DRQ
00517  C53C  50 FB           BVC REPT        ; IF NOT SET, REPEAT
00518  C53E  B1 FB     GO     LDA (BPNTR),Y   ; GET BYTE FROM BLOCK
00519  C540  8D 0C DE        STA SDMA        ; WRITE BYTE TO CHIP
00520  C543  C8              INY             ; INC  Y POINTER
00521  C544  CC 25 C0        CPY COUNT       ; COMPARE WITH BYTE CNT
00522  C547  D0 F0           BNE REPT        ; IF Y NOT EQ. SEND MORE
00523  C549  E8              INX             ; IF EQUAL INCR. X
00524  C54A  EC 26 C0        CPX XCNT        ; COMPARE W/ MULTIPLIER
00525  C54D  F0 05           BEQ NXTPHS      ; IF EQ, NEXT PHASE
00526  C54F  E6 FC           INC BPNTR+1     ; MORE THAN 256 BUMP MSB
00527  C551  4C 3E C5        JMP GO          ; SEND MORE DATA
00528  C554                 ;
00529  C554                 ;   NEXT PHASE
00530  C554                 ;
00531  C554  EE 06 C0  NXTPHS INC PHSIDX     ; PHASE POINTER INDEX +3
00532  C557  EE 06 C0        INC PHSIDX      ;
00533  C55A  EE 06 C0        INC PHSIDX      ;
00534  C55D  60              RTS             ; RETN TO INIT OR TRGT OPER.
```

A3. 65C02 Interface Diagram



A4. NCR 5381—Differential Pair Option

The NCR 5381 is a 48 pin version of the NCR 5380 device designed to support external differential pair transceivers. These external transceivers are controlled with the additional signals provided in the higher pinout package. The NCR 5381 may still operate at a single-ended device if the SINGLEND signal (pin 2) is active. In single-ended operation, the signals provided for differential support remain functional.

The use of the DIFFERENTIAL ENABLE bit (bit 5) in the Initiator Command Register reflects the only software difference between the two parts. When active, this bit is used to assert the signals IGS (pin 18) or TGS (pin 14) depending on the status of the TARGET MODE bit (port 2, bit 6). (IGS is active if TARGET MODE is false and TGS is active if TARGET MODE is true.) As in the NCR 5385/86, IGS is used to enable the external drivers for the signals $\overline{ACK}$ (pin 17) and $\overline{ATN}$ (pin 18) and TGS is used to enable the external drivers for the signals $\overline{I/O}$ (pin 20), $\overline{C/D}$ (pin 21), $\overline{MSG}$ (pin 22) and $\overline{REQ}$ (pin 23).

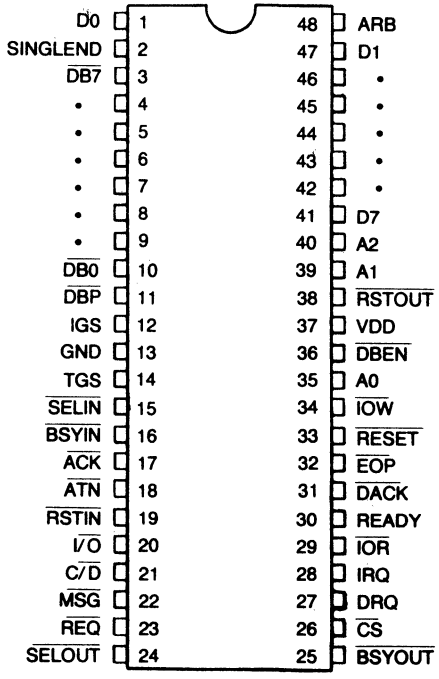
The signal differences between the NCR 5380 and the NCR 5381 are as follows:

Pin	SIGNAL NAME	DESCRIPTION
2	SINGLEND	This signal, when active (1) selects the single-ended mode of operation. When inactive, the NCR 5381 operates in the differential pair mode. These signals will change from input/output pins to input only pins if the SINGLEND signal is false.

	SINGLEND = 1	SINGLEND = 0
PIN #16	$\overline{BSY}$	$\overline{BSYIN}$
PIN #15	$\overline{SEL}$	$\overline{SELIN}$
PIN #19	$\overline{RST}$	$\overline{RSTIN}$

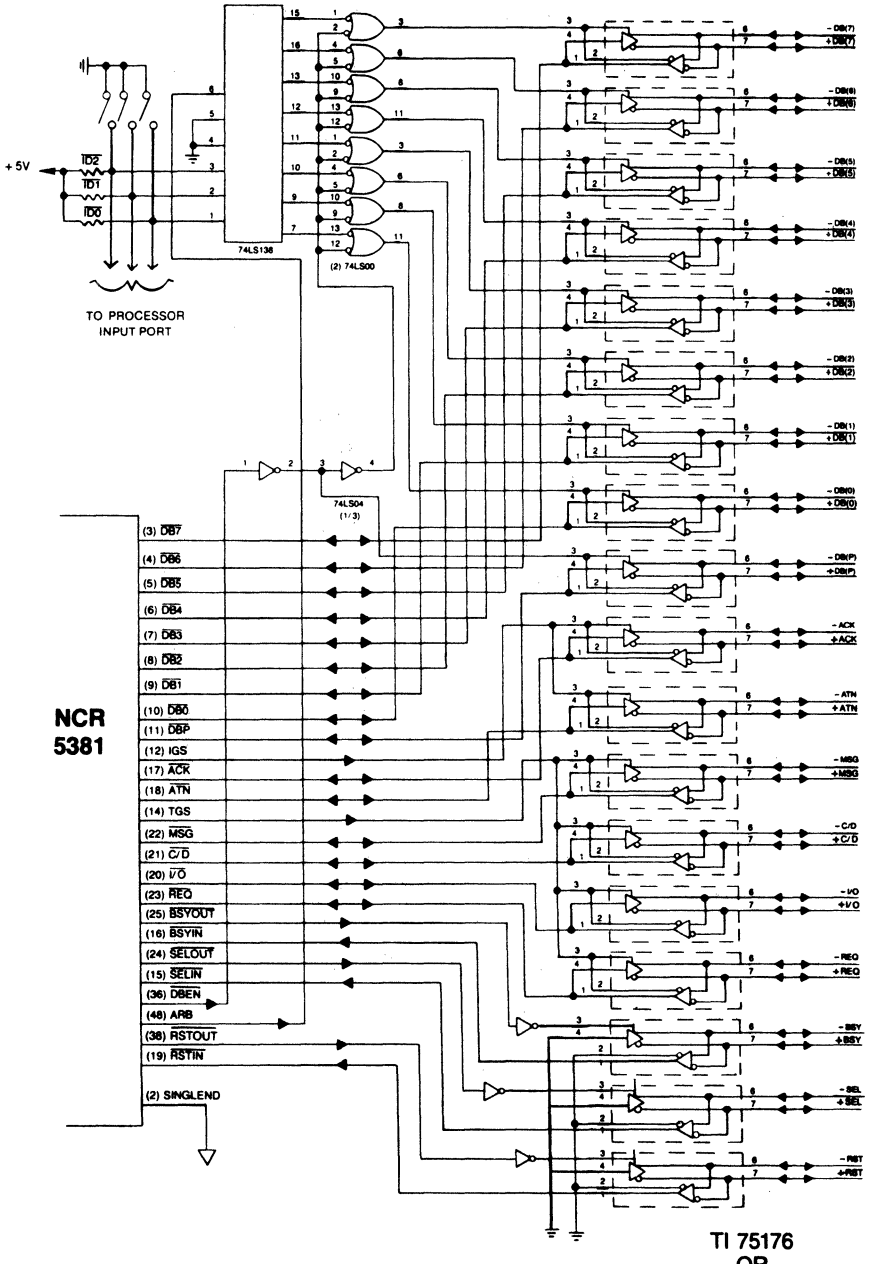
36	$\overline{DBEN}$	This signal is asserted whenever the ASSERT DATA BUS bit (port 1, bit 0) and the TARGET MODE bit are set (1). It is also asserted when ASSERT DATA BUS and PHASE MATCH are true and both TARGET MODE and $\overline{I/O}$ are false. This signal is used to enable the external transceivers to drive the data bus.
14	TGS	This signal is active when the TARGET MODE bit and the DIFFERENTIAL ENABLE bit are true. It is used to enable the external transceivers to drive $\overline{I/O}$, $\overline{C/D}$, $\overline{MSG}$, and $\overline{REQ}$.
12	IGS	This signal is active when the TARGET MODE bit is false and the DIFFERENTIAL ENABLE bit is true. It is used to enable the external transceivers to drive $\overline{ACK}$ and $\overline{ATN}$.
48	ARB	The NCR 5380 chip asserts this signal when the ARBITRATION bit is set and the device has detected a bus free condition. It is used to assert the proper device ID on the bus during the arbitration phase.
25	$\overline{BSYOUT}$	This signal is active whenever $\overline{BSY}$ is asserted. This signal will be inactive at all other times.

Pin	SIGNAL NAME	DESCRIPTION
24	$\overline{\text{SELOUT}}$	This signal is active if the ASSERT $\overline{\text{SEL}}$ bit is true. Conversely, this signal is inactive if the ASSERT $\overline{\text{SEL}}$ bit is reset.
38	$\overline{\text{RSTOUT}}$	This signal is active if the ASSERT $\overline{\text{RST}}$ bit is true. Conversely, this signal is inactive if the ASSERT $\overline{\text{RST}}$ bit is reset.



NCR 5381 PINOUT

NCR 5381 Suggested Differential Pair Interface



NOTE: DB0-7, DBP, ACK, ATN, MSG, C/D, I/O, and REQ require pull-ups.

A5. NCR 53C80 Product Information

The NCR 53C80 CMOS SCSI interface device is functionally equivalent to the NMOS NCR 5380 device. However, this chip does not offer pin compatibility.

CMOS and NMOS are entirely different processes with inherently different device characteristics. All CMOS devices are designed to prohibit a condition commonly referred to as "latch-up." When the high current 48 ma drivers in the NCR 53C80 switch, noise level is sufficiently high, the output drivers become more prone to "latch-up." Therefore, for the CMOS design, pin compatibility has been foresaken for reliability. Four additional ground lines have been added to increase design stability. Having additional ground signals also allows the output buffers to switch more quickly, creating the added benefit of a faster design. This does not imply that the NMOS device is marginal in any respect.

DIFFERENCES:

To make the device easier to design with, the NCR 53C80 contains some minor modifications. An explanation of each is listed below.

Spurious RST Interrupt

If the NCR 5380 is not terminated on the SCSI bus, the floating condition on the input of the RST signal can generate spurious interrupts.

The NCR 53C80 contains an internal 30 uamp pull-up on the RST signal to prevent an interrupt due to an un-terminated bus.

True End Of DMA Status For Send Operations

If sending data to the NCR 5380 and EOP is asserted on the last byte, the END OF DMA status bit indicates only that the last byte has been received from the DMA device. There is no indication that this byte has been transferred to the SCSI bus.

The NCR 53C80 uses bit 7 of the Target Command Register to indicate that the last byte of a DMA transfer has been sent to the SCSI bus.

Improved REQ/ACK Transition Times

The NCR 53C80 has improved the response times of the REQ/ACK handshake signals. This has an overall impact on achievable transfer rates. This performance improvement has been attained by increasing the number of ground lines, using faster CMOS cells, and giving cell placement priority to the REQ/ACK signal paths.

Prevents The Possibility Of An Additional ACK From Occurring

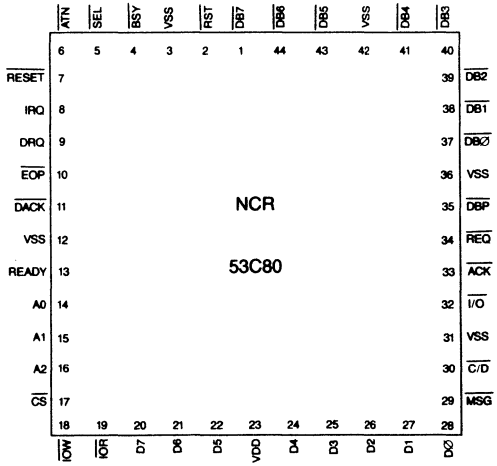
The NCR 5380, upon receipt of an EOP signal, sets the END OF DMA status bit and prevents additional DMA requests (DRQ). It does not reset the DMA MODE bit. If receiving data as an Initiator and the Target continues to request data for the same bus phase, after receiving an EOP pulse, the NCR 5380 will assert ACK without issuing a DMA request (DRQ).

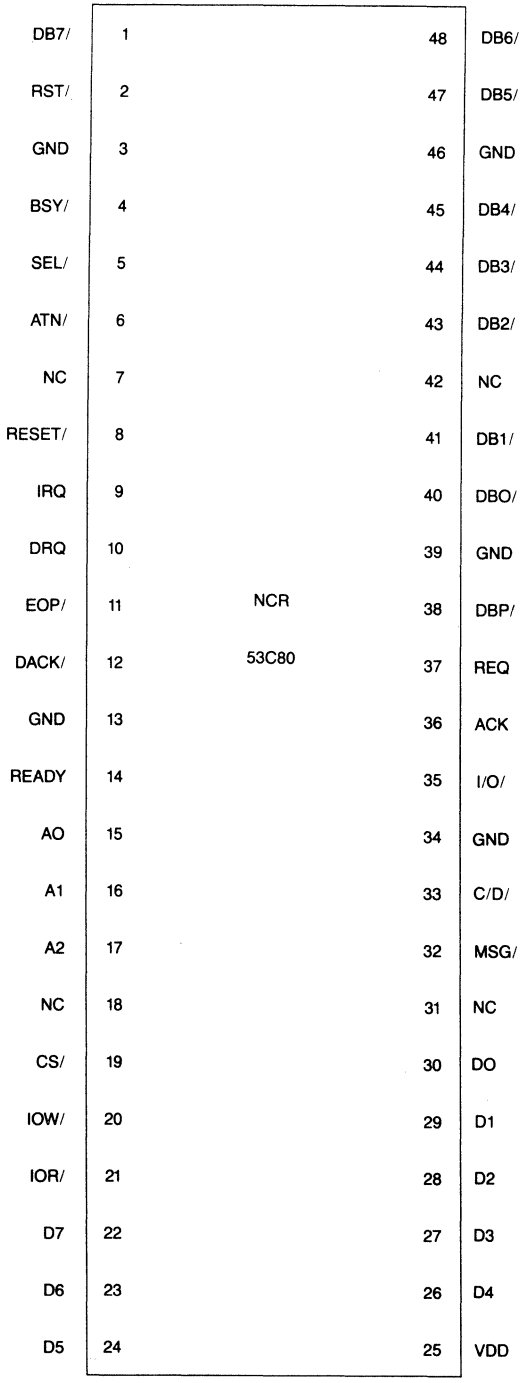
The NCR 53C80 prevents ACK from being asserted until the device is instructed to continue by writing the Start DMA Initiator Receive register.

PERFORMANCE:

Even though early material on the NCR 53C80 has demonstrated better performance than the NCR 5380, device characterization was not completed before this section was printed.

PINOUT:




DB7/ 1
RST/ 2
GND 3
BSY/ 4
SEL/ 5
ATN/ 6
NC 7
RESET/ 8
IRQ 9
DRQ 10
EOP/ 11
DACK/ 12
GND 13
READY 14
AO 15
A1 16
A2 17
NC 18
CS/ 19
IOW/ 20
IOR/ 21
D7 22
D6 23
D5 24
NCR
53C80
48 DB6/
47 DB5/
46 GND
45 DB4/
44 DB3/
43 DB2/
42 NC
41 DB1/
40 DBO/
39 GND
38 DBP/
37 REQ
36 ACK
35 I/O/
34 GND
33 C/D/
32 MSG/
31 NC
30 DO
29 D1
28 D2
27 D3
26 D4
25 VDD

A6. SCSI/PLUS*

AMPRO Computers, Inc. is proposing a general enhancement to the SCSI specification which allows the bus to operate as either a single or multi-master high speed parallel bus, capable of accessing up to 64 modules. This new bus structure is referred to as SCSI/PLUS. The table below describes the types of devices that may now be added due to the enhanced SCSI specification.

SCSI/PLUS DEVICE TYPES	EXAMPLES
Operating System Processors and Co-processors	UNIX MS-DOS CP/M FORTH Lisp Porlog
Communication Servers	Modems Arcnet Ethernet SDLC Mainframe links
Display Controllers	Graphics Text Touch
System Resources	Printer Spooler Time-of-day clock Speech I/O Protocol Converter DBMS Processor Array Processor
Real World Interfaces	A/D D/A AC & DC Control

Examples of SCSI/PLUS Devices

* SCSI/PLUS is a trademark of AMPRO Computers Inc.

SCSI/PLUS provides three functional additions to the SCSI specification which allow the bus to operate as either a loosely coupled distributed system bus or a low-cost single master I/O bus. As proposed, SCSI/PLUS is a superset of the original specification, and its operation will not interfere with any existing SCSI implementation.

To allow for more complex system configurations, SCSI/PLUS provides Binary Arbitration and Binary Selection phases. The data bus represents a binary address and accommodates 64 physical bus devices, compared to eight in the current specification. In addition, four logical units may be associated with each bus device for a total of 256 logical bus devices. As in the SCSI specification, the arbitration phase is optional.

The addition of a master/slave mode to the specification provides for a cost-effective single-master/multi-slave configuration. This mode allows the design of SCSI/PLUS Targets which have no on-board intelligence. An optional interrupt protocol allows these "dumb" targets to asynchronously notify the bus master that they desire service.

To encourage board-level interchangability, a recommended board size and interface connector is defined. The preferred board size is the single-wide Eurocard format with the double-wide card used as an option. The proposed interface connector is the DIN 41612—Type C connector. By using this form-factor and connector specification, bussed backplane or ribbon cable systems may be implemented.

The SCSI/PLUS architectural concept has inherent advantages over traditional microprocessor backplane architectures. SCSI/PLUS is CPU-independent, provides flexibility of form factor, operates across a ribbon cable bus, and allows both high-performance multimaster and low-cost single-master operation.

The NCR 5380 is an ideal part for designing an interface to connect to SCSI/PLUS. Its simplicity provides the flexiblity needed to support the defined protocol modifications, and its popularity with SCSI users guarantees plug compatibility with existing host adapters.

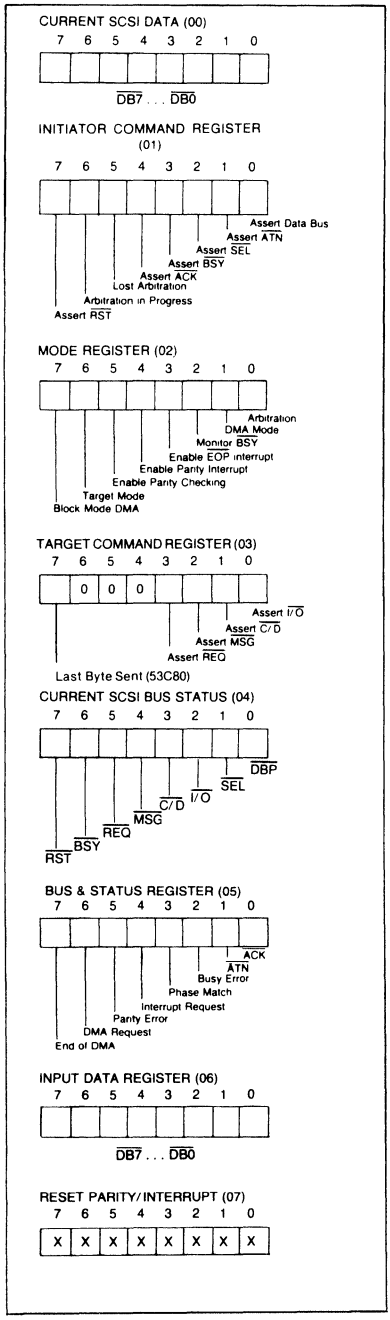
The NCR 5380 uses the Output Data Register to assert the proper device ID onto the SCSI bus during the Arbitration and Selection phases. Since the user is not restricted by the number of bits he is allowed to assert on the SCSI data bus, the Binary Arbitration and Binary Selection phases can be easily supported. In a Target role the Select Enable Register may be used to generate an interrupt if any bit in this register matches the binary address on the SCSI bus. Here again the NCR 5380 does not restrict this implementation.

The ability to support the master/slave operation requires independent control over the SCSI control signals by the bus slave devices and recognition by the bus master of the newly defined bus phase. The NCR 5380 provides independent signal control during Target operation and can be configured to generate an interrupt when a bus phase mismatch occurs if operating as an Initiator.

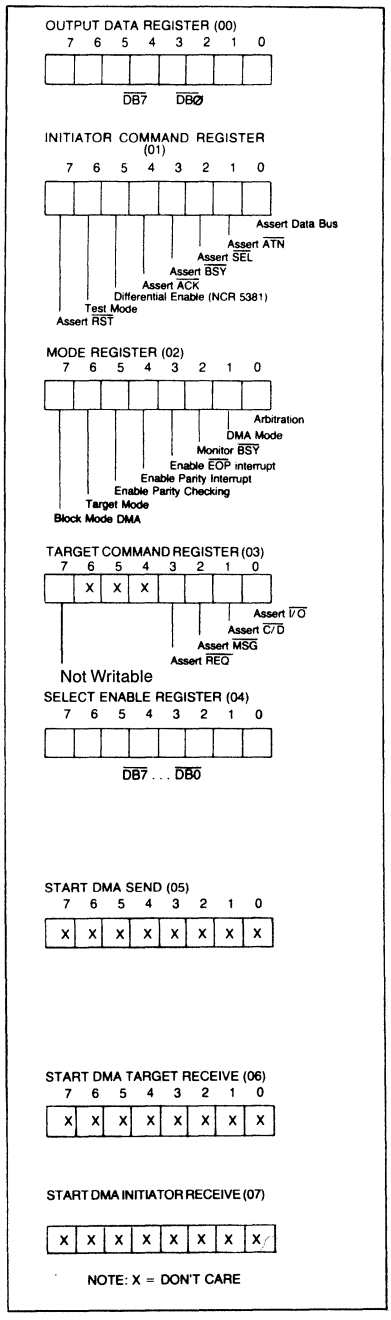
As in normal SCSI implementations, the use of on-chip bus transceivers significantly reduces parts count and provides for a highly reliable, cost effective SCSI/PLUS design. An additional advantage of on-chip MOS transceivers is the low leakage current. The NCR 5380 maximum leakage current of 50 μA meets the SCSI/PLUS bus load requirements. Up to 64 devices may occupy SCSI/PLUS bus positions if low-leakage integrated circuits such as the NCR 5380 are used.

A7. Register Reference Chart

READ



WRITE



53C80-40 SCSI INTERFACE

The NCR 5380 family of devices are designed to meet the SCSI protocol as defined by the ANSI standard. All the chips in the family can function as initiators or targets allowing them to be used in host adaptor and peripheral controller applications. These devices support arbitration as well as reselection. The 5380 family contains on-chip single-ended drivers which allow for direct connection to the SCSI Bus. These chips are controlled by reading and writing several internal registers which may be accessed by standard or memory-mapped I/O. Minimal processor intervention is required for DMA transfers because the chips control the SCSI handshaking signals.

There are five products that make up the NCR 5380 family: the 5380, 53C80-40, 53C80, 5381, and 53C81.

The 5380 is an NMOS SCSI protocol controller that contains several registers which are used to control the SCSI signals.

The 53C80-40 is a CMOS device designed as a higher performance replacement for the 5380. It has a faster SCSI handshake and uses less power than the 5380.

The 53C80 is a CMOS device that is functionally equivalent to the 5380 except that it has additional ground lines to maximize noise immunity which results in a larger package. Additionally, the 53C80 chip can be used as a core cell in ASIC designs. The design methodology and functionality has been proven in many semi-custom designs.

The 5381 is an NMOS device that is also functionally equivalent to the NMOS 5380 except that the 5381 includes signals which support the SCSI differential mode of operation.

The 53C81 is identical to the 5381 except that it is a CMOS device. All of these devices come in a variety of packages.

A product manual for the entire NCR 5380 family will be available the second quarter of 1988.

SCSI ASIC ABSTRACT

One of the main changes in the integrated circuit industry is the trend toward customization. NCR Microelectronics has recognized this trend and, as a result, implemented the NCR SCSI ASIC Design Program.

The NCR SCSI ASIC Design Program can be used by the customer to incorporate SCSI into a customized logic design while conforming to the ANSI standard. Cost reduction, enhanced performance, reduced board space, product security and a unique product offering are some of the advantages of customizing a SCSI design.

NCR Microelectronics offers the VS2000 Standard Cell Library of logic functions for use in designing semi-custom integrated circuits. The VS2000 has a 2-micron minimum feature size and is upwardly compatible with the existing 3-micron VS3000 Standard Cell Library and the 62A00 Gate Array Library.

The NCR 53C80 SCSI Supercell is a 2-micron standard cell that is functionally equivalent to the NCR 53C80 SCSI Interface Chip. It is designed to accommodate ASIC designs incorporating SCSI. The Supercell operates in both the initiator and target roles and can therefore be used in a variety of applications. Included in the cell are special high-current open-collector output drivers, capable of sinking 48mA at 0.5V, that allow for direct connection to the SCSI Bus.

The Supercell provides the following benefits: isolation of the SCSI function, high density, no dedicated test pins, clean customer interface, minimal requirements on external ASIC logic, consistent performance, and a proven production test program that assures a high quality level.

At present, a simulation model of the Supercell is available as a DABL™ model for Daisy® workstations. The Supercell is also available for Mentor Graphics® workstations.

Additionally, an optional oscillator cell has been developed which can be used in conjunction with a design using the Supercell.

5386 SCSI PROTOCOL CONTROLLER

General Description

The NCR SCSI Protocol Controller (SPC) is designed to accommodate the Small Computer Systems Interface (SCSI) as defined by the ANSI X3.131 SCSI standard. The SPC operates in both the Initiator and Target roles and can threfore be used in host adapter and control unit designs. This device supports arbitration, including reselection, and is intended to be used in systems that require either open collector or differential pair transceivers.

The NCR 5386 SCSI Protocol Controller communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory mapped I/O. A 24-bit Transfer Counter and the appropriate handshake signals accomodate large DMA transfers with minimal processor intervention. Since the NCR 5386 interrupts the MPU when it detects a bus condition that requires servicing, the MPU is freed from polling or controlling any of the SCSI bus signals.

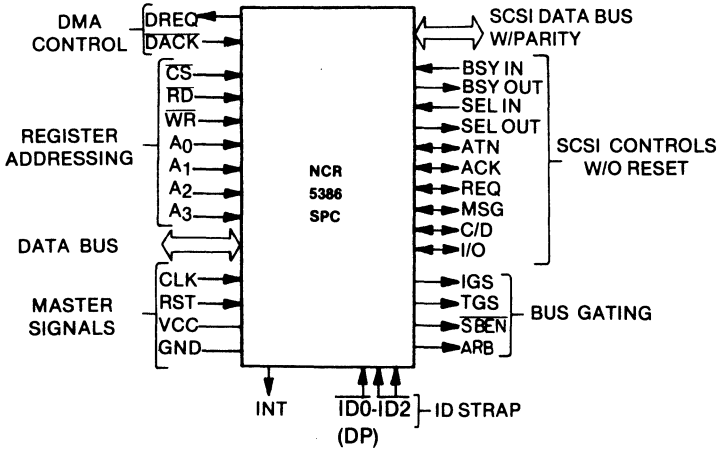
Below is a list of important features:

SCSI Interface

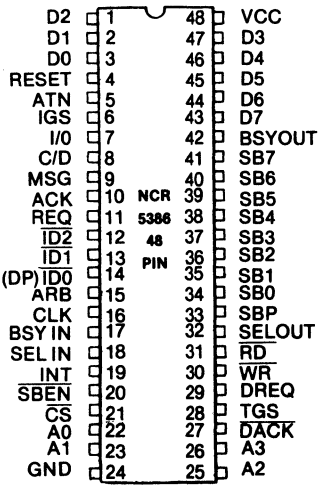
- Supports ANSI X3T9.2 SCSI Standard
- Asynchronous data transfers to 2.0 MBPS
- Supports both Initiator and Target roles
- Parity generation with optional checking
- Supports arbitration
- Controls all bus signals except Reset
- Doubly-buffered Data Register

MPU Interface

- Versatile MPU Bus Interface
- Memory or I/O mapped MPU interface
- DMA or programmed I/O transfers
- 24-bit Internal Transfer Counter
- Programmable (Re) Selection Timeouts
- Interrupts MPU on all bus conditions requiring service
- SCSI pass parity optional with checking

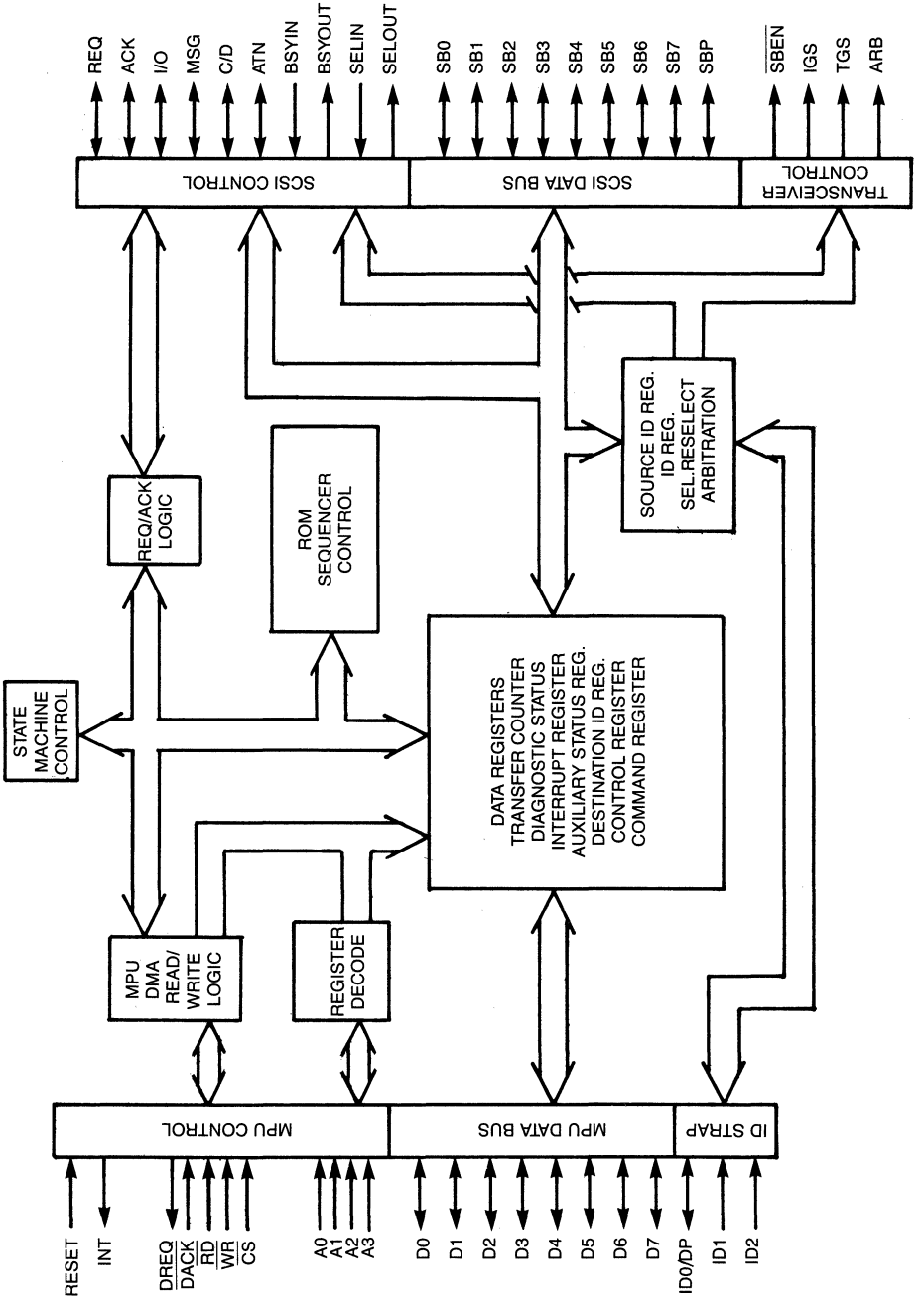


Functional Pin Grouping



Pinout

NCR 5386 Block Diagram



Pin Description

Microprocessor Interface Signals

CLK	16	Symmetrical square wave signal which generates internal chip timing. Maximum frequency is 10 MHz.
RESET	4	When high (1), this signal forces the chip into a reset state. All current operations are terminated. Internal storage elements are cleared and self-diagnostics are performed.
D0-D7	3-1 47-43	These signals comprise an active high data bus. It is intended that these signals be connected to the microprocessor data bus.
INT	19	This signal is used to interrupt the microprocessor for various bus conditions that require service. INT is set high for request and cleared when the chip is reset or the Interrupt Register is read.
$\overline{WR}$	30	Write pulse (active low) is used to strobe data from the data bus into an internal register which has been selected.
$\overline{RD}$	31	Read pulse (active low) is used to read data from an internal register that has been selected. The contents of the register are strobed onto the data bus.
$\overline{CS}$	21	When low (0), this signal enables reading from or writing to the internal register which has been selected.
A0-A3	22,23,25,26	These signals are used in conjunction with $\overline{CS}$, to address all the internal registers.
DREQ	29	Data request. When high (1), this signal indicates that the internal Data Register has a byte to transfer (inputting from the SCSI bus) or needs a byte to transfer (outputting to the SCSI bus). This signal becomes active only if the DMA mode bit in the Command Register is on. It is cleared when $\overline{DACK}$ becomes active.
$\overline{DACK}$	27	Data acknowledge. When low (0), this signal resets DREQ and selects the Data Register for input or output. $\overline{DACK}$ acts as a chip select for the Data Register when in the DMA mode. $\overline{DACK}$ and $\overline{CS}$ must never be active at the same time.

SCSI Interface Signals

$\overline{ID0}$-$\overline{ID2}$ (DP)	14-12	These active low signals determine the three-bit code of the SCSI bus ID assigned to the chip. External pullup resistors are required only if tied to switches or straps. (Optionally, pin 14 may be used as the data bus parity signal. In this case, the device ID is programmed by the system processor and pins 12 and 13 are not used. Refer to Section 4.6, "ID Register," for a detailed description.)

SB0-SB7, SBP	34-41, 33	Active high data bus. These signals comprise the SCSI data bus and are intended to be connected to the external SCSI bus transceivers.
BSYIN	17	When high (1), this signal indicates to the chip that the SCSI BSY signal is active.
BSYOUT	42	When high (1), the chip is asserting the BSY signal to the SCSI bus.
SELIN	18	When high (1), this signal indicates to the chip that the SCSI SEL signal is active.
SELOUT	32	When high (1), the chip is asserting the SEL signal to the SCSI bus.
ATN	5	INITIATOR ROLE: The chip asserts this signal when the microprocessor requests the attention condition or a parity error has been detected in a byte received from the SCSI bus. TARGET ROLE: This signal is an input which indicates the state of the ATN signal on the SCSI bus.
ACK	10	INITIATOR ROLE: The chip asserts this signal in response to REQ for a byte transfer on the SCSI bus. TARGET ROLE: This signal is an input which, when active, indicates a response to the REQ signal.
REQ	11	INITIATOR ROLE: This signal is an input which, when active, indicates that the Target is requesting a byte transfer on the SCSI bus. TARGET ROLE: Asserted by the chip to request a byte transfer on the SCSI bus.
MSG, C/D, I/O	9, 8, 7	INITATOR ROLE: These signals are inputs which indicate the current SCSI bus phase. TARGET ROLE: The chip drives these signals to indicate the current bus phase.
IGS	6	Initiator Group Select. When high (1), this signal indicates to the external SCSI drivers that the chip is controlling in the Initiator role. Its purpose is to enable the external drivers for ATN and ACK. When low (0), ATN and ACK should be received.
TGS	28	Target Group Select. When high (1), this signal indicates to the external SCSI drivers that the chip is controlling in the Target role. Its purpose is to enable the external drivers for REQ, MSG, C/D, and I/O. When low (0), REQ, MSG, C/D, and I/O should be received.
$\overline{\text{SBEN}}$	20	SCSI data Bus Enable. When low (0), this signal directly enables the external SCSI data bus drivers.
ARB	15	Arbitration phase. When high (1), this signal enables the external circuitry to place the ID bit on the SCSI bus for the Arbitration phase.

Power Signals

VCC	48	+5V input
GND	24	Signal reference input

OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	MAX	UNITS
Supply Voltage	V_{DD}	4.75	5.25	VDC
Supply Current	I_{DD}		300	mA
Ambient Temperature	T_A	0	70	°C

INPUT SIGNAL REQUIREMENTS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
High-level Input, V_{IH}		2.0	5.25	VDC
Low-level Input, V_{IL}		-0.3	0.8	VDC
High-level Input Current, I_{IH}	$V_{IH} = 5.25V$		10	μA
Low-level Input Current, I_{IL}	$V_{IL} = 0V$		-10	μA

OUTPUT SIGNAL REQUIREMENTS (Except $\overline{SBEN}$, IGS, and TGS)

PARAMETER	CONDITIONS	MIN	MAX	
High-level Output Voltage, V_{OH}	$V_{DD} = 4.75V$ @ $I_{OH} = -400\mu A$	2.4	—	VDC
Low-level Output Voltage, V_{OL}	$V_{DD} = 4.75V$ @ $I_{OL} = 2.0mA$	—	0.4	VDC

$\overline{SBEN}$, IGS, and TGS SIGNALS

PARAMETER	CONDITIONS	MIN	MAX	UNITS
High-level Output Voltage, V_{OH}	$V_{DD} = 4.75V$ @ $I_{OH} = -400\mu A$	2.4	—	VDC
Low-level Output Voltage, V_{OL}	$V_{DD} = 4.75V$ @ $I_{OL} = 4.0mA$	—	0.4	VDC

Internal Registers

General

The NCR SCSI Protocol Controller has a set of internal registers which are used by the microprocessor to direct the operation of the SCSI bus. These registers are read (written) by activating $\overline{CS}$ with an address on A3-A0 and then issuing a $\overline{RD}/(\overline{WR})$ pulse. They can be made to appear to a microprocessor as standard I/O ports or as memory-mapped I/O ports depending on the external circuitry that controls $\overline{CS}$. The following sections describe the operation of these internal registers.

Register Summary

A3	A2	A1	A0	R/W	Register Name
0	0	0	0	R/W	Data Register 1
0	0	0	1	R/W	Command Register
0	0	1	0	R/W	Control Register
0	0	1	1	R/W	Destination I/D
0	1	0	0	R	Auxillary Status
0	1	0	1	R/W	ID Register
0	1	1	0	R	Interrupt Register
0	1	1	1	R	Source ID
1	0	0	0	R	Data Register 2
1	0	0	1	R	Diagnostic Status
1	1	0	0	R/W	Transfer Counter (MSB)
1	1	0	1	R/W	Transfer Counter (2nd Byte)
1	1	1	0	R/W	Transfer Counter (LSB)
1	1	1	1	R/W	Reserved for Testability

Data Registers

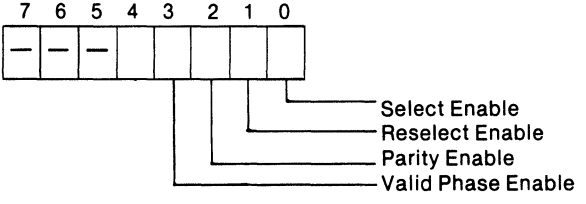
The Data Registers are used to transfer SCSI commands, data, status and message bytes between the microporcessor data bus and the SCSI bus. These are eight-bit registers which are doubly-buffered in order to support the maximum throughput. In the non-DMA mode, the microprocessor reads from (writes to) Data Register 1 by activating $\overline{CS}$ with A3-A0 = 0000 and issuing a $\overline{RD}/(\overline{WR})$ pulse. Two bits have been provided in the Auxiliary Status Register to indicate the status of both Data Register 1 and Data Register 2. In the DMA mode, the DMA logic reads from (writes to) Data Register 1 by responding to DREQ with $\overline{DACK}$ and issuing a $\overline{RD}/(\overline{WR})$ pulse. The SCSI bus reads from or writes to Data Register 2 when the chip is connected as an Initiator or Target and the bus is in one of the Information Transfer Phases.

Command Register

The Command Register is an eight-bit register used to give commands to the SCSI chip. The microprocessor and write to (read from) the Command Register by activating $\overline{CS}$ with A3-A0 = 0001 and issuing a $\overline{WR}/(\overline{RD})$ pulse. Writing to the Command Register causes the chip to execute the command that is written. The Command Register can be read; however, the chip resets the Command Register when the SPC sets an Interrupt. Therefore, one cannot guarantee that the data in the register will be correct after loading an interrupting command or enabling selection or reselection. To be safe, a copy of the last command issued whould be stored in the microprocessor's memory. Immediate commands are not stored.

Control Register

The eight-bit read/write register is used for enabling certain modes of operation for the SCSI Protocol Controller. The microprocessor reads from (writes to) the Control Register by activating $\overline{CS}$ with A3-A0 = 0010 and issuing a $\overline{RD}/(\overline{WR})$ pulse.

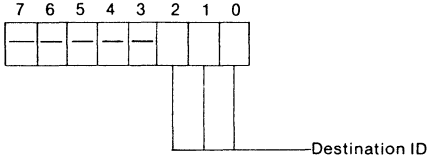


BIT 7-4	RESERVED	
BIT 3	VALID PHASE ENABLE	When this bit is a "1", the chip generates an interrupt for phase changes only when REQ becomes active. When bit 3 is a "0", the chip generates interrupts for phase changes that occur even though REQ may not be valid.
BIT 2	PARITY ENABLE	When the parity enable bit is a "1", the chip generates and checks parity on all transfers on the SCSI but. When the parity enable bit is a "0", the chip generates but does not check parity on bus transfers.
BIT 1	RESELECT ENABLE	When this bit is a "1", the chip will respond to any attempt by a Target to reselect it. When the bit is a "0", the chip will ignore all attempts to reselect it.
BIT 0	Select Enable	When this bit is a "1", the chip will respond to any attempts to select it as a Target. When it is a "0", the chip will ignore all selections.

NOTE: After being reset and completing self-diagnositcs, the control register will contain all zeros.

Destination ID Register

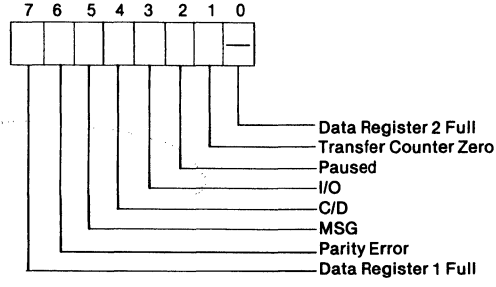
The Destination ID Register is an eight-bit register that is used to program the SCSI bus address of the destination device prior to issuing a Select or Reselect command to the chip. Bits 0-2 specify the address and bits 3-7 are always zeroes. The ID register is written (read) by activating $\overline{CS}$ with A3-A0 equal to "0011" and then pulsing $\overline{WR}(\overline{RD})$.



Auxiliary Status Register

The Auxiliary Status Register is an eight-bit read-only register. It contains bits which indicate the status of the chip's operational condition. Some of these bits are used to determine the reason for interrupts. Therefore, the Auxiliary Status Register should always be read prior to reading the Interrupt Register when servicing interrupts. After the Interrupt Register is read, the Auxiliary Status Register bits needed to service the interrupt may change.

The Auxiliary Status Register is read by activating $\overline{CS}$ with A3-A0 = 0100 and then pulsing $\overline{RD}$. The individual bits of the Auxiliary Status Register are defined below.



BIT 7 Data Register 1 Full

This bit indicates the status of Data Register 1 and must be monitored by the microprocessor during non-DMA mode commands that use Data Register 1. When the DMA mode bit in the Command Register is off (0) and the command being executed is one of Send, Receive or Transfer Info command (refer to Page 141, COMMANDS), data is transferred to (from) the chip by writing (reading) Data Register 1. Data Register 1 Full is set on (1) when data is written and turned off (0) when data is read. Therefore, Data Register 1 Full should be on before taking data from the chip, and off when sending data to the chip.

The Data Register 1 Full bits are always reset (to 0) at the time an interrupting type command is loaded into the Command Register. Therefore, when issuing such commands, the Command Register should be loaded prior to loading the Data Register 1 and monitoring the Data Register 1 Full flag.

BIT 6 Parity Error

When this bit is one, it indicates that the chip has detected a parity error on a byte of data received across the SCSI bus. It can be set when the chip is executing one of the Receive commands or the Transfer Info command (when the transfer is an input). This bit is reset after the Interrupt Register is read.

In Pass Parity mode, this bit will indicate whether MPU data had a parity error before being sent onto the SCSI bus. This parity error will not generate an interrupt to the MPU, but the receiving SCSI device will indicate an error was detected. The flag is useful in identifying where the error occurred.

BIT 3-5 I/O, C/D, MSG

These bits indicate the status of the SCSI I/O, C/D, and MSG signals at all times. They define the Information Phase type being requested by the Target. These signals are significant when servicing interrupts and the chip is logically connected to the bus in the Initiator role. An interrupt will occur with any phase change. This allows the Initiator to prepare for the next phase of data transfer. These bits are only held while INT is active. The bits are coded as follows:

I/O	C/D	MSG	Bus Phase
0	0	0	Data Out
0	0	1	Unspecified Info Out
0	1	0	Command
0	1	1	Message Out
1	0	0	Data In
1	0	1	Unspecified Info In
1	1	0	Status
1	1	1	Message In

BIT 2 Paused

When on (1), this bit indicates that the chip has aborted the command being executed in response to the Pause command. It is turned off when the interrupting type command code is loaded into the Command Register.

BIT 1 Transfer Counter Zero

This bit is provided to indicate the status of the 24-bit Transfer Counter. When on (1), it indicates that the Transfer counter is equal to zero. It is intended to facilitate interrupt servicing. This bit is also set when the Single-byte bit in the Command Register is active.

BIT 0 Data Register 2 Full

Since the 5386 design includes a doubly-buffered data register, this bit is used to indicate if data has been transferred into the second data register. If bit 0 is a "1", Data Register 2 is full; if this bit is a "0", Data Register 2 is empty.

NOTE: The Auxiliary Status Register will contain the following pattern after a Reset and self-diagnostics: 00xxx010.

ID Register

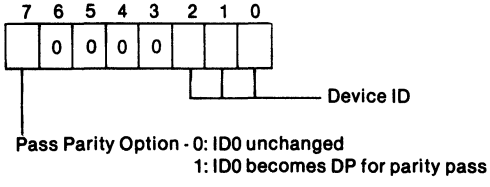
The ID Register operates in to configurations, the "strapped ID" mode or the "programmed ID" mode. If the ID register is written before the Control Register is written, the NCR 5386 assumes the "programmed ID" mode. In the "programmed ID" mode, pin 14 becomes the data bus parity signal (DP) and pins 12 and 13 are not used. If the Control Register is initialized and the ID register has not been written previously, then the device will assume the "strapped ID" mode which is identical to the NCR 5385E operation.

Strapped ID Configuration

The ID Register is an eight-bit read-only register in the "strapped ID" mode, which indicates the logical SCSI bus address occupied by the chip. Bits 0-2 directly reflect the logical inversion of the chip ID input signals ID0-ID2; the ID Register is active high whereas the ID input signals are active low. The ID Register allows the microprocessor to read the chip's SCSI bus address which would normally be strapped in hardware. Bits 3-6 of the ID Register will always be zeros. The ID Register is read by activating $\overline{CS}$ with A3-A0 = 0101 and then pulsing $\overline{RD}$.

Programmed ID Configuration

The ID Register is an eight-bit read/write register in the "programmed ID" mode, which is intended to be programmed by the system MPU. Bits 0-2 directly reflect the logical SCSI device ID. Bit 7 indicates if pass parity feature is enabled. In the "programmed ID" mode pin 14 becomes the parity signal for the data bus interface (DP) only if bit 7 is a "1", and pins 12 and 13 are not used.

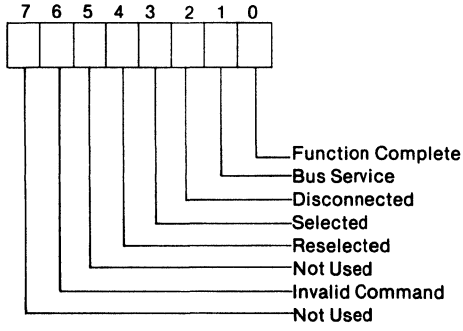


Interrupt Register

The Interrupt Register is an eight-bit read-only register. It is used in conjunction with the Auxiliary Status Register to determine the reason for an interrupt condition. This register is read by activating $\overline{CS}$ with A3-A0 = 0110 and then pulsing $\overline{RD}$. When the Interrupt Register is read, it automatically resets itself (after the read is complete) and enables the chip for a new interrupt condition. Since I/O, C/D, and MSG are only held while INT is active, the Auxiliary Status Register should always be read prior to reading the Interrupt Register.

If a Selected or Reselected interrupt occurs after issuing a command that would normally cause an interrupt, the chip will ignore the last command issued. This allows the microprocessor to service the Selected or Reselected interrupt prior to proceeding with the other operation. An example of this situation is when the microprocessor issues a command to select a Target at about the same time another Target reselects the chip. If the chip sees the reselection first, the microprocessor will receive an interrupt for the reselection, and the chip will ignore the Select command, which would now be invalid since the chip is now logically connected on the SCSI bus to another device.

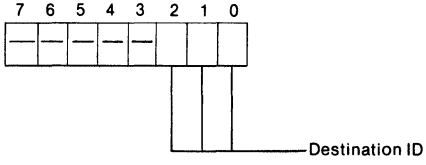
Individual interrupt conditions are described below. (Note: that for all cases, an interrupt condition is on, when the corresponding bit is a one (1), and off when zero (0).)



BIT 7	Not Used	May be either (1) or (0).
BIT 6	Invalid Command	When on (1), this bit indicates that the last command loaded into the command Register is not valid.
BIT 5	Not Used	(Reserved)
BIT 4	Reselected*	This interrupt will be on (1) when the chip has been reselected by another SCSI device. After setting this interrupt, the chip is logically connected to the bus in an Initiator role and is waiting for the Target to send REQ or disconnect from the bus.
BIT 3	Selected*	This interrupt will be on (1) whenever the chip has been selected by another SCSI device. After setting this interrupt, the chip is logically connected to the bus in the Target role and is waiting for a command to be loaded into the Command Register. * The chip will become selected (reselected) only if the ID data byte put on the SCSI bus during the Selection (Reselection) Phase has good parity and not more than one ID other than the chip's own ID is on.
BIT 2	Disconnected	This interrupt will be set on (1) when the chip is connected to the bus in the Initiator role and the Target disconnects or when the chip is executing a Select or Reselect command and the destination device does not respond before the Transfer Counter times out.
BIT 1	Bus Service	When the chip is logically connected to the bus in the Initiator role, this bit will be set on (1) whenever the Target sends a REQ which the chip cannot automatically handle. This happens when the first REQ for connection is received or when the chip is executing a Transfer Info or Transfer Pad command and either the Transfer Counter is zero or the Target changes the Information Phase type. If the valid phase enable bit is not set, Bus Service interrupt may be set if a phase change occurs before REQ appears. (See "valid phase enable," page 135. This early notification will allow the Initiator extra time to prepare for a phase change in some systems. (Note: the chip may Generate Bus Service Interrupts for phases that never request transfers. This is not an error condition, merely transitional staus of I/O, C/D, and MSG.) If the chip is logically connected in the Target role, this bit will be set on (1) whenever the Initiator asserts ATN. When indicating ATN the Bus Service interrupt may occur with a Selected interrupt or with a Function Complete interrupt.
BIT 0	Function Complete	When this bit is on (1), it indicates that the last interrupting command has completed. It is the normal successufl completion interrupt for Select, Reselect, Send and Receive commands. (Refer to page 141, COMMANDS) During any of the Receive commands, it is set on (1) along with the parity error bit as soon as a parity error is detected. A Bus Service interrupt may also occur simultaneously with the Function Complete if an ATN signal was activated during a Send or a Receive command. The Function Complete interrupt is also generated at the end of a Message In phase for a Transer Info command. (See TRANSFER INFO command for details.)

Source ID Register

The Source ID Register is an eight-bit read-only register which contains the three-bit encoded ID of the last device which selected or reselected the chip. The following in the format of the Source ID Register.



The ID Valid bit indicates that the source device placed its own ID bit on the SCSI bus during the Selection Phase. The SPC chip has encoded the source ID and placed it in bits 2-0. This information remains valid until the chip disconnects from the SCSI bus, at this time the ID Valid bit is reset.

Data Register 2

Data Register 2 is an eight-bit read-only register which may contain data that has been sent to the chip but not transferred across the bus. Bit 0 in the Auxiliary Status Register is used to determine if this register is full. The microprocessor can read Data Register 2 by activating $\overline{CS}$ with A3-A0 equal to 1000 and issuing a $\overline{RD}$ pulse.

Diagnostic Status Register

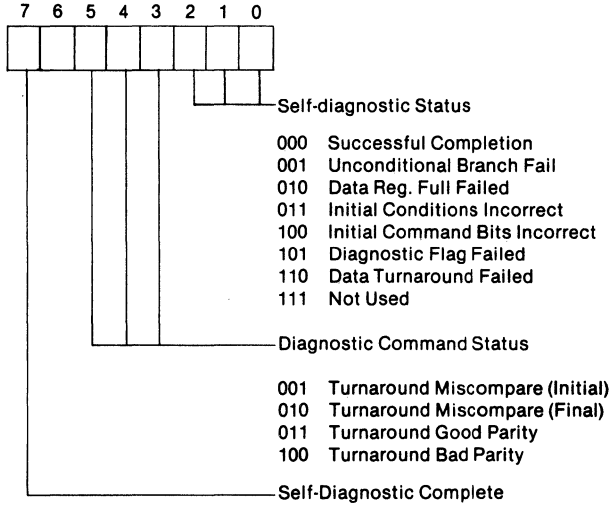
The Diagnostic Status Register is an eight-bit read-only register which indicates the result of self-diagnostics and the last diagnostic command issued to the chip. The format of the Diagnostic Status Register is shown below.

Bit 7 = 1 indicates that self-diagnostics have been completed. (NOTE: A reset will clear bits 6-3 if possible.) After a reset to the chip, the microprocessor should make sure that the Diagnostic Status Register contains the following pattern before attempting any commands: 10000000. This code indicates self-diagnostics are complete and no errors were detected. After a diagnostic command has been executed, bits 6-3 will contain the resulting status, but bit 7 and bits 2-0 are not affected.

The microprocessor may read the Diagnostic Status Register by activating $\overline{CS}$ with A3-A0 = 1001 and issuing a $\overline{RD}$ pulse.

If an error is detected during self-diagnostics, the proper status is loaded into the Diagnostic Status Register and the chip halts until a Reset command or a Reset signal is asserted. Refer to the Self-Diagnostic Status coe Summary for an explanation of the individual codes.

When a diagnostic command is issued to the chip, the chip will attempt to perform the function, load a status into bits 6-3, and initiate a Function Complete interrupt.



Self-Diagnostic Status Code Summary

000 - Successful Completion. The chip executed all self-diagnostics following a reset and detected no errors.

001 - Unconditional Branch Failed. The chip's internal sequencer attempted an unconditional branch and failed to reach the desired location.

010 - Data Register Full Failed. The chip attempted to set and reset the Data Register Full status bit in the Interrupt Register and failed.

011 - Initial Conditions Incorrect. The chip detected one of its internal initial conditions in the wrong state.

100 - Initial Command Bits Incorrect. The chip tested bits 6,4,2,1 and 0 of the Command Register and found at least one was not zero.

101 - Diagnostic Flag Failed. The chip failed in its attempt to set and reset its internal diagnostic flag.

110 - Data Turnaround Failed. During self-diagnostics the chip attempts to flush several bytes of data through its internal data paths. It also attempts to set and reset the Parity Error bit in the Interrupt Status Register. This status indicates that one of these operations failed.

Transfer Counter (Three Eight-Bit Counters)

The Transfer Counter is comprised of three, eight-bit register/counters. It is used by the chip for Send, Receive and Transfer commands that require more than a single byte of data to be transferred. It may also be used with Select and Reselect commands to set a timeout for no response. To write to (read from) the Transfer Counter, $\overline{CS}$ is activated with A3-A0 selecting a byte and then pulsing $\overline{WR}$ ($\overline{RD}$). The Transfer counter is addressed as shown below.

A3	A2	A1	A0	Selected Byte
1	1	0	0	Most Significant Byte
1	1	0	1	Middle Byte
1	1	1	0	Least Significant Byte

For Send, Receive and Transfer commands with single-byte not specified, the Transfer Counter specifies to the chip the maximum number of bytes to be sent or received before interrupting. The Transfer Counter must be loaded prior to issuing the command. When single-byte is specified, the chip neither uses nor alters the Transfer Counter. To facilitate servicing interrupts for commands that use the Transfer Counter, a bit is provided in the Auxiliary Status Register to indicate when the Transfer Counter is zero.

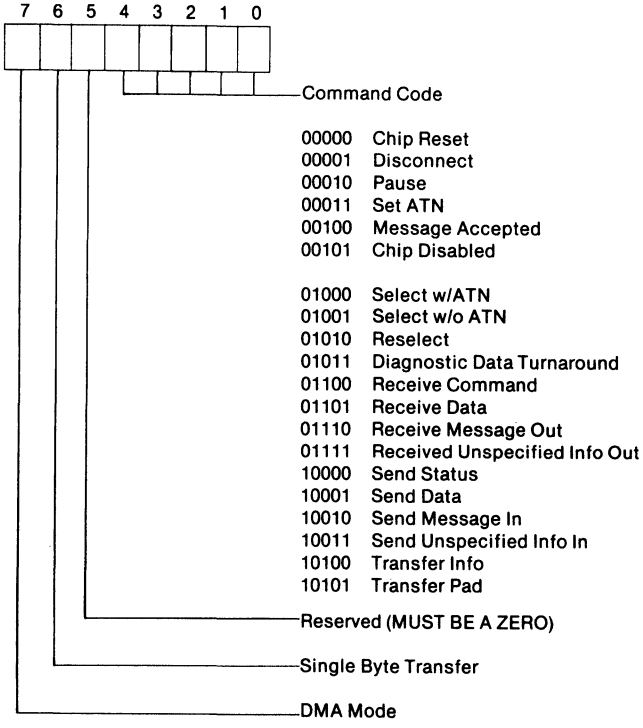
For Select and Reselect commands, the Transfer Counter specifies the number of time intervals (1024 CLK periods) that the chip will wait before automatically aborting the command due to no response (BSY) from the destination device. The Transfer Counter must be loaded prior to issuing the command. If the Transfer Counter is loaded with all zeroes, the timeout logic in the chip will be disabled, and the chip will not automatically abort the command due to no response.

Commands

This section defines command format, types, codes and operation. Commands are given to the chip by loading the Command Register.

Command Format

The bits in the Command Register are defined as follows.



BIT 7	DMA Mode	This bit is applicable only for commands that use the Data Register. When this bit is on (1), it indicates that data will be transferred to (from) the Data Register using the DMA signals DREQ and $\overline{\text{DACK}}$. When it is off (0), the microprocessor must monitor the state of the Data Register Full flag in the Auxiliary Status Register. Data is then transferred by using the appropriate input/output command.
BIT 6	Single Byte Transfer	When on (1), this bit indicates that only one byte of data is to be transferred for this command. The Transfer Counter will not be used or altered by the chip. Therefore, for common single byte message and status transfers, the Transfer Counter does not need to be loaded prior to issuing a command with this bit set. When this bit is off (0), the Transfer Counter is used by the chip to determine the length of the transfer for the command.
BIT 5	Reserved	This bit is not used and should always be programmed off (0).
BIT 4-0	Command Code	These bits are used to specify the command to be executed.

Command Types

There are two types of commands: Immediate and Interrupting. All of the Immediate commands, except for Pause, cause immediate results within three clock cycles from the time the Command Register is loaded. The Pause command is explained in a later section (See PAUSE). Interrupting commands do not result in immediate action. Their completion is always flagged by an interrupt.

Command codes 00000-00111 specify Immediate Commands. Immediate Commands that are listed as reserved, will be ignored if issued to the SPC chip. Command codes 01000-10101 specify Interrupting Commands. When one of these codes is loaded into the Command Regsiter, a second Interrupting command code should not be loaded until after the interrupt has occurred for the first command. However, an Immediate type command may be loaded before the interrupt for an Interrupting command occurs. If a reserved Interrupting command code is issued, the chip will respond with an Invalid Command interrupt.

Invalid Commands

The user of the chip can be in one of three states at any particular time: disconnected, connected as an Initiator, or connected as a Target. Commands are valid only in specified states. If an invalid Immediate command is issued, the chip will ignore the command. If an Interrupting command is issued in an invalid state, or a reserved Interrupting command code is issued, an Invalid Command interrupt will result. The exceptions are described in the next column.

The microprocessor must never issue any interrupting type command when the chip is not expecting such a command. Unpredictable results will occur in this case. The following is a list of user states in which the chip is not expecting an interrupting command:

1. The chip is currently processing an Interrupting type command and has not yet set the interrupt to signal the completion.

2. The chip is currently processing an Interrupting type command, a Pause command has been issued but the Paused bit in the Auxiliary Status Register has not been set.

3. The chip is connected as an Initiator, but the Target has not yet requested an Information Transfer.

4. The chip has completed a Transfer Info or Transfer Pad command and the Target has not requested additional information or has not changed the Information Phase.

In user states three and four, described above, the microprocessor must wait for a Bus Service, Disconnected, or Function Complete interrupt.

If an interrupting command is illegitimately issued in these states, no interrupt will occur for it, and it is likely that the current function will be altered.

Command Summary

Below is a summary that lists all commands. In the table the following abbreviations are used.

INT = INTERRUPTING D = DISCONNECTED I = CONNECTED AS AN INITIATOR
IMM = IMMEDIATE T = CONNECTED AS A TARGET

Command Code	Command	Type	Valid States
00000	Chip Reset	IMM	D,I,T
00001	Disconnect	IMM	I,T
00010	Paused	IMM	D,T
00011	Set ATN	IMM	I
00100	Message Accepted	IMM	I
00101	Chip Disable	IMM	D,I,T
00110-00111	Reserved	IMM	
01000	Select w/ATN	INT	D
01001	Select w/o ATN	INT	D
01010	Reselect	INT	D
01011	Diagnostic	INT	D
01100	Receive Command	INT	T
01101	Receive Data	INT	T
01110	Receive Message Out	INT	T
01111	Receive Unspecified Into Out	INT	T
10000	Send Status	INT	T
10001	Send Data	INT	T
10010	Send Message In	INT	T
10011	Send Unspecified Info In	INT	T
10100	Transfer Info	INT	I
10101	Transfer Pad	INT	I
10110-11111	Reserved	INT	

Command Definitions

Chip Reset

Chip Reset immediately stops any chip operation and resets all registers, counters, etc. on the chip. It performs the same operation as the hardware "reset" input.

Disconnect

Upon receipt of this command, the chip immediately releases all SCSI bus signals and returns to a Disconnected idle state. For the Initiator role, this is the normal method of disconnecting from the bus when a transfer is complete. For the Initiator role, Disconnect may be used to release the bus signals as a result of a timeout condition. In this case, the chip ignores the Target and is left in the Disconnected state. For the Disconnected state, it is not valid to issue a Disconnect command. If issued, the chip will ignore this command.

Pause

Pause is an Immediate command that is valid in the Disconnected state or when logically connected to the bus as a Target device. Pause is not valid when connected as an Initiator.

When connected as a Target, the Pause command provides a means of halting a Send or Receive command without having to wait for the transfer to complete. When Pause is issued, it immediately sets a flag in the chip. Within one byte transfer cycle, the chip recognizes the flag, aborts the Send or Recieve operation, and then sets the Paused status bit in the Auxiliary Status Register. At this time, the chip is still connected to the bus in the Target role, and it is waiting for another command.

The Pause command stops the Send or Receive command in an orderly manner leaving the Transfer Counter in a valid state that indicates the remaining number of bytes to be transferred. Also no REQ or ACK is asserted on the bus and no data is left in the chip waiting to be transferred. An operation that is paused may be resumed, if desired, simply by reloading the original command into the Command Register. (Note: after issuing the Pause while executing Send or Receive, it is necessary to continue transferring data with the chip (due to double-buffering) until the Paused status bit is set or an interrupt occurs.)

When in the disconnected state, Pause may be issued to abort a Select or Reselect command. After a Select or Reselect command is issued and before an interrupt occurs, a Pause command may be issued to abort the operation. The Pause command immediately sets an internal flag. If the chip has not yet won arbitration, it sets the Paused bit in the Auxiliary Status Register and waits in the disconnected state for another command. if the chip has won arbitration, it releases the bus by dropping the two ID bits with SELOUT on for a minimum of 100μs, checks for no BYSIN, and then releases the bus. After this procedure, it sets the Paused bit in the Auxiliary Status Register and waits for another command in the Disconnected state.

Since Pause is an Immediate command, it does not cause an interrupt. As previously noted, the chip sets the Paused status bit to indicate that it has been executed. If an interrupt-causing event occurs before the chip sees the pause flag set, the chip will set the interrupt. In this case, the Paused status bit is ot set by the chip either before or after the interrupt. In all cases, an interrupt-causing event will take precedence over Pause. For example, in the Target role if ATN is on when Pause is issued, a Bus Service interrupt will occur and the Paused status bit will not be set.

If the Pause command is issued when the chip is Disconnected, the Paused status bit will be set by the chip, provided it has not already detected a Selection or Reselection.

Set ATN

The Set ATN command causes ATN to be asserted immediately if the chip is connected as an Initiator. This command is invalid and ignored if issued when the chip is Disconnected or is operating in a Target role. The ATN signal is de-asserted in a Message Out phase when the transfer count becomes zero or one byte has been transferred (in a one-byte transfer command) during the execution of a Transfer Info command.

The chip automatically sets ATN in two cases:

1. If a Select w/ATN command is issued and arbitration is won.

2. If a parity error is detected on an input byte during execution of a Transfer Info command.

Message Accepted

The Message Accepted command is an Immediate command that is valid only when connected as an Initiator. It is used after a Transfer Info or Pad command (See pages 148, 149 Transer Info and Transfer Pad) to indicate to the chip that ACK can be de-asserted for the last byte.

When an Initiator receives a message, a Transfer command is used. If the transfer is an input (I/O = 1) and the information is a message (MSG = 1, C/D = 1), the chip interrupts after receiving the last byte with a Function Complete interrupt. For this one special case, the chip also leaves ACK asserted on the bus. By interrupting and leaving ACK asserted, the chip gives the microprocessor a chance to interpret the message and set ATN, prior to ACK being de-asserted. This allows the chip to properly request a Message Out phase if the Initiator wants to send a "Reject Message" to the Target.

Message Accepted must always be issued after a Transfer Info for a Message In phase, whether or not Set ATN is issued, in order to have the chip de-assert ACK. If the Initiator wants to reject the message, Set ATN would be issued first followed by Message Accepted. If the message is not to be rejected, only Message Accepted is issued. (Note: until Message Accepted is issued, the Target will not send another REQ since ACK is still asserted.)

Chip Disable

Chip Disable immediately stops all chip operations and logically disconnects it from the circuit. All outputs will be placed in a high impedance state and the chip will not respond to any commands (other than chip reset). The chip will also not respond to any activity on the SCSI bus. The only way to exit this condition is to activate the "reset" input or issue a Reset command.

Select w/ATN

This command causes the chip to attempt to select a Target. It may only be used if the SPC is in the Disconnected state. Any attempt to issue this command in another state will result in an Invalid Command interrupt. Before issuing this command, the microprocessor must load the Transfer Counter for a timeout on the Target's response. This value is computed according to the following formula:

Transfer Counter = Desired Timeout/(1024 × Clock Period)

If the Transfer Counter is loaded with the value zero, the chip will wait indefinitely for a response from the Target being selected.

The microprocessor must also load the Destination ID Register with the three-bit code of the Target to be selected before issuing the Select w/ATN command.

When the chip detects the Select w/ATN command, it begins by attempting to arbitrate for control of the SCSI bus. If, at any time during arbitration the chip becomes selected or reselected, the Select w/ATN is aborted and forgotten and the chip will interrupt with one of the following conditions:

1. Selected
2. Selected and Bus Service
3. Reselected

If arbitration is won, the chip places the SCSI bus in the Selection phase with ATN asserted, and uses the Destination ID Register to identify the desired Target. At the same time, the chip begins a timer based on the value computed above. If the Target does not respond within the timeout period, the chip will disconnect from the bus and interrupt with the Disconnected flag set in the Interrupt Register. (Note: The microprocessor should never monitor the Transfer Counter Zero flag in the Auxiliary Status Register to determine when a timeout has occurred.) If the Target responds within the allotted time, the chip will interrupt with a Function Complete status. Control of the SCSI bus then belongs to the selected Target and after the interrupt status has been read, another interrupt may occur indicating either that the Target has disconnected or is requesting a transfer.

If the timeout is disabled and the Target does not respond, or if arbitration is not won, the only way to abort the Select w/ATN command is to issue the Pause command. After the Pause command is issued, it is still possible that the Function Complete or Disconnect interrupts may occur. This happens if one of the interrupts get set before the chip detects the Pause command, or if the Target responds while the chip is sequencing off the SCSI bus in a timeout condition. If the chip does not set either interrupt, it will set the Paused bit in the Auxiliary Status Register. If the microprocessor detects this bit after issuing the Pause command, then it is assured taht the chip aborted the Select w/ATN command and no connection exists.

Select w/o ATN

The Select w/o ATN is identical to the Select w/ATN command except that the ATN signal is not asserted during the Selection phase.

Reselect

This command causes the chip to attempt to reselect an Initiator. It may only be used if the microprocessor is in the Disconnected state. Any attempt to issue this command in another state will result in an Invalid Command interrupt. Before issuing this command, the microprocessor must load the Transfer Counter for a timeout on the Initiator's response. This value is computed according to the following formula:

Transfer Counter = Desired Timeout/(1024 × Clock Period)

If the Transfer counter is loaded with the value zero, the chip will wait indefinitely for a response from the Initiator being reselected.

The microprocessor must also load the Destination ID Register with the three-bit code of the Initiator to be reselected before issuing the Reselect command.

When the chip detects the Reselect command, it begins by attempting to arbitrate for control of the SCSI bus. If, at any time during arbitration, the chip becomes selected or reselected, the Reselect is aborted and forgotten and the chip will interrupt with one of the following conditions:

1. Selected
2. Selected and Bus Service
3. Reselected

If arbitration is won, the chip places the SCSI bus in the Reselection phase using the Destination ID Register to identify the desired Initiator. At the same time, the chip begins a timer based on the value computed above. If the Initiator does not respond within the timeout period, the chip will disconnect from the bus and interrupt with the Disconnected flag set in the Interrupt Register. (Note: The microprocessor should never monitor the Transfer Counter Zero flag in the Auxiliary Status Register to determine when a timeout has occurred.) If the Initiator responds within the alloted time, the chip will interrupt with a Function Complete status. The chip (acting as the Target) is then in control of the SCSI bus, and waits for the Interrupt Register to be read by the microprocessor. After it has been read, the chip waits for a command from the microprocessor or ATN from the Initiator. If the ATN occurs, the chip will set the Bus Service interrupt. This interrupt may happen immediately after a command has been issued due to internal timing. In this case, the chip waits for the Interrupt Register to be read and the command is ignored. The chip then waits for a new command.

If the timeout is disabled and the Initiator does not respond, or if arbitration is not won, the only way to abort the Reselect command is to issue the Pause command. After the Pause command is issued, it is still possible that the Function Complete or Disconnected interrupts may occur. This happens if one of the interrupts get set before the chip detects the Pause command, or if the Initiator responds while the chip is sequencing off the SCSI bus in a timeout condition. If the chip does not set either interrupt, it will set the Paused bit in the Auxiliary Status Register. If the microprocessor detects this bit after issuing the Pause command, then it is assured that the chip aborted the Reselect command and no connection exists.

Diagnostic (Data Turnaround)

This Interrupting command causes the chip to attempt to turn a data byte around through its internal data paths. When the command is loaded into the Command Register, the Data Register Full bit is reset. The microprocesor then writes one byte into the Data Register. The chip moves the byte to another register and compares the contents of the Data Register. The byte is then moved to a third register (the SCSI output register) and good parity is generated if bit 6 of the command is off (0); bad parity is generated if bit 6 is on (1). Finally, the chip moves the byte back to the Data Register and compares it with the contents of the second register. Based on these comparisons and parity checking, the chip stores a result into the Diagnostic Status Register and sets the Function Complete interrupt. After reading the Interrupt Register, the microprocessor should make sure the Data Register Full bit is on (1) and read the contents of the Data Register. If the Data Register Full bit is not on (0), then an error has occurred. The following is a list of codes which are loaded into bits 6-3 of the Diagnostic Status Register as a result of this command.

BIT 6543	Result
0001	Data Miscompare (INITIAL)
0010	Data Miscompare (FINAL)
0011	Good Parity Detected
0100	Bad Parity Detected

Receive Commands

The Receive commands are Interrupting commands that are valid only when connected as a Target device. They are used by the Target to receive commands, data, and message information from an Initiator.

The Receive commands transfer data; therefore, the Single Byte Transfer and DMA mode bits in the Command Register are valid for these commands. If the Single Byte Transfer bit is off (0), the Transfer Counter must be loaded before a Receive command is issued to the chip. In this case, the chip uses the Transfer Counter to determine the number of bytes to receive.

When a Receive command is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. The chip then drives I/O, C/D, and MSG outputs for the proper information phase as follows.

Command Name	I/O	C/D	MSG
Receive Command	0	1	0
Receive Data	0	0	0
Receive Message Out	0	1	1
Receive Unspecified Info Out	0	0	1

The chip then proceeds to request and receive the specified number of information bytes. the DMA mode bit in the Command Register determines how the chip transfers these bytes from its Data Register to the microprocessor.

When a Receive command is terminated, the chip generates an interrupt. The following two events can cause termination:

1. The operation completes successfully; the Transfer Counter is zero. This event results in a Function Complete interrupt with the parity Error bit in the Auxiliary Status Register off (0). If the initiator activated ATN during the operation, the Bus Service bit will also be on.

2. A Parity Error occurs. The last byte transferred is the byte that caused the error. This event causes a Function Complete interrupt with the Parity Error bit in the Auxiliary Status Register on (1). If the Initiator activated ATN during the operation, the Bus Service bit will also be on.

After any of the interrupts, the chip is always left in the connected Target state. The Transfer Counter indicates the number of bytes remaining to be transferred (zero if completed successfully, and the Data Register is empty (the last byte received is sent to the microprocessor). Also, ACK and REQ are inactive on the bus.

(Note: if a Bus Service interrupt alone occurs after issuing a Receive command, the Initiator activated ATN before the chip began executing the command. In this case, the command is ignored by the chip.)

A Receive command may be stopped prior to an interrupt causing event by issuing a Pause command. Operation of the Pause command is explained in an earlier section. in the event the Initiator does not respond, or stops responding, the chip is left in a state where it cannot respond to a Pause command. For this case, a Disconnect command can be used to abort the command and the connection. The Disconnect command is explained in an earlier section.

Send Commands

The Send commands are Interrupting commands that are valid only when connected to the bus in the Target role. They are used by a Target to send status, data, and message information to an Initiator.

The Send commands transfer data, and therefore, the Single Byte Transfer and DMA mode bits in the Command Register are valid for these commands. If the Single Byte Transfer bit is off (0), the Transfer Counter must be loaded before a Send command is issued to the chip. In this case, the chip uses the Transfer Counter to determine the number of bytes to send.

When a Send command is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. Therefore, the first byte of data for the transfer cannot be put into the Data Register until after a Send command is loaded into the Command Register.

In executing a Send command, the chip drives the I/O, C/D, and MSG outputs for the proper information phase. These lines are logically driven for each Send command as shown below.

Command Name	I/O	C/D	MSG
Send Status	1	1	0
Send Data	1	0	0
Send Message In	1	1	1
Send Unspecified Info In	1	0	1

After resetting Data Register Full and driving I/O, C/D, and MSG, the chip then proceeds to monitor Data Register Full, take the data from the Data Register, and send it to the Initiator. The DMA mode bit in the Command Register specifies how the data is loaded into the chip.

After interrupting, the chip is left in the connected Target state, and ACK and REQ are inactive on the bus. When the transfer is complete, the chip interrupts with a Function Complete Interrupt. If the Initiator activated ATN during the transfer, a Bus Service bit will also be set by the chip.

(NOTE: if a Bus Service interrupt alone occures after issuing a Send command, the Initiator activated ATN before the chip began executing the command. In this case, the command is ignored by the chip.)

A Send command may be stopped prior to an interrupt causing event by issuing a Pause command. Operation of the Pause command is explained in an earlier section. In the event the Initiator does not, or stops responding, the chip is left in a state where it cannot respond to a Pause command. For this case, a Disconnect command can be used to abort the command and the connection. The Disconnect command is explained in a earlier section.

Transfer Info

The Transfer Info command is an Interrupting command that is valid only when connected to the bus in the Initiator role. It is used by the Initiator for all information transfers across the SCSI bus.

A Transfer Info command is issued by an Initiator in response to a Bus Service interrupt. The Bus Service interrupt, as explained in a previous section, is received by the connected Initiator upon the following conditions: receiving the first REQ from a Target, a previous command has completed and the Target changes phases, the Target changes phases before termination, or when a previous command has completed and the Target is requesting more information. It is not valid to issue a Transfer Info command without having a Bus Service interrupt, because the Target requests and controls all transfers. The chip will only permit one Transfer Info or Transfer Pad per Bus Service interrupt.

After an Initiator receives a Bus Service interrupt, and prior to issuing a Transfer Info command, the I/O, C/D, and MSG bits from the Auxiliary Status Register (read prior to reading the Interrupt) should be examined to determine the type of information phase and the direction of transfer requested by the Target. The Initiator then prepares for the transfer. If the Single Byte Transfer bit is not going to be set in the Command Register, the Transfer Counter must be loaded prior to issuing the Transfer Info command. This is done in order to specify to the chip the maximum number of bytes to be transferred.

When a Transfer Info is issued, the chip immediately resets the Data Register Full bit in the Auxiliary Status Register. For this reason, the first byte of data for an output operation cannot be loaded into the Data Register until after the command is loaded into the Command Register. The chip then proceeds with the transfer, expecting data to be read from (input), or written to (output), its Data Register as indicated by the DMA Mode bit in the Command Register. The chip automatically detects the direction of the transfer from the I/O bit which is stored in the Auxiliary Status Register.

The chip continues a transfer until an interrupt causing event occurs. The following four events will cause the chip to terminate and interrupt.

1. The maximum number of bytes specified have been transferred and the Target activated REQ or the Information Phase changed. This event results in a Bus Service Interrupt. Either single byte transfer was specified or the Transfer Counter is zero as indicated by a bit in the Auxiliary Status Register. The Target may or may not have changed the information phase type. The I/O, C/D, and MSG bits in the auxiliary Status Register need to be examined at the time of the interrupt to determine what phase the Target is requesting.

   (NOTE: Due to early notification of the phase change, a phase may be selected spuriously and not transfer any data. The microprocessor should not consider this an error condition. The early notification can be blocked with "valid phase" bit.)

2. The Target changes the information phase type before the maximum number of bytes are transferred. This event also causes a Bus Service interrupt. The new information phase may be determined by examining the I/O, C/D, and MSG bits in the Auxiliary Status Register. The Transfer Counter may be read at the time of the interrupt to determine the number of bytes remaining to be transferred. When this interrupt occurs for an output transfer, the chip may take one more byte from the microprocessor than it transfers, because of pre-fetching. However, the Transfer Counter still reflects the number of bytes remaining to be transferred.

3. The Target releases the bus by dropping BSY. This event results in a Disconnected interrupt. Following this interrupt, the chip is no longer in the Initiator role. It now remains in the Disconnected state.

4. The last byte of a Message Input phase has been received. This event results in a Function Complete interrupt. For this case, ACK is left active on the bus to allow the microprocessor to Set ATN for the purpose of rejecting the message. After this interrupt is received and a Set ATN is issued (if desired), a Message Accepted must be issued to turn off ACK for the last byte of the Message In phase.

For input transfers (I/O = 1), the chip checks parity for each byte received if the Parity Enable bit in the Control Register is on. When checking parity and the parity error occurs, the chip activates ATN prior to deactivating ACK for the byte that causes the error. It also turns on the Parity Error bit in the Auxiliary Status Register. The parity error, however, does not result in an interrupt. The chip waits for one of the four events listed above before interrupting. Therefore, the Parity Error bit should be examined when servicing any interrupt after issuing Transfer Info command for an input transfer.

If ATN is asserted by the chip, either because of a parity error or because a SET ATN command is issued, the ATN will remain asserted until the end of the connection, or until a Message Out is transferred. Therefore, during each cycle of Transer Info operation for output, the chip checks for a message phase (C/D = 1, MSG = 1) and also either a single byte transfer or the Transfer Counter set at zero. If these conditions exist, the chip turns off ATN prior to activating ACK for the last byte of the message.

As previously stated, a Transfer Info normally terminates with an interrupt. If a Transfer Info command must be aborted, possibly because of a timeout violation, either a Chip Reset or Disconnect command can be used. It is noted, however, that although these commands will force the chip into a disconnected state, the Target device is left on the bus. A SCSI bus reset, which is not a chip function, is the only way an Initiator can force a Target to disconnect.

Transfer Pad

The Transfer Pad command is an Interrupting command that is valid only when connected to the bus as an Initiator. It is similiar to the Transfer Info command except that the data transfer between the chip and the microprocessor but will be different.

Transfer Pad can be used by an Initiator to continue handshaking with a Target without giving data to, or taking data from, the chip. This may be useful if the Target requests an invalid Information Transfer Phase. The chip operates in the same manner as it does for a Transfer Info command, except that for output transfers it takes only one byte of data from the microprocessor and sends the same byte repeatedly until the transfer terminates. For input transfers, it accepts data from the SCSI bus but does not check parity or send it to the microprocessor. Though data is not exchanged with the microprocessor bus, the Transfer Counter is still used by the chip so that a maximum number of pad bytes can be specified.

Protocol for using a Transfer Pad command is the same as the Transfer Info except that the DMA Mode bit has significance only for output transfers. The Transfer Pad terminates because of the same four events that cause a Transfer Info command to terminate. Also, similar to the Transfer Info command, Chip Reset and Disconnect can be used to abort the command.

Bus Initiated Functions

Selection

If the Select Enable bit in the Control Register is on, the chip may be selected by another SCSI device to be a TARGET for an I/O operation. Selection occurs in the chip only if all the following conditions exist: SELOUT = 0, BSYIN = 0, SELIN = 1, I/O = 0, the chip's ID bit is asserted by the selecting device on the data bus, no more than one other ID bit (the Initiator's) is asserted on the data bus parity is good.

When all of these conditions exist, the chip is selected. It then encodes the Initiator's ID and loads it into bits 2-0 of the Source ID Register. The chip also detects whether or not the Initiator asserted its ID during selection, and either sets or resets the ID Valid bit in the Source ID Register.

The chip then asserts BSYOUT, waits for SELIN to turn off, and proceeds to take one of the following actions as a result of being selected:

1. If ATN is not asserted by the Initiator during selection, the chip generates a Selected interrupt indicating that the chip is connected as a Target.

2. If ATN is asserted, the chip simultaneously generates Selected, and Bus Service interrupts, indicating that the chip is connected as a Target and ATN is asserted.

Reselection

If the Reselect Enable bit in the Control Register is on, the chip may be reselected by a SCSI Target device. Reselection occurs only if SELOUT = 0, SELIN = 1, BYSIN =0, I/O = 1, the chip's ID bit and the Target's ID bit are asserted on the data bus, no other ID bits are asserted, and data bus parity is good.

When all of these conditions exist, the chip is reselected. It then encodes the Target's ID and loads it into the Source ID Register. The chip also sets the ID Valid bit in the Source ID Register.

The chip then asserts BSYOUT and waits for SELIN to be released by the Target. When the chip detects SELIN = 0, it de-asserts BSYOUT and then generates a Reselected interrupt.

Reselection is now complete and the chip is in the connected Initiator state.

Initialization

The SCSI device may be initialized by asserting RST for a period of at least 100ns, or by issuing a Chip Reset command to the device. The NCR 5386 will respond to the RST pulse or the Chip Reset command, by immediately disconnecting from the SCSI bus, initializing all storage elements and executing an internal self-diagnostic program. The self-diagnostic is explained in a previous section (See page 139, Diagnostic Status Register). The following table lists the status of all registers after the initialization procedure.

The controlling processor should loop on reading the Diagnostic Status Register until the Self-Diagnostic Complete bit (bit 7) is on (1). This should take approximately 350 clock cycles after reset occurs. If Programmed ID Mode is used, the processor should then check the remaining bits in this register for all zeros (no errors), and then load the ID Register. Following this, the Control Register should be programmed to enable the proper bits for SCSI operation. The SCSI Protocol Controller is now connected to the SCSI bus in a disconnected state. It is ready to receive commands from the controlling processor or respond to (re) selection attempts.

	7	6	5	4	3	2	1	0
Data Register 1	x	x	x	x	x	x	x	x
Command Register	0	0	0	0	0	0	0	0
Control Register	0	0	0	0	0	0	0	0
Destination ID Register	0	0	0	0	0	0	0	0
Auxiliary Status Register	0	0	x	x	x	0	1	0
ID Register	0	0	0	0	0	x	x	x
Interrupt Register	0	0	0	0	0	0	0	0
Source Register	0	0	0	0	0	1	1	1
Data Register 2	x	x	x	x	x	x	x	x
Diagnostic Status Register	1	x	x	x	x	x	x	x
Transfer Counter (MSB)	0	0	0	0	0	0	0	0
Transfer Coutner (2nd)	0	0	0	0	0	0	0	0
Transfer Counter (LSB)	0	0	0	0	0	0	0	0

x = Unknown

Register Initialization

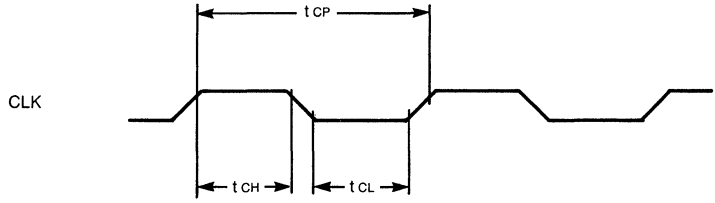
External Chip Timing

Timing requirements must be over the operating temperature (0-70 °C) and voltage (4.75 to 5.25V) ranges. Loading for all output signals, except $\overline{SBEN}$, is assumed to be four low-power Schottky inputs, including 50 pF capacitance. Loading for $\overline{SBEN}$ is assumed to be ten low-power Schottky inputs, including 100 pF capacitance.

Microprocessor Interface

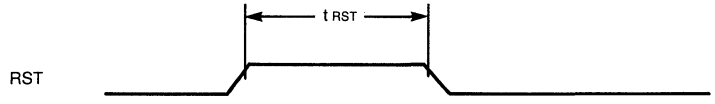
CLK

NAME	DESCRIPTION	MIN	MAX	UNITS
t_{CP}	Clock Period	100	200	ns
t_{CH}	Clock High	.45 t_{CP}	.55 t_{CP}	
t_{CL}	Clock Low	.45 t_{CP}	.55 t_{CP}	



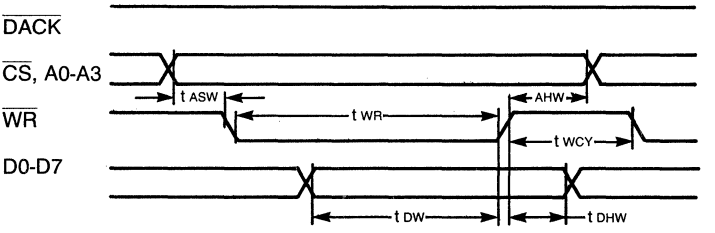
Reset

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{RST}	Reset Pulse Width	100			ns



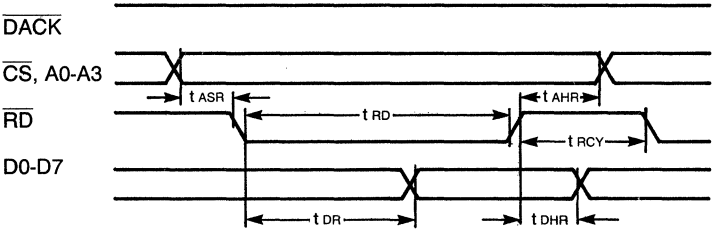
MPU Write

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{ASW}	Address Set-up Time	0			ns
t_{WR}	$\overline{WR}$ Pulse Width	95			ns
t_{DW}	Data-to $\overline{WR}$ High	50			ns
t_{AHW}	Address Hold Time	0			ns
t_{DHW}	Data Hold Time	20			ns
t_{WCY}	$\overline{WR}$ Off to $\overline{WR}$ or $\overline{RD}$ On	125			ns



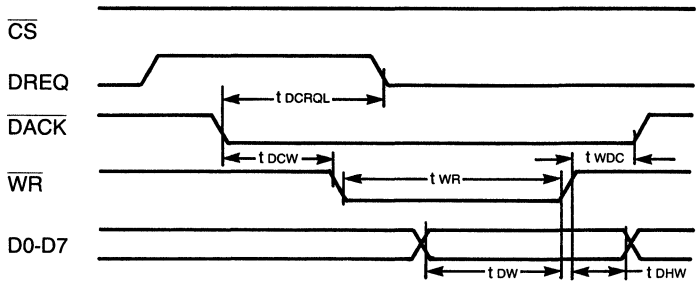
MPU Read

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{ASR}	Address Set-up Time to $\overline{RD}$	0			ns
t_{RD}	$\overline{RD}$ Pulse Width	125			ns
t_{DR}	$\overline{RD}$ to Data			90	ns
t_{AHR}	Address Hold Time	0			ns
t_{DHR}	Data Hold Time	10		65	ns
t_{RCY}	$\overline{RD}$ Off to $\overline{WR}$ or $\overline{RD}$ On	125			ns



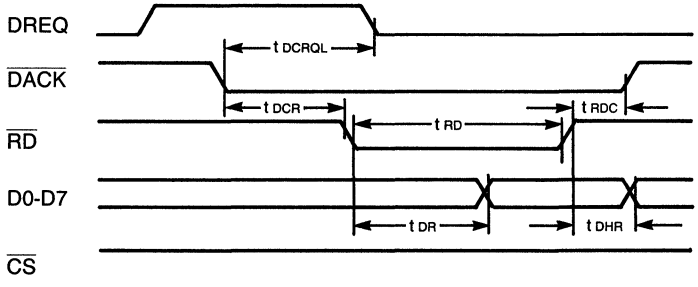
DMA Write

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DCRQL}	$\overline{DACK}$ to $\overline{DREQ}$ Low	0		40	ns
t_{DCW}	$\overline{DACK}$ to $\overline{WR}$	0			ns
t_{WR}	$\overline{WR}$ Pulse Width	70			ns
t_{WDC}	$\overline{WR}$ High to $\overline{DACK}$ High	0			ns
t_{DHW}	Data Hold Time	20			ns
t_{DW}	Data to $\overline{WR}$ High	50			ns



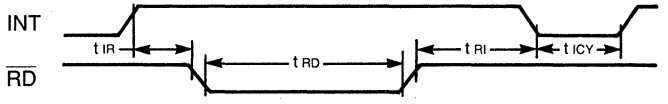
DMA Read

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DCRQL}	DACK to DREQ Low	0		40	ns
t_{DCR}	DACK to RD	0			ns
t_{RD}	RD Pulse Width	75			ns
t_{RDC}	RD High to DACK High	0			ns
t_{DHR}	Data Hold Time	10		65	ns
t_{DR}	RD to Data			75	ns



Interrupt

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{IR}	INT to $\overline{RD}$	0			ns
t_{RD}	$\overline{RD}$ Pulse Width	95			ns
t_{RI}	$\overline{RD}$ High to INT Low			125	ns
t_{ICY}	INT Off to INT On	125			ns



SCSI Interface

Selection (Initiator)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{BF}	Bus Free	385			ns
$t_{BIA(5)}$	BSYIN low to ARB high	1.2		2.6	us
t_{SLA}	SELOUT high to ARB low & ID bit Disabled	3.2			us
$t_{BIBO\ (5)}$	BSYIN low to BSYOUT high	1.2		2.8	us
t_{BCD}	Bus Clear Delay			225	ns
t_{AD}	Arbitration Delay	3.0			us
t_{PC}	Priority check to SELOUT	0			ns
$t_{BID\ (5)}$	BSYIN low to ID bit high	1.2		2.9	us
t_{ADV}	Arbitration Data Valid to Priority Check	0			ns
t_{SI}	SELOUT to IGS	2.0			us
t_{IDBL}	Target ID high to BSYOUT low	1.1			us
t_{BOBI}	BSYOUT low to BSYIN low	0		400	ns
t_{BSL}	BSYIN high to SELOUT low	800			ns
t_{DID}	SBEN active to Bus enabled	150			ns

NOTES:

1. The chip ensures that the bus remains free (BSYIN and SELIN inactive) for t_{BF} before attempting arbitration.

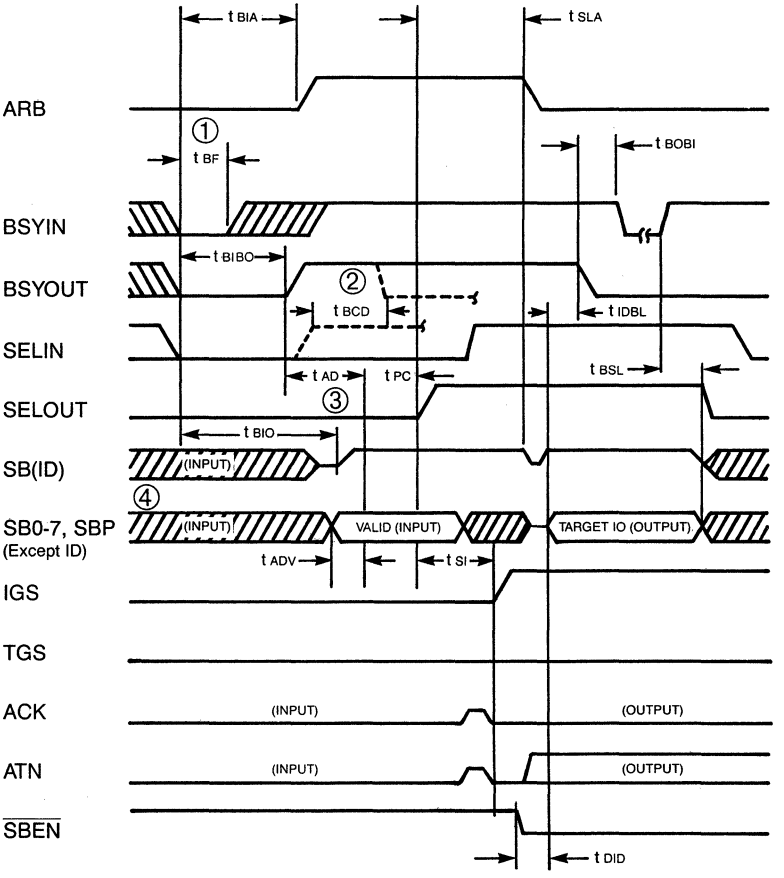
2. IF SELIN becomes active at any time during arbitration, the chip must de-assert BSYOUT within t_{BCD}.

3. The chip waits (t_{AD}), and then checks to see if arbitration is won (t_{PC}). The chip then asserts SELOUT if arbitration is won.

4. One of the data bits is assigned as an ID bit by the IDO-ID2 signals. During Bus Free, the chip places all of the data bits, including ID, in a high impedance state. During arbitration the chip enables its ID bit and drives it high, but the remainder of the data bits remain in the high impedance state for reading.

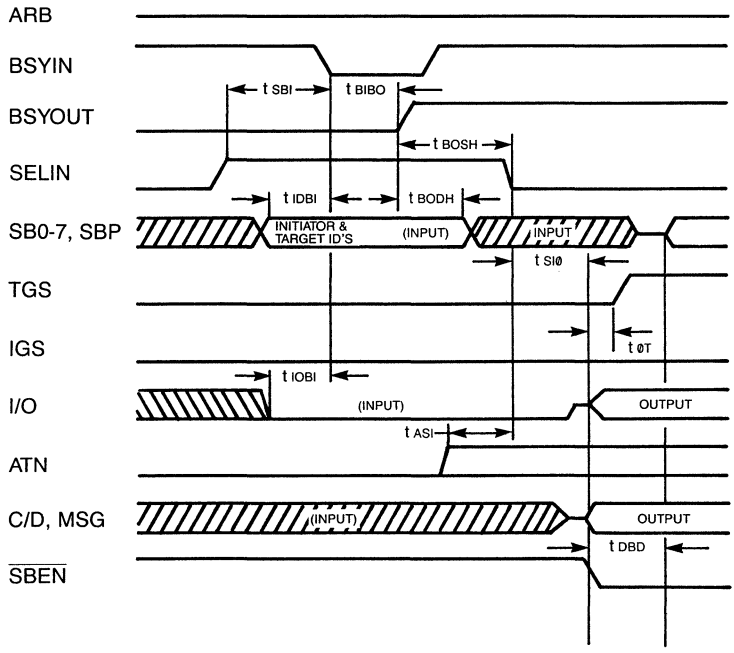
5. To verify these timings in a test environment, the user must allow a minimum of 45 clock cycles after the select command has been issued before the device begins to check for BSYIN low.

Selection (Initiator)



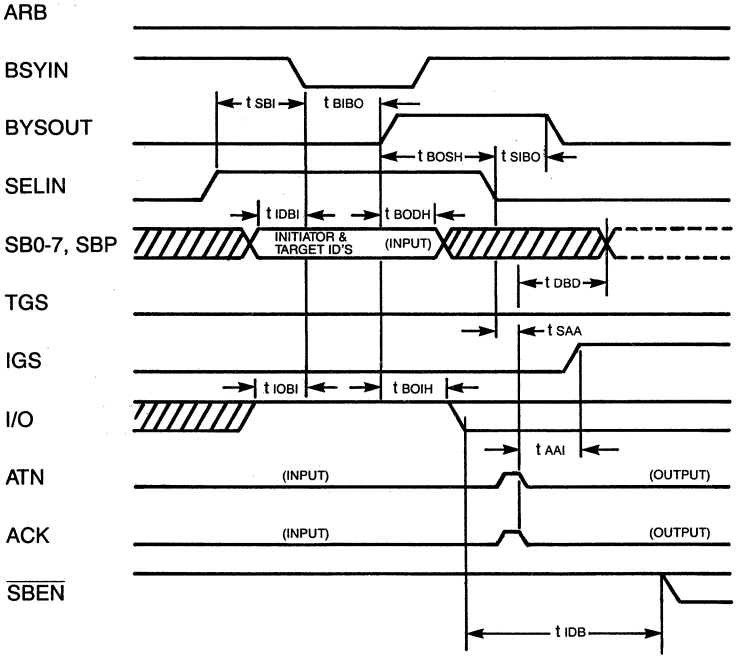
Selection (Target)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
T_{SBI}	SELIN high to BSYIN low	50			ns
t_{IDBI}	ID's valid to BSYIN low	0			ns
t_{IOBI}	I/O low to BSYIN low	0			ns
t_{BIBO}	BSYIN low to BSYOUT high	0		2.0	μs
t_{BODH}	BSYOUT high Data Hold	0			ns
t_{BOSH}	BSYOUT high SELIN Hold	0			ns
t_{ASI}	ATN high to SELIN low	0			ns
$t_{SI\emptyset}$	SELIN low to Phase signals Enabled	150			ns
t_{OT}	Phase signals enabled to TGS High	150			ns
t_{DBD}	SBEN low to Data Bus Enabled	150			ns



Reselection (Initiator)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{SBI}	SELIN high to BSYIN low	50			ns
t_{IDBI}	ID's valid to BSYIN low	0			ns
t_{IOBI}	I/O high to BSYIN low	0			ns
t_{BIBO}	BSYIN low to BSYOUT high	0		2.0	μs
t_{BODH}	BSYOUT high Data Hold	0			ns
t_{BOSH}	BSYOUT high SELIN Hold	0			ns
t_{BOIH}	BSYOUT high I/O hold	0			ns
t_{SIBO}	SELIN low to BSYOUT low	0			ns
t_{SAA}	SELIN low to ACK & ATN enabled	750			ns
t_{AAI}	ACK & ATN enabled to IGS high	150			ns
t_{IDB}	I/O low to $\overline{SBEN}$ low	0			ns
t_{DBD}	$\overline{SBEN}$ low to Data Bus Enabled	150			ns



Reselection (Target)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{BF}	Bus Free	385			ns
t_{BIA} (5)	BSYIN low to ARB high	1.2		2.6	us
t_{SLA}	SELOUT high to ARB low & ID bit Disabled	3.2			us
t_{BIBO} (5)	BSYIN low to BSYOUT high	1.2		2.8	us
t_{BCD}	Bus Clear Delay			225	ns
t_{AD}	Arbitration Delay	3.0			us
t_{PC}	Priority check to SELOUT	0			ns
t_{BID} (5)	BSYIN low to ID bit high	1.2		2.9	us
t_{ADV}	Arbitration Data Valid to Priority Check	0			ns
t_{S0}	SELOUT Phase signals Enabled & SBEN Low	2.4			us
t_{0T}	Phase Signals Enabled to TGS High	150			ns
t_{DID}	$\overline{SBEN}$ low to Bus Enabled	150			ns
t_{IDBL}	INITIATOR ID high to BSYOUT low	2.7			us
t_{BOBI}	BSYOUT low to BSYIN	0		400	ns
t_{BIBO}	BSYIN high to BSYOUT high	0.7		2.0	us
t_{BSL}	BSYOUT high to SELOUT low	450			ns

NOTES:

1. The chip ensures that the bus remains free (BSYIN and SELIN inactive) for t_{BF} before attempting arbitration.

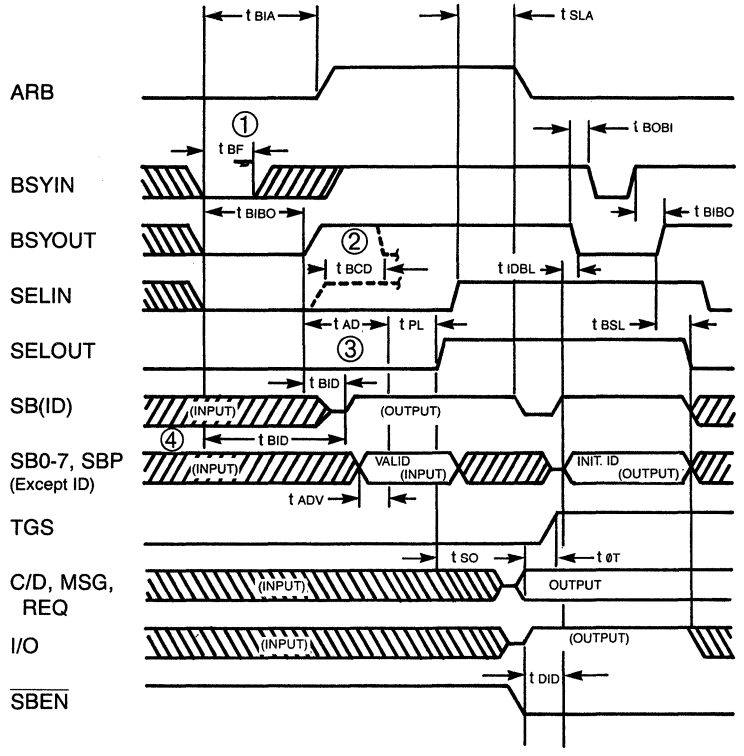
2. IF SELIN becomes active at any time during arbitration, the chip must de-assert BSYOUT within t_{BCD}.

3. The chip waits (t_{AD}), and then checks to see if arbitration is won (t_{PC}). The chip then asserts SELOUT if arbitration is won.

4. One of the data bits is assigned as an ID bit by the IDO-ID2 signals. During Bus Free, the chip places all of the data bits, including ID, in a high impedance state. During arbitration the chip enables its ID bit and drives it high, but the remainder of the data bits remain in the high impedance state for reading.

5. To verify these timings in a test environment, the user must allow a minimum of 45 clock cycles after the select command has been issued before the device begins to check for BSYIN low.

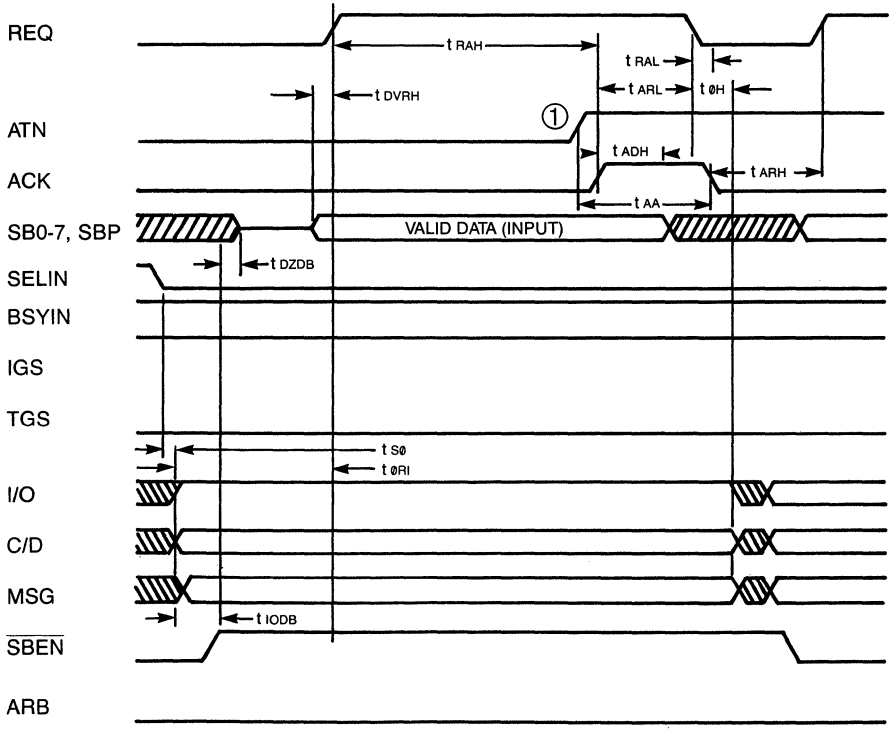
Reselection (Target)



Information Transfer Phase Input (Initiator)

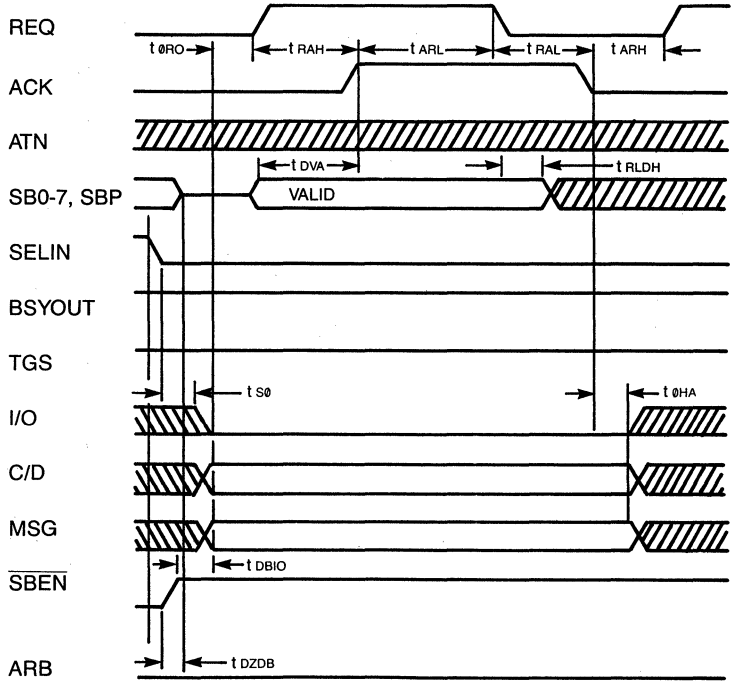
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{DVRH}	Data Valid to REQ high	0			ns
$t_{\emptyset RI}$	Phase Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	0			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{AA}	ATN high to ACK low	100			ns
$t_{S\emptyset}$	SELIN low to Phase change	0			ns
$t_{\emptyset H}$	Phase hold from ACK low	20			ns
t_{ADH}	Data hold from ACK high	0			ns
t_{ARL}	ACK high to REQ low	35			ns
t_{IODB}	I/O high to $\overline{SBEN}$ high			50	ns
t_{DZDB}	Data Bus disable from $\overline{SBEN}$ high			10	ns
t_{ARH}	ACK Low to REQ High	35			ns

NOTE 1: If the chip detects a parity error it must assert ATN at least t_{AA} before it de-asserts ACK.



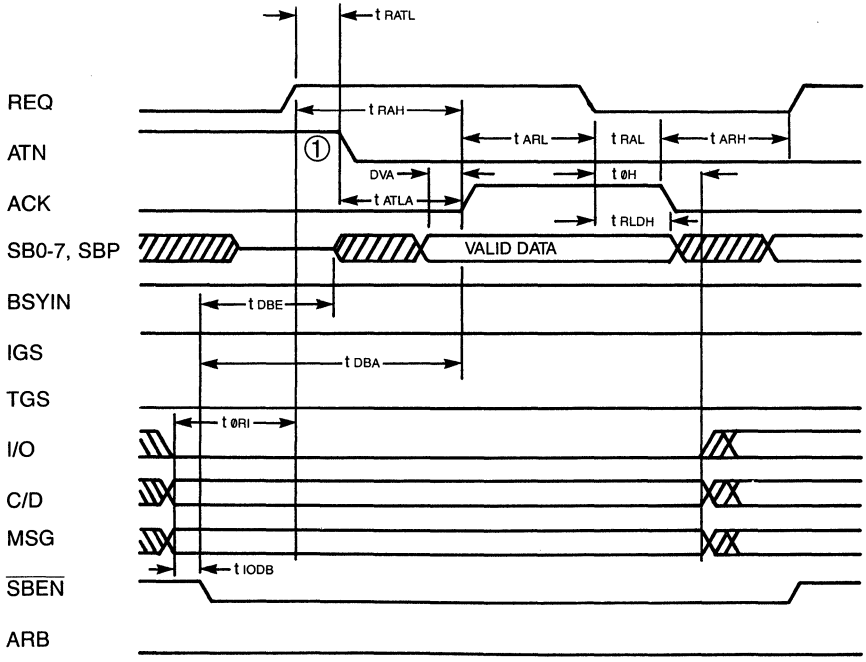
Information Transfer Phase Input (Target)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{S\emptyset}$	SELIN low to Phase Change	0			ns
$t_{\emptyset RO}$	Phase Change to REQ out	500			ns
t_{RAH}	REQ high to ACK high	35			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{DVA}	Data Valid to ACK high	0			ns
t_{RAL}	REQ low to ACK low	35			ns
t_{ARH}	ACK low to REQ high	0			ns
t_{RLDH}	REQ low Data Hold	0			ns
$t_{\emptyset HA}$	Phase Hold from ACK low	0			ns
t_{DBIO}	SBEN high to I/O low	0			ns
t_{DZDB}	Data Bus disable to SBEN high	0			ns



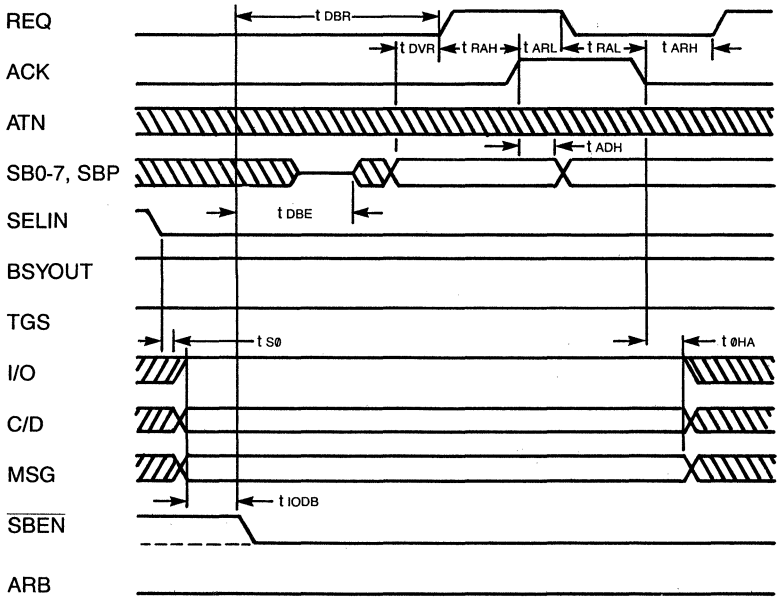
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{\phi RI}$	Phase Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	35			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{DVA}	Data Valid to ACK high	100			ns
t_{RLDH}	REQ low Data hold	0			ns
$t_{\phi H}$	Phase hold from ACK low	20			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{IODB}	I/O low to $\overline{SBEN}$ low	0			ns
t_{DBE}	$\overline{SBEN}$ low to Data Bus Enable	85			ns
t_{DBA}	$\overline{SBEN}$ low to ACK high	185			ns
t_{RATL}	REQ High to ATN low	0			ns
t_{ATLA}	ATN Low to ACK High	25			ns
t_{ARH}	ACK Low to REQ High	35			ns

NOTE 1: ATN is only de-asserted in this manner during the last byte of a Message Out Phase.



Information Transfer Phase Output (Target)

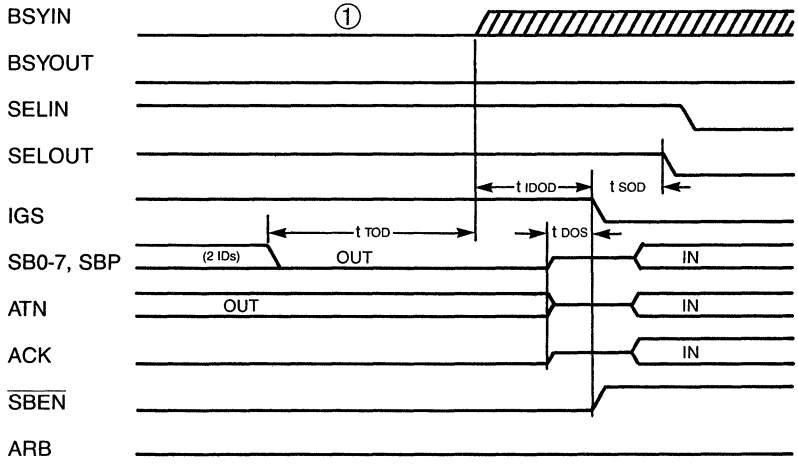
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
$t_{SØ}$	SELIN low to Phase Change	0			ns
t_{IODB}	I/O high to $\overline{SBEN}$ low	500			ns
t_{DBR}	$\overline{SBEN}$ low to REQ out	185			ns
t_{DVA}	Data Valid to REQ high	100			ns
t_{RAH}	REQ high to ACK high	0			ns
t_{ARL}	ACK high to REQ low	0			ns
t_{RAL}	REQ low to ACK low	0			ns
t_{ARH}	ACK low to REQ high	0			ns
$t_{ØHA}$	Phase hold from ACK low	0			ns
t_{ADH}	Data hold from ACK low	0			ns
t_{DBE}	$\overline{SBEN}$ low to Data Bus Enabled	85			ns



Bus Release From Selection (Initiator)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TOD}	Bus Release Timeout Delay	100			μs
t_{IDOD}	IGS & $\overline{SBEN}$ Turn-off Delay	0			ns
t_{SOD}	SELOUT Turn-off Delay	0			ns
t_{DOS}	Driver Turn-off set-up to IGS & $\overline{SBEN}$ off	0			ns

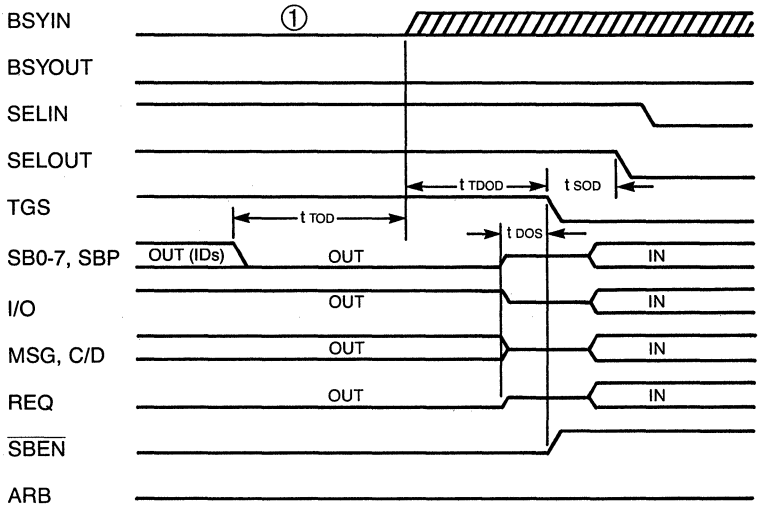
NOTE 1: If the chip detects BSYIN active by the end of the timeout delay, the bus release sequence shall be aborted since selection has been successful.



Bus Release From Reselection (Target)

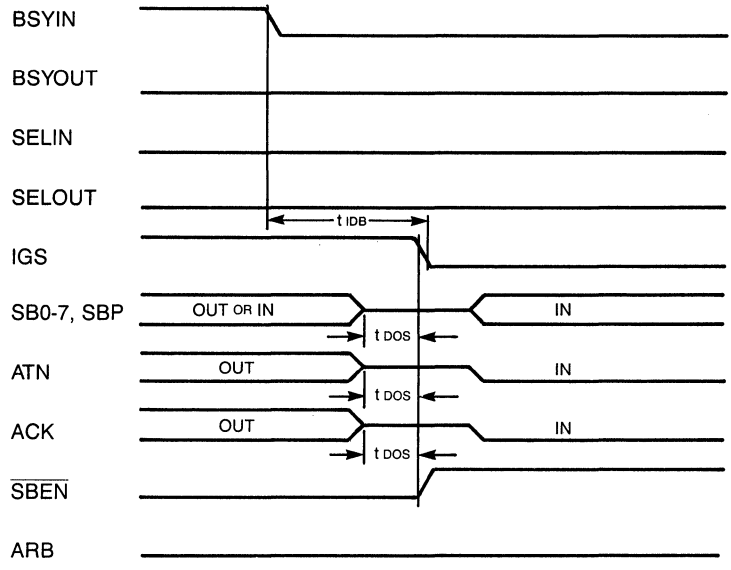
NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TOD}	Bus Release Timeout Delay	100			*us*
t_{DOD}	TGS & $\overline{SBEN}$ Turn-off Delay	0			ns
t_{SOD}	SELOUT Turn-off Delay	0			ns
t_{DOS}	Driver Turn-off set-up to TGS & $\overline{SBEN}$ off	0			ns

NOTE 1: If the chip detects BSYIN active by the end of the timeout delay, the bus release sequence shall be aborted since selection has been successful.



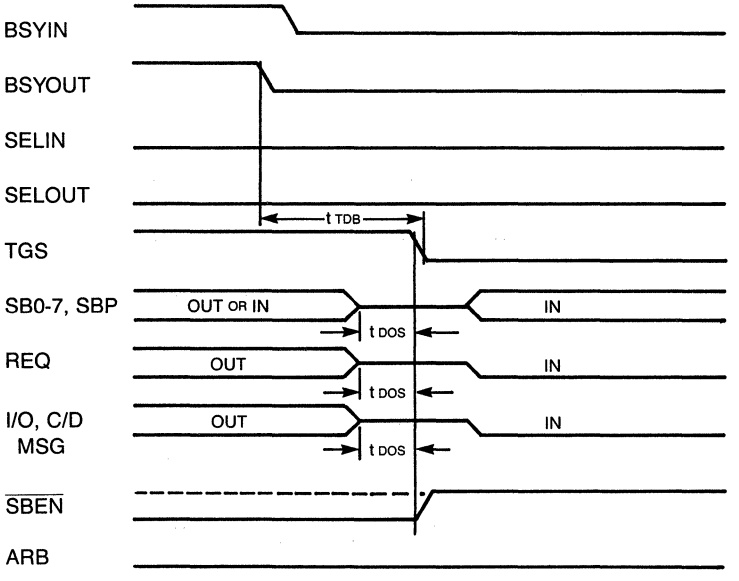
Bus Release From Information Phase (Initiator)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{IDB}	IGS & $\overline{SBEN}$ Turn-off Delay from BSYIN off			225	ns
t_{DOS}	Driver Turn-off set-up to IGS off	0			ns



Bus Release From Information Phase (Target)

NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
t_{TDB}	TGS & $\overline{SBEN}$ Turn-off from BSYOUT off			225	ns
t_{DOS}	Driver Turn-off set-up to TGS off	0			ns



8310/C10/C11 GENERAL PURPOSE BUS TRANSCEIVER

Introduction

Product Description

The NCR 8310 General Purpose Bus Transceiver Chip, an NMOS device available in either a 48 pin DIP or 52 pin PLCC, is designed as a 48 mA bus transceiver chip for all of the Small Computer System Interface (SCSI) bus signals. It contains high-current single-ended drivers for the SCSI bus. The NCR 8310 is specifically intended to be used with the NCR 5386 SCSI Protocol Controller Family and interfaces directly to it. Additionally, it may be used with other interfaces where a general purpose 48 mA bus transceiver is required.

The NCR 83C10 is functionally equivalent to the NCR 8310 but has been designed in CMOS technology. It is available in a 52 pin PLCC package and is pin compatible with the NMOS device in this package, with pins 33, 46 and 47 listed as No Connects (NC) on the NCR 8310 and listed as Vss pins on the NCR 83C10.

Additionally, the NCR 83C11 is available. Designed in CMOS, the NCR 83C11 is functionally equivalent to both the NCR 8310 and NCR 83C10, but offers a pinout that has been optimized for connection to the standard SCSI connector. This device is also available in a 52 pin PLCC package.

NOTE: The NCR 5386 family includes the NCR 5385E and NCR 5386.

Key Features

- Low Cost
- CMOS and NMOS technology
- Interfaces Directly to NCR 5386 Family
- Can Be Used With Other Interfaces Requiring 48 mA Drivers
- Standard 48 Pin DIP (NCR 8310 Only) or 52 Pin PLCC (8310, 83C10, 83C11)

Application Overview

Figures 1.1 and 1.2 respectively, illustrate the interconnection of the NCR 5386 family with the NCR 8310 Family, and the interconnection of any interface device to the NCR 8310 Family.

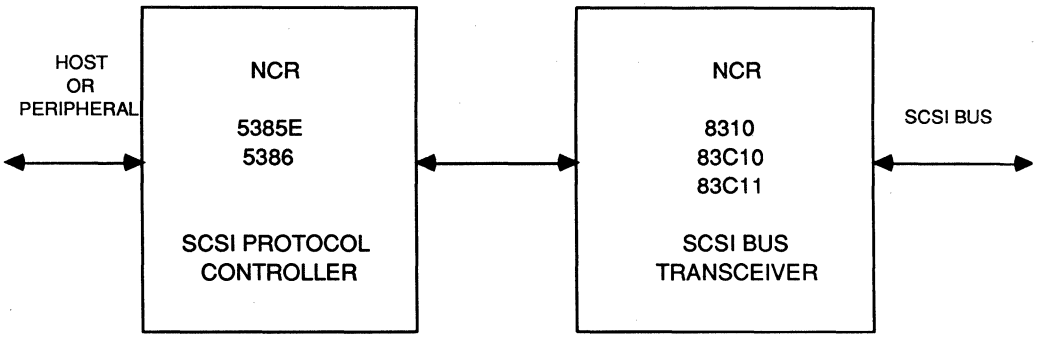


FIGURE 1.1 Interconnection of NCR 5386 Family to NCR 8310 Family

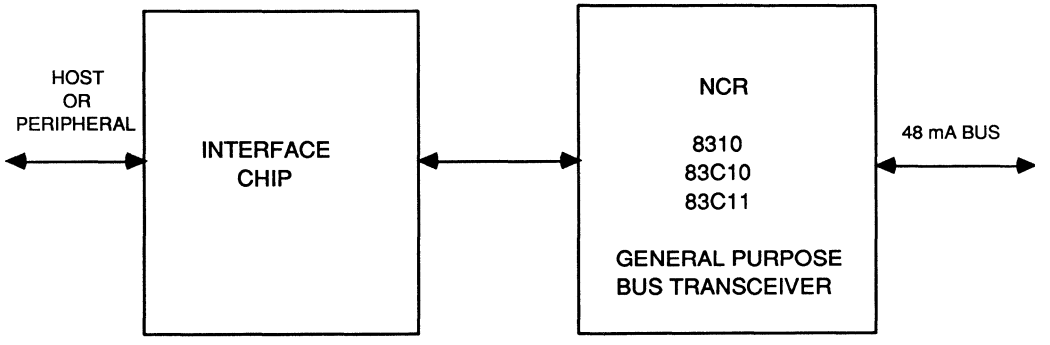


FIGURE 1.2 Interconnection of Any Interface Chip to NCR 8310 Family

Figure 1.3 shows the specific interface between the NCR 5386 Family and the NCR 8310 Family. With this implementation, a two-chip solution is possible.

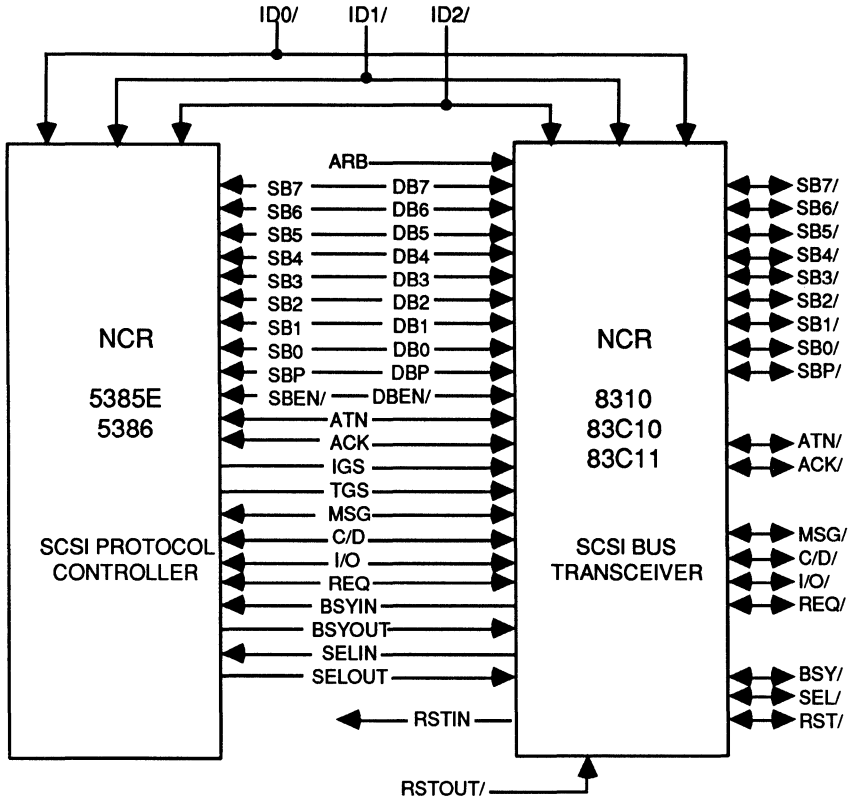


FIGURE 1.3 Single-Ended Interface Using the NCR 5386 SCSI Protocol Controller Family and the NCR 8310 General Purpose Bus Transceiver Family

Functional Pin Grouping

Figure 1.4 shows the functional pin grouping for the NCR 8310 Family of products.

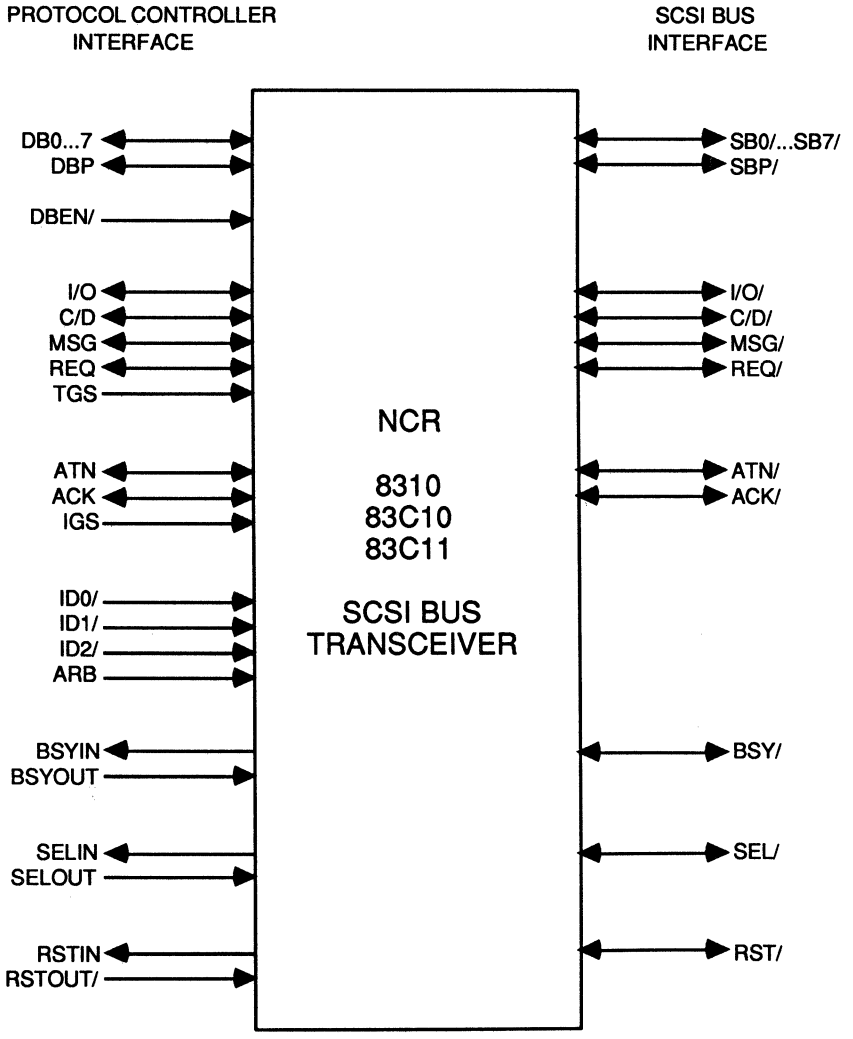


FIGURE 1.4 NCR 8310 Family Functional Pin Grouping

Pinout

Figures 1.5, 1.6 and 1.7 show the pinouts for the NCR 8310, NCR 83C10 and NCR 83C11 respectively. The pinout for the NCR 8310 in the 52 pin PLCC package is identical to Figure 1.6 with pins 33, 46 and 47 listed as No Connects (NC) for the NCR 8310. Pins 33, 46 and 47 of the NCR 83C10 MUST be tied to Vss.

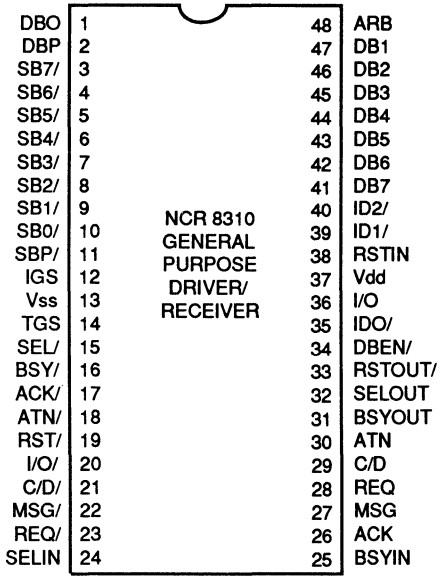


FIGURE 1.5 NCR 8310 - 48 Pin DIP

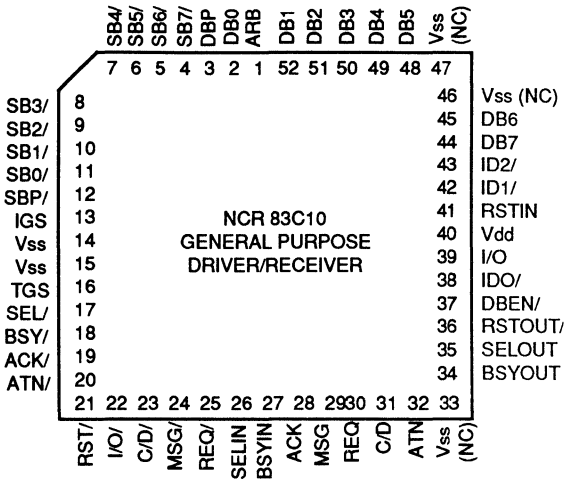


FIGURE 1.6 NCR 83C10 - 52 Pin PLCC

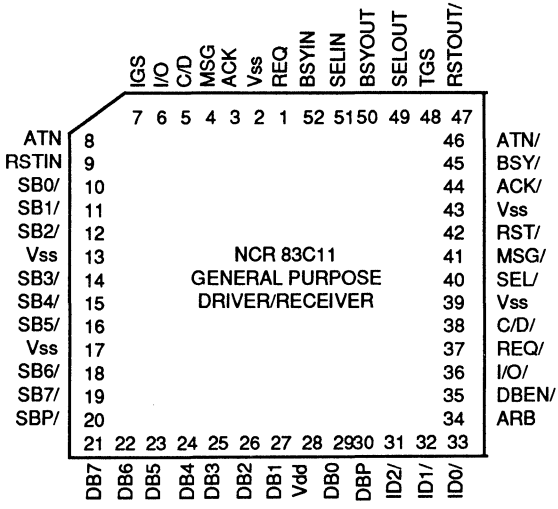


FIGURE 1.7 NCR 83C11 - 52 Pin PLCC

Pin Description

SCSI Interface Signals

The following signals are all bi-directional, active-low, open-drain signals. With 48 mA sink capacity, all pins interface directly to the SCSI bus.

SIGNAL	DESCRIPTION
SB0/...SB7/ SBP/	These eight data bits (SB0/-SB7/) plus a parity bit (SBP/) form the data bus. SB7/ is the most significant bit and has the highest priority during the Arbitration phase. Data Parity is odd. Parity is not valid during Arbitration.
I/O/	I/O/ is a signal driven by a target which controls the direction of data movement on the SCSI bus. True indicates input to the initiator. This signal is also used to distinguish between Selection and Reselection phases.
C/D/	A signal driven by the target, C/D/ indicates Command or Data information is on the data bus. This signal is received by the initiator.
MSG/	MSG/ is a signal driven by the target during the Message phase. This signal is received by the initiator.
REQ/	Driven by a target, REQ/ indicates a request for a REQ/ACK data transfer handshake. This signal is received by the initiator.
ACK/	Driven by an initiator, ACK/ indicates an acknowledgement for a REQ/ACK data transfer handshake. In the target role, ACK/ is received as a response to the REQ/ signal.
ATN/	Driven by an initiator, ATN/ indicates an attention condition to the target.
BSY/	This signal indicates that the SCSI bus is being used and can be driven by both the initiator and the target device.
SEL/	SEL/ is used by the initiator to select a target or by a target to reselect an initiator.
RST/	The RST/ signal indicates an SCSI bus RESET condition.
DB0...DB7	(I/O) This bidirectional data bus is connected to the NCR 5386 signal SB0/...SB7/.
DBP	(I/O) This bidirectional parity signal is connected to the NCR 5386 signal SBP/.
DBEN/	(Input) This signal enables the SCSI bus drivers for SB0/...SB7/ and SBP/. It is connected to the NCR 5386 signal SBEN/.
TGS	(Input) This signal enables the SCSI bus drivers for I/O/, C/D/, MSG/ and REQ/. It is connected to the NCR 5386 signal TGS.
IGS	(Input) This signal enables the SCSI bus drivers for ACK/ and ATN/. It is connected to the NCR 5386 signal IGS.
ARB	(Input) This signal enables decode of ID0/...ID2/ and asserts the selected SCSI data bus ID. It is connected to the NCR 5386 signal ARB.
ID0/...ID2/	(Input) These ID signals are used during arbitration to select the correct SB(n)/line. They are connected to the NCR 5386 signals ID0/...ID2/.
I/O	(I/O) INPUT/OUTPUT - This pin is used to drive or receive the SCSI signal I/O. It is connected to the NCR 5386 signal I/O.
C/D	(I/O) CONTROL/DATA - This pin is used to drive or receive the SCSI signal C/D/. It is connected to the NCR 5386 signal C/D/.
MSG	(I/O) MESSAGE - This pin is used to drive or receive the SCSI signal MSG/. It is connected to the NCR 5386 signal MSG.

SIGNAL	DESCRIPTION
REQ	(I/O) REQUEST - This pin is used to drive or receive the SCSI signal REQ/. It is connected to the NCR 5386 signal REQ.
ACK	(I/O) ACKNOWLEDGE - This pin controls the SCSI signal ACK/. It is connected to the NCR 5386 signal ACK.
ATN	(I/O) ATTENTION - This pin controls the SCSI signal ATN/. It is connected to the NCR 5386 signal ATN.
BSYIN	(Output) This signal indicates the received state of the SCSI signal BSY/. It is connected to the NCR 5386 signal BSYIN.
BSYOUT	(Input) This signal drives the SCSI signal BSY/. It is connected to the NCR 5386 signal BSYOUT.
SELIN	(Output) This signal indicates the received state of the SCSI signal SEL/. It is connected to the NCR 5386 signal SELIN.
SELOUT	(Input) This signal drives the SCSI signal SEL/. It is connected to the NCR 5386 signal SELOUT.
RSTIN	(Output) This signal indicates the received state of the SCSI signal RST/. It is not connected to the NCR 5386.
RSTOUT/	(Input) This signal drives the SCSI signal RST/. It is not connected to the NCR 5386.

NOTES:

1) NCR 5386 indicates the NCR 5385E, NCR 5386.

2) Input, Output, or I/O are indicated relative to the bus transceiver.

Power Signals

SIGNAL	DESCRIPTION
Vdd	5 Volts DC
Vss	Ground

Electrical Characteristics

Operating Conditions

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT
Supply Voltage	Vdd	4.75		5.25	Volts DC
Supply Current					
(NCR 8310)	Idd			145	mA
(NCR 83C10)	Idd*			5	mA
(NCR 83C11)	Idd*			5	mA
Ambient Free Air Temperature	Ta	0		70	°C
Hysteresis					
(NCR 8310 only)			250		mV
(NCR 83C10/C11)			450		mV
Latch-Up Protection (83C10/C11)			150		mA
Signal Rise/Fall Times					
(NCR 8310 only) (Ref. only)			22		nsec
(NCR 83C10/C11)					
Rise (Ref. only)		3		20	nsec
Fall (Ref. only)		3		15	nsec
Output Load Capacitance	C_L		100		pF

* Idd specified at a frequency of 0.25 MHz (83C10/C11)

Input Signal Requirements

PARAMETER	CONDITION	MIN	MAX	UNIT
High Level Input Voltage, Vih		2.0	Vdd	Volts DC
Low Level Input Voltage, Vil	8310 only	−0.3	0.6	Volts DC
	83C10/C11	−0.3	0.8	Volts DC
High Level Input Current, Iih				
SCSI:	Vih = Vdd		50	µA
Other Pins:	Vih = Vdd		10	µA
Low Level Input Current, Iil				
SCSI: (except RST/ for 83C10/C11)	Vil = 0 V		−50	µA
RST/: (83C10/C11)	Vil = 0 V		−90	µA
Other Pins:	Vil = 0 V		−10	µA
DBP: (8310 only)	VIl = 0 V		−2	mA

Output Signal Requirements

PARAMETER	CONDITION	MIN	MAX	UNIT
High Level Output Voltage, Voh				
All pins except RSTIN, SCSI:	Vdd = 4.75 V Ioh = −400 μA	2.4		Volts DC
RSTIN:	Vdd = 4.75 V Ioh = −800 μA	2.4		Volts DC
Low Level Output Voltage, Vol				
All pins except RSTIN, SCSI:	Vdd - 4.75 V Iol = 2.0 mA		0.5	Volts DC
RSTIN:	Vdd = 4.75 V Iol = 4.0 mA		0.5	Volts DC
SCSI:	Vdd = 4.75 V Iol = 48.0 mA		0.5	Volts DC

Aboslute Maximum Stress Ratings

Voltage on any pin with respect to ground (Vss)

NCR 8310 only	−0.3 to 7.0 V
NCR 83C10/C11	−0.3 to Vdd + 0.3 V

ESD Protection

NCR 8310 only	3,500 Volts
NCR 83C10/C11	4,000 Volts

Power Dissipation

NCR 8310 only	800 mW
NCR 83C10/C11	30 mW

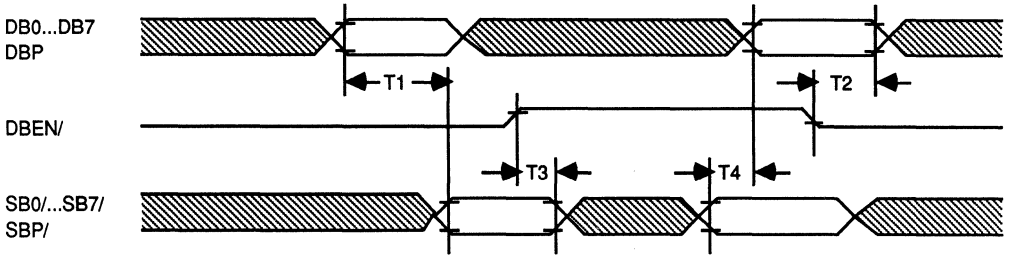
Storage Temperature Range −65 to 150 °C

The values listed here are absolute maximums which if exceeded could cause permanent damage to the device. All voltages are with respect to circuit ground.

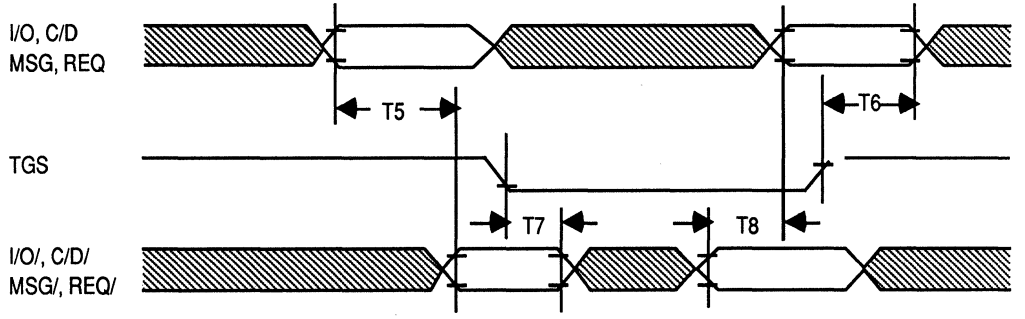
Chip Timing

Timing Diagrams

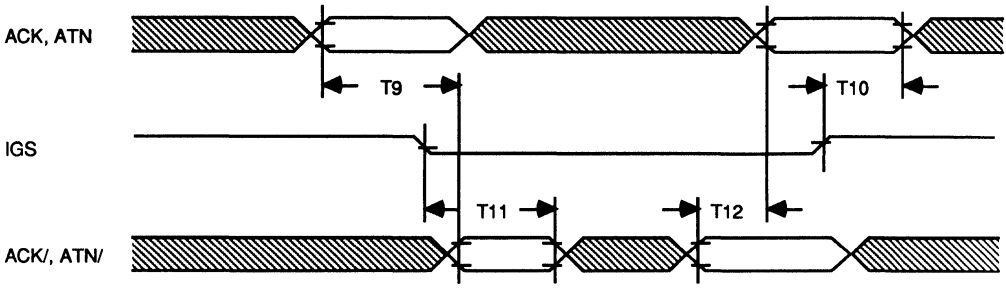
All AC timings are referenced to the .8 and 2.0 Volt Output levels.



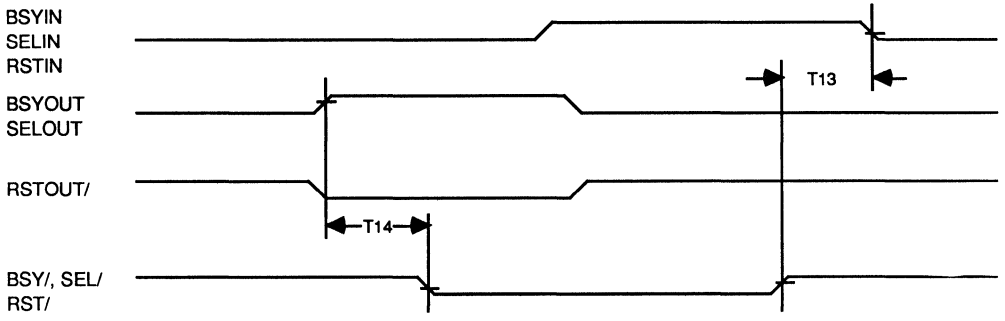
NAME	DESCRIPTION	8310 Max	83C10/C11 Max	UNIT
T1	Data Bus to SCSI Bus Delay	90	75	ns.
T2	DBEN true to Data Bus Tri-state	90	90	ns.
T3	DBEN false to SCSI Bus Release	100	100	ns.
T4	SCSI Bus to Data Bus Delay	65	65	ns.



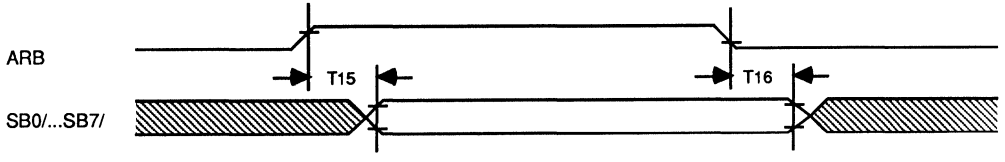
NAME	DESCRIPTION	8310 Max	83C10/C11 Max	UNIT
T5	Control Signal Delay to SCSI Bus	90	70	ns.
T6	TGS True to Input Release	65	50	ns.
T7	TGS False to SCSI Bus Release	90	70	ns.
T8	SCSI Bus Receiver Delay	65	65	ns.



NAME	DESCRIPTION	8310 Max	83C10/C11 Max	UNIT
T9	Control Signal Delay to SCSI Bus	90	70	ns.
T10	IGS True to Input Release	55	50	ns.
T11	IGS False to SCSI Bus Release	90	70	ns.
T12	SCSI Bus Receiver Delay	65	65	ns.



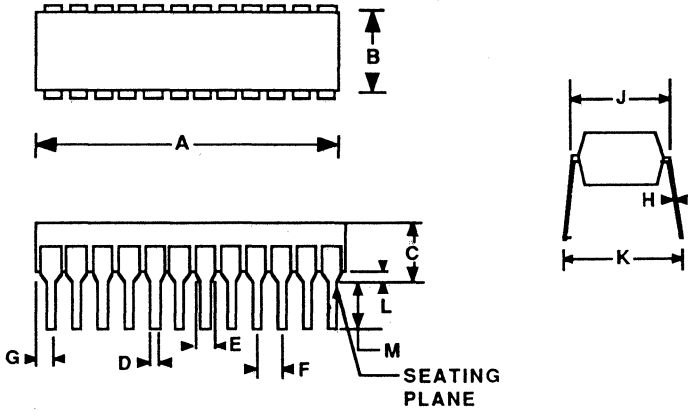
NAME	DESCRIPTION	8310 Max	83C10/C11 Max	UNIT
T13	Receiver Delay BSY RST SEL	65	65	ns.
T14	Driver Delay BSY RST SEL	85	70	ns.



NAME	DESCRIPTION	8310 Max	83C10/C11 Max	UNIT
T15	ARB True to DB(id) True	80	70	ns.
T16	ARB False to DB(id) False	80	70	ns.

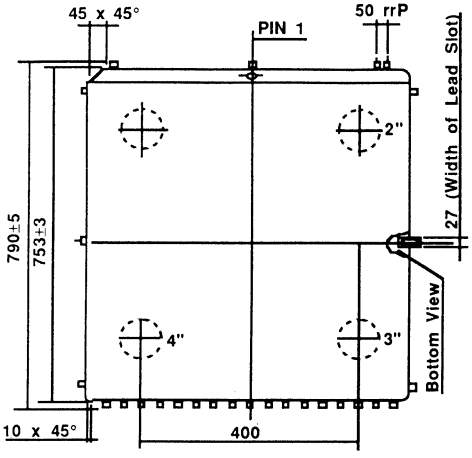
Mechanical Specifications

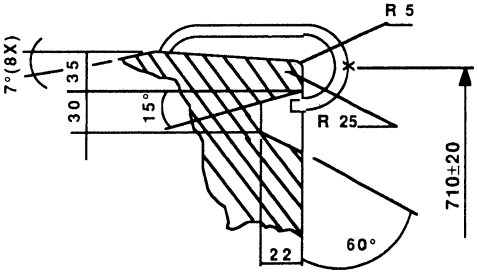
48 Pin Dual-In-Line Package



	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	59.18	62.48	2.330	2.460
B	13.08	14.73	0.515	0.580
C	2.41	5.84	0.095	0.230
D	0.38	0.56	0.015	0.022
E	0.38	1.27	0.015	0.050
F	2.29	2.79	0.090	0.110
G	0.51	1.91	0.020	0.075
H	0.20	0.36	0.008	0.014
J	14.97	15.75	0.590	0.620
K	14.48	17.78	0.570	0.700
L	0.38	1.52	0.015	0.060
M	2.54	5.59	0.100	0.220

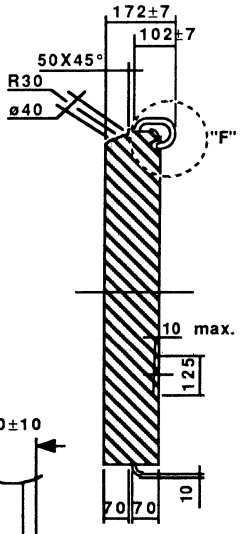
52 Pin Plastic Leaded Chip Carrier

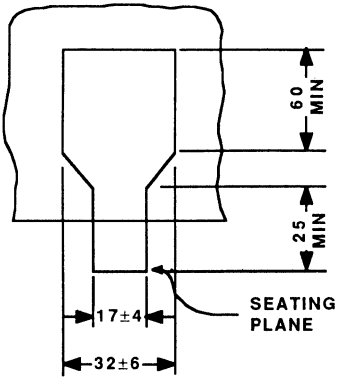


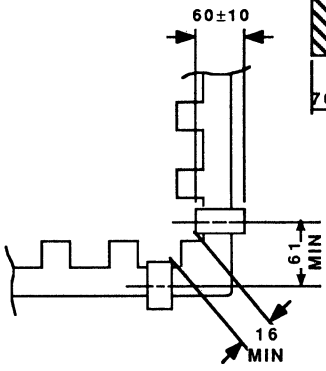


DETAIL OF "F"

X: SURFACE MOUNT POINT



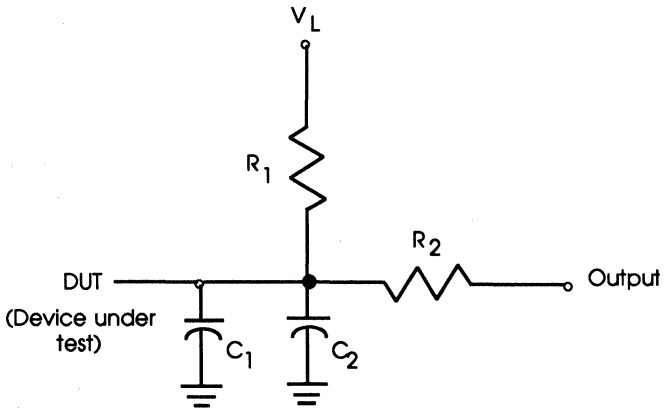




SCSI PRODUCTS

Application Notes

NCR 8310/83C10/83C11 Load Schematic



	SCSI PINS	ALL OTHER PINS
V_L	3.0 V	2.07 V
R_1	91Ω ± 1%	820Ω ± 1%

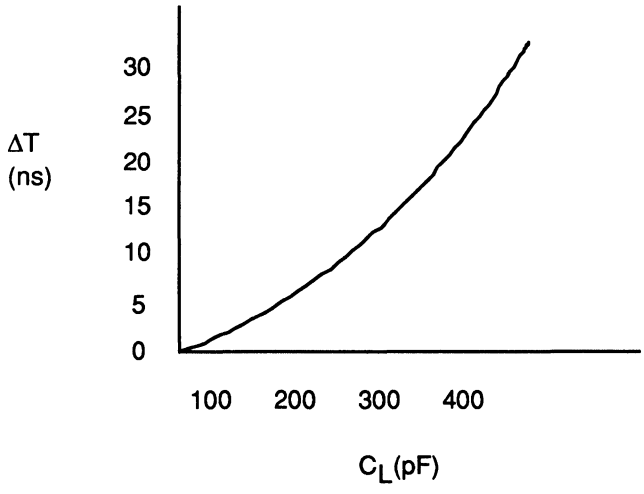
$C_L = C_1 + C_2$

C_1 = Socket/hardware capacitance = 50 pf

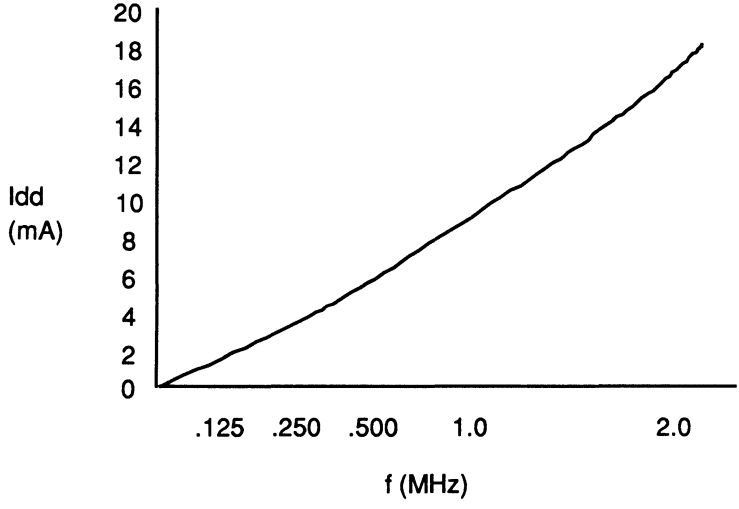
R_1 = External load resistor

R_2/C_2 = 50 Ω/50 pf inherent in tester pin electronics

NCR 83C10/83C11 Delta Time vs. Load Capacitance



NCR 83C10/83C11 Idd vs. Frequency



53C300 SCSI BUFFER CONTROLLER

Introduction

General Description

The NCR 53C300 SCSI Dual-Ported Buffer Controller combines the functionality of the NCR 53C80 SCSI Interface Chip with a dual-ported buffer controller. The SCSI portion of the chip supports the 1986 ANSI X3.131 SCSI Standard and can be used in both the initiator and target roles. The 53C300 supports arbitration, including reselection. Special high current single-ended output drivers, capable of sinking 48mA at 0.5V, allow for direct connection to the SCSI Bus.

The 53C300 communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing the internal registers which may be addressed as standard or memory-mapped I/O. Minimal processor intervention is required for DMA transfers because the chip controls the SCSI handshake signals. The 53C300 also interrupts the microprocessor when it detects a bus condition that requires attention.

The buffer controller portion of the chip is specifically designed to simplify buffering and increase the throughput of block-oriented high-performance peripheral controllers. The dual-ported buffer controller uses static RAM as a dual-ported circular FIFO. The controller supervises data transfers to the buffer which reduces the possibility of host overruns to the peripheral and allows for high-speed DMA transfers. The 53C300 also contains the necessary logic for resolving peripheral controller/SCSI transfer requests by giving priority to the peripheral controller with the SCSI request honored immediately afterward. The buffer controller can address buffer sizes from 256 bytes to 64K bytes.

The 53C300 contains a general purpose 4-bit output port. It supports Intel's multiplexed address/data bus scheme and also provides a demultiplexed address bus for use by other peripherals. This chip is intended for use in intelligent controllers and is software configurable via the multiplexed microprocessor I/O bus and is also adaptable to other microprocessor I/O buses that are not multiplexed. The 53C300 provides three input signals with internal pull-ups for SCSI device identification.

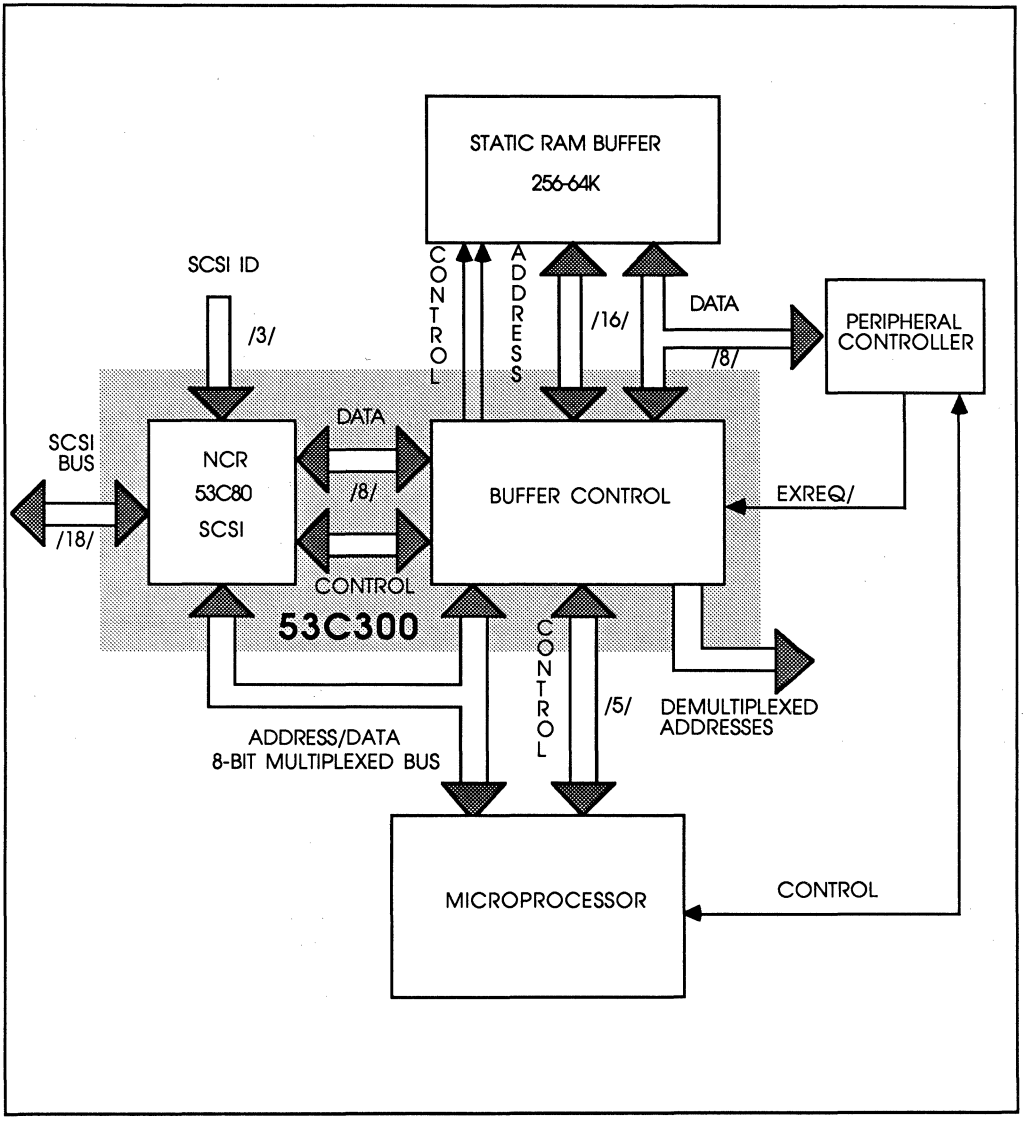
Another document which may prove helpful in understanding the 53C300 is the NCR 5380/C80 Design Manual. This, and information on other NCR products, may be obtained by calling the NCR Hotline: 1-800-334-5454.

Features Summary

- Supports the 1986 ANSI X3.131 SCSI Standard
- Uses static RAM as a dual-ported circular FIFO
- Handles buffer sizes from 256 to 64K bytes in 256-byte increments
- Supports Intel's multiplexed address/data bus
- Provides demultiplexed address bus for use by other peripheral devices which reduces external logic
- Provides three input signals with internal pull-ups for SCSI device identification
- Functionally compatible with the NCR 5380 family
- On-chip single-ended bus transceivers
- Flexible SCSI protocol control
- Stop pointer eliminates possibility of host overruns
- Additional 4-bit output port for general purpose use
- Parity generation with optional checking
- Functions in both the initiator and target roles
- Additional grounding and controlled fall times reduce noise generated by SCSI Bus switching
- Presents no D.C. load to the SCSI Bus when powered down
- CMOS low power requirements
- 84-pin PLCC package

Block Diagram

NCR 53C300 BLOCK DIAGRAM



Pin Information

(Note: A Slash "/" indicates an active low signal.)

Pin Description

Microprocessor Interface Signals

SYMBOL	PIN #	TYPE
ALE	21	Input

Address Latch Enable - The ALE Signal latches the address from the multiplexed Address/Data Bus (AD0-AD7). The falling edge of ALE latches the address onto the Address Bus (A0-A7).

PWR/	24	Input

Processor Write Request - The PWR/ Signal latches data from the multiplexed Address/Data Bus (AD0-AD7) to any selected internal register which can be written. The rising edge of PWR/ latches the data into the register. The address specified by the Address Bus (A0-A7) determines to which register the data is written.

PRD/	23	Input

Processor Read Request - The PRD/ Signal reads data from any selected internal register which is readable. The data will appear on the multiplexed Address/Data Bus (AD0-AD7). This data is valid on the rising edge of PRD/. The address specified by the Address Bus (A0-A7) determines from which register the data is read.

PCS	25	Input

Processor Chip Select - The PCS Signal enables the microprocessor access to all registers within the chip. If the chip is selected and the Address Bus (A0-A7) holds a valid internal register address, the PRD/ and PWR/ Signals can be enabled to read or write data. The ALE Signal can latch data onto the Address Bus (A0-A7) regardless of the logic state of this signal.

AD0-AD7	20-13	Input/Output

Address/Data Bus - The Address/Data Bus is an 8-bit bus multiplexed between address and data. ALE latches the address onto the Address Bus (A0-A7). During a read, (PRD/ active) these lines are driven by the chip; during a write, (PWR/ active) they are driven by the microprocessor.

SYMBOL	PIN #	TYPE
IRQ/	41	Output

Interrupt Request - The IRQ/ Signal is sent to the microprocessor when an interrupt condition has occurred within the chip.

A0-A7	33-26	Output

Address Bus - The Address Bus holds the last valid address to have appeared on the multiplexed Address/Data Bus (AD0-AD7). The falling edge of ALE latches the address of AD0-AD7 onto this bus.

Peripheral Interface Signals

These signals are used to interface the peripheral controller to the chip.

SYMBOL	PIN #	TYPE
EXREQ/	11	Input

External Data Transfer Request - The EXREQ/ Signal is asserted when the peripheral controller requests the next available memory cycle to do a transfer between itself and the buffer memory. It has priority over data transfer requests from the SCSI Bus.

HSD0-HSD7	74-67	Input/Output

High-Speed Data Bus - The High-Speed Data Bus allows the 53C300 and the peripheral controller to transfer data to the buffer memory. This common data bus is used during transfers between the SCSI Bus and the buffer memory or the peripheral controller and the buffer memory.

Buffer Control and Interface Signals

These signals are used to interface the chip to the buffer memory.

SYMBOL	PIN #	TYPE
OE/	10	Output

Output Enable - The OE/ Signal enables the buffer memory for a read. It is asserted when the buffer is read by the 53C300. The peripheral controller can read the buffer when the 53C300 asserts this signal.

SYMBOL	PIN #	TYPE
WR/	9	Output

Write Enable - The WR/ Signal enables the buffer memory for a write. It is asserted low when the 53C300 writes to the buffer. The peripheral controller can write to the buffer when the 53C300 asserts this signal.

MB0-MB15	7-1, 84-76	Output

Memory Buffer Address Bus - The Memory Buffer Address Bus selects the desired memory locations within the buffer memory. The value of the Read Address Pointer (RAP) appears on this bus during a buffer read, and the value of the Write Address Pointer (WAP) appears on the bus during a buffer write.

HSD0-HSD7	74-67	Input/Output

High-Speed Data Bus - The High-Speed Data Bus allows the chip and the peripheral controller to transfer data to the buffer memory. This common data bus is used during transfers with the SCSI Bus and the buffer memory or the peripheral controller and the buffer memory.

Miscellaneous Signals

SYMBOL	PIN #	TYPE
CLKA/	8	Input

Clock - The CLKA/ Signal synchronizes data transfers with the buffer memory and the peripheral controller.

PORST/	66	Input

Power On Chip Reset - The PORST/ Signal, when active, resets all registers in the 53C300. This signal will not assert the SCSI RST/ Signal.

D0-D2	36-34	Output

Data Output Pins - The Data Output Pins are general purpose output pins that can be controlled by accessing the D0-D2 Bits (Register 30, Bits 3-5). They can be used for general purpose control functions for external components. Outputs D0-D2 have 4mA source capability and 8mA sink capability.

LED/	64	Output

Light Emitting Diode Driver Output - The LED/ Signal can be used to drive an external light emitting diode. It can be controlled by accessing the LED/ Bit (Register 30, Bit 6). This output contains a special open-drain high-current output driver with 48mA sink capability.

SYMBOL	PIN #	TYPE
ID0/-ID2/	38-40	Input

SCSI ID Input Pins - The SCSI ID Input Pins are three inputs which may be read by reading the ID0/-ID2/ Bits (Register 30, Bits 0-2). These pins have on-chip pull-up resistors. Before starting the arbitration process, the microprocessor should read these bits, decode their value, and set up the correct value in the Output Data Register (Register 20).

TP/	37	Input

Test Pin - The Test Pin is intended for device functionality tests only. This pin has an on-chip pull-up resistor and should be tied high for normal operation.

SCSI Interface Signals

The following signals are all bi-directional, active low, open-drain signals. With 48mA sink capability, all pins interface directly with the SCSI Bus.

SYMBOL	PIN #	TYPE
ACK/	55	Input/Output

Acknowledge - The ACK/ Signal, driven by an initiator, indicates an acknowledgment for a SCSI data transfer handshake. In the target role, ACK/ is received as a response to the REQ/ Signal.

ATN/	53	Input/Output

Attention - The ATN/ Signal, driven by an initiator, indicates an attention condition. In the target role, ATN/ is received.

BSY/	54	Input/Output

Busy - The BSY/ Signal indicates that the SCSI Bus is being used and can be driven by both the initiator and the target device.

C/D/	60	Input/Output

Control/Data - The C/D/ Signal, driven by a target, indicates that control or data information is on the SCSI Bus. This signal is received by the initiator.

SYMBOL	PIN #	TYPE
I/O/	63	Input/Output

Input/Output - The I/O/ Signal, driven by a target, controls the direction of data transfer on the SCSI Bus. When active, this signal indicates input to the initiator. When inactive, this signal indicates output from the initiator. This signal is also used to distinguish between the Selection and Reselection Phases.

MSG/	58	Input/Output

Message - The MSG/ Signal is driven active by a target during the Message Phase. This signal is received by the initiator.

REQ/	61	Input/Output

Request - The REQ/ Signal, driven by a target, indicates a request for a SCSI data transfer handshake. This signal is received by the initiator.

RST/	57	Input/Output

Reset - The RST/ Signal indicates an SCSI Bus reset condition.

SYMBOL	PIN #	TYPE
SDB0/-SDB7/ SDBP/	42-43, 45-49, 51 52	Input/Output

SCSI Data Bits and Parity Bit - These eight Data Bits (SDB0/-SDB7/), plus a Parity Bit (SDBP/), form the SCSI Data Bus. SDB7/ is the most significant bit and has the highest priority ID during the Arbitration Phase. Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during arbitration.

SEL/	59	Input/Output

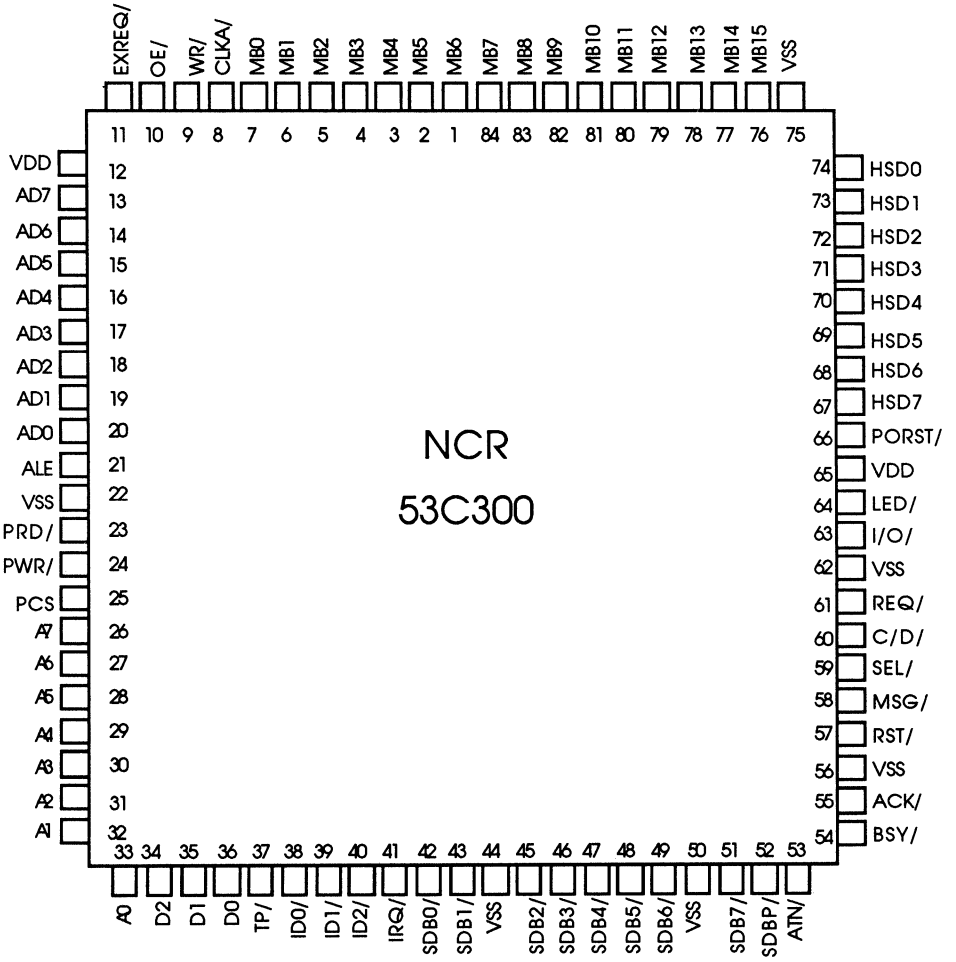
Select - The SEL/ Signal is used by an initiator to select a target or by a target to reselect an initiator.

Power Signals

SYMBOL	PIN #	TYPE
VDD	12, 65	+5 Volts
VSS	22, 44, 50, 56, 62, 75	Ground

Pin Diagram

NCR 53C300 PIN DIAGRAM



Registers

53C80 Registers

Current SCSI Data Register - Address 20 (Read Only)

The Current SCSI Data Register is a read-only register which allows the microprocessor to monitor the active SCSI Data Bus. If parity checking is enabled, the SCSI Data Bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during arbitration.

Output Data Register - Address 20 (Write Only)

The Output Data Register is a write-only register that sends data to the SCSI Data Bus. This register should contain the proper ID bits to drive the SCSI Data Bus during the Arbitration and Selection Phases.

Initiator Command Register - Address 21 (Read/Write)

The Initiator Command Register is a read/write register which asserts certain SCSI Bus signals, monitors those signals, and monitors the progress of SCSI Bus arbitration. Many of these bits are significant only when used as an initiator. However, most can be used during target role operation.

Bit 7 - Assert RST/ - Read/Write

When set, the RST/ Signal is asserted on the SCSI Bus. The RST/ Signal will remain asserted until this bit is reset or until PORST/ (Pin 66) is asserted. After this bit is set, IRQ/ goes active, all SCSI signals are removed except for RST/, and Registers 20 through 27 are reset.

Bit 6 - Arbitration In Progress - Read Only

This bit is used to determine if arbitration is in progress. For this bit to be active, the Arbitrate Bit (Register 22, Bit 0) must previously have been set. The Arbitration In Progress Bit indicates that a bus-free condition has been detected, and the chip has asserted BSY/ and the contents of the Output Data Register onto the SCSI Bus. Arbitration In Progress will remain set until the Arbitrate Bit is reset.

Bit 6 - Test Mode - Write Only

This bit may be set in a test environment to disable all output drivers. It should not be set for normal operation.

Bit 5 - Lost Arbitration - Read Only

When set, this bit indicates that the chip has detected a bus-free condition, arbitrated for use of the bus by asserting BSY/ and its ID on the SCSI Bus, and lost arbitration due to SEL/ being asserted by another bus device. For this bit to be set, the Arbitrate Bit (Register 22, Bit 0) must be set.

Bit 4 - Assert ACK/ - Read/Write

This bit is used by the initiator to assert ACK/ onto the SCSI Bus. In order to assert ACK/, the Target Mode Bit (Register 22, Bit 6) must not be set. Writing a zero to this bit de-asserts ACK/ on the SCSI Bus. Reading this register simply reflects the status of this bit.

Bit 3 - Assert BSY/ - Read/Write

When set, this bit asserts BSY/ onto the SCSI Bus. When reset, this bit de-asserts the BSY/ Signal. Asserting BSY/ indicates a successful attempt to gain control of the SCSI Bus. Resetting this bit creates a bus-disconnect condition, except during the Selection Phase. Reading this register simply reflects the status of this bit.

Bit 2 - Assert SEL/ - Read/Write

When set, this bit asserts SEL/ onto the SCSI Bus. SEL/ is normally asserted after arbitration has been successfully completed. When reset, this bit de-asserts the SEL/ Signal. Reading this register simply reflects the status of this bit.

Bit 1- Assert ATN/ - Read/Write

When set, this bit asserts ATN/ onto the SCSI Bus if the Target Mode Bit (Register 22, Bit 6) is not set. ATN/ is normally asserted by the initiator to request a Message Out Phase. When reset, this bit de-asserts the ATN/ Signal. Reading this register simply reflects the status of this bit.

Bit 0 - Assert Data Bus - Read/Write

When set, this bit allows the contents of the Output Data Register to be enabled as chip outputs on SCSI signals SDB0/ through SDB7/. Parity is also generated and asserted on SDBP/. When connected as an initiator, the outputs are only enabled if the Target Mode Bit (Register 22, Bit 6) is not set, and the phase signals C/D/, I/O/, and MSG/ match the contents of the Assert C/D/, Assert I/O, and Assert MSG/ in the Target Command Register. The Assert Data Bus Bit should also be set during DMA operations.

Mode Register - Address 22 (Read/Write)

The Mode Register is used to control the operation of the chip. This register determines whether the chip operates as an initiator or target, whether parity is checked, and whether interrupts are generated on various external conditions.

Bit 7 - Do Not Set

Bit 6 - Target Mode - Read/Write

When set, the chip operates as a SCSI Bus target device. When reset, the chip operates as a SCSI Bus initiator device. In order for the signals ATN/ and ACK/ to be asserted onto the SCSI Bus, the Target Mode Bit must not be set. In order for the signals C/D/, I/O/, MSG/, and REQ/ to be asserted onto the SCSI Bus, the Target Mode Bit must be set.

Bit 5 - Enable Parity Checking - Read/Write

This bit determines whether parity errors will be ignored or saved in the parity error latch. When set, parity errors are saved. When reset, parity errors are ignored.

Bit 4 - Enable Parity Interrupt - Read/Write

When set, this bit causes the IRQ/ Signal to be asserted if a parity error is detected. A parity interrupt will only be generated if the Enable Parity Checking Bit is also set.

Bit 3 - Enable EOP Interrupt - Read/Write

When this bit and the REOP Bit (Register 28, Bit 3) are set, the IRQ/ Signal is asserted when the DMA transfer is completed. The DMA transfer is completed when the value of STP is equal to the value of RAP during a DMA send operation, or equal to the value of WAP during a DMA receive operation.

Bit 2 - Monitor Busy - Read/Write

When set, this bit causes the IRQ/ Signal to be asserted when BSY/ unexpectedly changes from active to inactive. When the interrupt is generated, the lower six bits of the Initiator Command Register are reset and all signals are de-asserted on the SCSI Bus. The Busy Error Bit (Register 25, Bit 2) will also be set when this condition occurs.

Bit 1 - DMA Mode - Read/Write

The DMA Mode Bit allows a DMA transfer to occur and must be set prior to writing Registers 25 through 27. Registers 25 through 27 are used to start DMA transfers. The Target Mode Bit (Register 22, Bit 6) must be set prior to a write to Register 26 and reset prior to a write to Register 27. The Assert Data Bus Bit (Register 21, Bit 0) must be set for all DMA send operations. In the DMA Mode, REQ/ and ACK/ are automatically controlled.

The DMA Mode Bit is not reset when an internal 53C80 core End of DMA Transfer Process Signal (EOP/) is received. That is, it is not reset when the STP equals RAP or WAP. Any DMA transfer may be stopped by writing a zero into this bit location. However, care must be taken not to access Registers 20 through 27 when the internal 53C80 core DACK/ Signal is active. For this reason, the DMA Enable Bit (Register 28, Bit 1) has to be reset first to stop any potential conflict with DACK/. The SCSI BSY/ Signal must be active in order to set the DMA Mode Bit.

Bit 0 - Arbitrate - Read/Write

When set, this bit starts the arbitration process. Prior to setting this bit the Output Data Register should contain the proper SCSI device ID value. One data bit should be active for SCSI Bus arbitration. The chip will wait for a bus-free condition before entering the Arbitration Phase. The status of the Arbitration Phase may be determined by reading the Lost Arbitration and Arbitration In Progress Bits (Register 21, Bits 5 and 6 respectively).

Target Command Register - Address 23 (Read/Write)

The Target Command Register allows the microprocessor to control the SCSI Bus Information Transfer Phase and/or to assert REQ/ simply by writing this register. The Target Mode Bit (Register 22, Bit 6) must be set for bus assertion to occur. If connected as an initiator with the DMA Mode Bit (Register 22, Bit 1) set, and the I/O/, C/D/, and MSG/ phase lines do not match the phase bits in the Target Command Register, a phase mismatch interrupt will be generated when REQ/ goes active. In order to send data as an initiator, the Assert I/O/, Assert C/D/, and Assert MSG/ bits must match the corresponding bits in the Current SCSI Bus Status Register (Register 24).

Bit 7 - Last Byte Sent - Read Only

This bit indicates that the last byte of the DMA send operation has been sent on the SCSI Bus. This flag is necessary since the End of DMA Transfer Bit (Register 25, Bit 7) only reflects when the last byte was received from the buffer memory.

Bit 3 - Assert REQ/ - Read/Write

When this bit is set, REQ/ is asserted.

Bit 2 - Assert MSG/ - Read/Write

When this bit is set, MSG/ is asserted.

Bit 1 - Assert C/D/ - Read/Write

When this bit is set, C/D/ is asserted.

Bit 0 - Assert I/O/ - Read/Write

When this bit is set, I/O/ is asserted.

Current SCSI Bus Status Register - Address 24 (Read Only)

The Current SCSI Bus Status Register is a read-only register which monitors seven SCSI Bus control signals plus the SCSI Data Bus Parity Bit. When a bit is set, it represents an active signal; when a bit is not set or reset, it represents an inactive signal.

Bit 7 - RST/
Bit 6 - BSY/
Bit 5 - REQ/
Bit 4 - MSG/
Bit 3 - C/D/
Bit 2 - I/O/
Bit 1 - SEL/
Bit 0 - SDBP/

Select Enable Register - Address 24 (Write Only)

The Select Enable Register is a write-only register which masks all but a single ID bit during a selection attempt of the 53C300 by another device. The SCSI ID of the chip is written as a one (1) to the bit that corresponds to the SCSI ID value. For example, if SCSI ID 7 is the ID of the chip, then Bit 7 must be set for the interrupt to be generated after the chip is successfully selected. The simultaneous occurrence of the correct SCSI ID, BSY/ inactive, and SEL/ active will cause an interrupt. This interrupt can be disabled by resetting all bits in this register, effectively disallowing the chip to be selected. If the Enable Parity Checking Bit (Register 22, Bit 5) is active, parity will be checked during selection.

Bus and Status Register - Address 25 (Read Only)

The Bus and Status Register is a read-only register which monitors six status bits and the two SCSI control signals (ATN/ and ACK/) which are not found in the Current SCSI Bus Status Register (Register 24).

Bit 7 - End of DMA Transfer

This bit is set if the Buffer Control logic has terminated the DMA process. Since the termination happens as the last byte is transferred to the chip, the Last Byte Sent Bit (Register 23, Bit 7) must be monitored to ensure that the last byte sent to the Output Data Register (Register 20) has been transferred to the SCSI Bus. This bit is reset when the DMA Mode Bit (Register 22, Bit 1) is reset .

Bit 6 - DMA Request

This bit allows the microprocessor to sample the internal 53C80 core DMA Request (DRQ) Signal. DRQ can be cleared by an internal 53C80 core DMA Acknowledge (DACK/) Signal from the Buffer Control logic or by resetting the DMA Mode Bit (Register 22, Bit 1). The DRQ Signal does not reset when a phase mismatch interrupt occurs.

Bit 5 - Parity Error

This bit is set if a parity error occurs when receiving data or during a device selection. It can only be set if the Enable Parity Checking Bit (Register 22, Bit 5) is set. This bit may be cleared by reading the Reset Parity/Interrupt Register (Register 27).

Bit 4 - Interrupt Request Active

This bit indicates whether an interrupt condition has been detected within the 53C300 core. When set, the IRQ/ Signal is active; when reset, IRQ/ is inactive. It can be cleared by reading the Reset Parity/ Interrupt Register (Register 27).

Bit 3 - Phase Match

The SCSI MSG/, C/D/, and I/O/ Signals determine the current Information Transfer Phase. The Phase Match Bit indicates whether the current SCSI Bus phase matches the lower three bits of the Target Command Register (Register 23). The Phase Match Bit is continuously updated and is only significant when operating as a bus initiator. A phase match is required for data transfer to occur on the SCSI Bus.

Bit 2 - Busy Error

This bit is active if the BSY/ Signal is detected inactive for at least 400ns. The Busy Error Bit will disable any SCSI outputs and will reset the DMA Mode Bit (Register 22, Bit 1). This bit can be monitored to determine when the SCSI Bus is in a bus-free state.

Bit 1 - ATN/

This bit reflects the condition of the SCSI ATN/ Signal . When this bit is set, ATN/ is active. This signal is normally monitored by the target device. An active SCSI ATN/ can assert the IRQ/ by setting the Enable Attention Interrupt Bit (Register 28, Bit 4).

Bit 0 - ACK/

This bit reflects the condition of the SCSI Bus control signal ACK/. When this bit is set, ACK/ is active.

Start DMA Send Register - Address 25 (Write Only)

The Start DMA Send Register produces a strobe which starts a DMA send from the chip to the SCSI Data Bus. To initiate a DMA Send operation, the DMA Mode Bit (Register 22, Bit 1) and the DMA Enable Bit (Register 28, Bit 1) must be set. Any value written to this register will start the DMA send operation. The DMA Send operation can be initiated in either the initiator or the target mode.

Input Data Register - Address 26 (Read Only)

The Input Data Register is a read-only register that is used to receive data from the SCSI Data Bus during DMA transfers.

Start DMA Target Receive Register - Address 26 (Write Only)

The Start DMA Target Receive Register is written to initiate a DMA receive from the SCSI Data Bus to the chip for target operation only. To initiate a DMA Target Receive operation, the DMA Mode Bit (Register 22, Bit 1), the Target Mode Bit (Register 22, Bit 6), and the DMA Enable (Register 28, Bit 1) must all be set. Any value written to this register will start the DMA Target Receive Operation.

Reset Parity/Interrupt Register - Address 27 (Read Only)

The Reset Parity/Interrupt Register, when read, resets the Parity Error Bit, the Interrupt Request Bit, and the Busy Error Bit in the Bus and Status Register (Register 25).

Start DMA Initiator Receive Register - Address 27 (Write Only)

The Start DMA Initiator Receive Register is written to initiate a DMA Receive from the SCSI Data Bus to the chip for initiator operation only. To initiate a DMA Initiator Receive operation, the DMA Mode Bit (Register 22, Bit 1) and the DMA Enable Bit (Register 28, Bit 1) must be set, and the Target Mode Bit (Register 22, Bit 6) must not be set. Any value written to this register will start the DMA Initiator Receive Operation.

Buffer Control Registers

Memory Buffer Control Register - Address 28 (Read/Write)

The Memory Buffer Control Register controls SCSI DMA transfers. It also allows the microprocessor to access the buffer memory and contains the direction of transfer and end of DMA transfer information. Reading this register will strobe the contents of the RAP (Registers 2B and 2C) and WAP (Registers 2D and 2E) into their respective monitor registers.

Bit 7 - SCSI Transfer Started - Read Only

When set, this bit indicates that a SCSI transfer has started (at least one byte has been transferred). This bit is reset when either the STP (Registers 29 and 2A) is updated or the DMA Enable Bit is reset.

Bit 6 - Equal - Read Only

During DMA receive operations, this bit will be set when the STP is equal to the WAP. During DMA send operations, this bit will be set when the STP is equal to the RAP. This bit is reset when either the STP is updated or the DMA Enable Bit is reset.

Bit 5 - Status of SCSI ATN/ - Read Only

This bit reflects the status of the ATN/ Signal on the SCSI Bus. When this bit is set, ATN/ is active.

Bit 4 - Enable Attention Interrupt - Read/Write

When set, this bit enables the occurrence of an active SCSI ATN/ Signal to assert IRQ/. When reset, no IRQ/ will occur on the assertion of ATN/. This bit has no effect on the standard interrupt conditions detected within the 53C80 core.

Bit 3 - REOP - Read/Write

When set, this bit allows the Equal Bit to terminate a SCSI DMA operation by generating an internal 53C80 core End of DMA Transfer Process (EOP/) Signal. Resetting this bit will not generate an EOP/ when the Equal Bit is set. This bit can be used with the Enable EOP Interrupt Bit (Register 22, Bit 3) to signify the successful transfer of a block of data.

Bit 2 - Direction - Read/Write

This bit controls the direction of the data transfer. When set, the transfer direction is from the buffer to the 53C300. When reset, the transfer direction is from the 53C300 to the buffer. This bit should be set for DMA send operations and reset for DMA receive operations.

Bit 1 - DMA Enable - Read/Write

When set, this bit allows DMA transfers to occur. When reset, it prevents DMA transfers from occurring. When terminating a DMA transfer, this bit must be reset before resetting the DMA Mode Bit (Register 22, Bit 1).

Bit 0 - Access Memory - Read/Write

When set, this bit enables the microprocessor to access the buffer memory. The DMA Enable Bit must be reset, and the Direction Bit must be set when doing buffer memory reads and reset when doing buffer memory writes.

Stop Pointer (STP) Register - LSB - Address 29 (Read/Write)

The STP Registers contain the address of the last byte to be transferred to the buffer memory. This STP Register holds the LSB of the STP. This register must be written prior to writing the MSB (Register 2A).

STP Register - MSB - Address 2A (Read/Write)

This STP Register holds the MSB of the STP. Loading this location causes the STP (16 bit) to be updated to the new value now held within the STP Registers. When DMA is in progress, the value of the STP must not be incremented by less than two.

NOTE: Registers 2B to 2E have two parts. One part is the register itself, and the other part is a monitor register. When reading these registers, the value read is that of the monitor and not the register itself. In order to update the value in the monitor register to the present value of the actual register, a read of the Memory Buffer Control Register (Register 28) must be executed. This can be executed at any time while the DMA is in progress. Unlike the STP, either byte may be changed first, but the monitor register will not be changed until a read of Register 28 is performed.

Read Address Pointer (RAP) Register - LSB - Address 2B (Read/Write)

The RAP Registers contain the address in the buffer memory that will be read on the next read cycle. This RAP Register contains the LSB of the next buffer read address.

RAP Register - MSB - Address 2C (Read/Write)

This RAP Register contains the MSB of the next buffer read address.

Write Address Pointer (WAP) Register - LSB - Address 2D (Read/Write)

The WAP Registers contain the address in the buffer memory that will be written on the next write cycle. This WAP Register contains the LSB of the next buffer write address.

WAP Register - MSB - Address 2E (Read/Write)

This WAP Register contains the MSB of the next buffer write address.

Processor/Buffer Register - Address 2F (Read/Write)

The Processor/Buffer Register allows the microprocessor to access the buffer memory. It is important that the DMA Enable Bit (Register 28, Bit 1) is reset, and the Direction Bit (Register 28, Bit 2) is correctly written before accessing the memory. The procedure is as follows:

A. To read information from the buffer:

1. Stop all transfers from the peripheral controller and reset the DMA Enable Bit (Register 28, Bit 1).
2. Load the RAP (Registers 2B and 2C) with the buffer address to be read.
3. Set the Direction Bit (Register 28, Bit 2).
4. Set the Access Memory Bit (Register 28, Bit 0).
5. Read the byte from the Processor/Buffer Register (Register 2F). The RAP will automatically increment after the byte has been read.

B. To write information to the buffer:

1. Stop all transfers from the peripheral controller and reset the DMA Enable Bit (Register 28, Bit 1).
2. Load the WAP (Registers 2D and 2E) with the buffer address to be written.
3. Reset the Direction Bit (Register 28, Bit 2).
4. Set the Access Memory Bit (Register 28, Bit 0).
5. Write the byte to the Processor/Buffer Register (Register 2F). The WAP will automatically increment after the byte has been written to the buffer.

External Control Register - Address 30 (Read/Write)

The External Control Register monitors the status of three input pins and controls the status of four output pins on the chip.

Bit 7 - Not Used

Bit 6 - LED/ - Read/Write

When set, the open-drain LED/ Signal is driven low. Reading this bit reflects the status of this output.

Bit 5 - D2 - Read/Write

When set, the D2 Data Output Pin is driven high. Reading this bit reflects the status of this output.

Bit 4 - D1 - Read/Write

When set, the D1 Data Output Pin is driven high. Reading this bit reflects the status of this output.

Bit 3 - D0 - Read/Write

When set, the D0 Data Output Pin is driven high. Reading this bit reflects the status of this output.

Bit 2 - ID2/ - Read Only

This bit indicates the status of the SCSI ID2/ Input Pin. The SCSI ID0/-ID2/ Input Pins may contain the SCSI ID or can be general purpose input pins. When set, this bit indicates that ID2/ is active.

Bit 1 - ID1/ - Read Only

This bit indicates the status of the SCSI ID1/ Input Pin. When set, this bit indicates that ID1/ is active.

Bit 0 - ID0/ - Read Only

This bit indicates the status of the SCSI ID0/ Input Pin. When set, this bit indicates that ID0/ is active.

Functional Description

Reset Conditions

Hardware Chip Reset

The PORST/ Signal initializes the chip and clears all internal logic and control registers. This is a chip reset only and does not create a SCSI Bus reset condition.

SCSI Bus RST/ Received

When an active SCSI RST/ Signal is received, the IRQ/ Signal is asserted and the SCSI part of the chip is initialized. Registers 20 through 27 are cleared, but the Assert RST/ Bit (Register 21, Bit 7) remains unchanged.

SCSI Bus RST/ Issued

If the microprocessor sets the Assert RST/ Bit (Register 21, Bit 7), the RST/ Signal goes active on the SCSI Bus and an internal reset is performed. The SCSI part of the chip is initialized. Registers 20 through 27 are cleared, but the Assert RST/ Bit remains set. The RST/ Signal will continue to be active until the Assert RST/ Bit is reset or until a hardware reset occurs.

SCSI and Peripheral Request Prioritization

Both the SCSI interface and the peripheral controller can request a data transfer with the external buffer. When a DMA transfer is in operation, data is transferred to the buffer. If both a peripheral request and a SCSI interface request occur at the same time, the priority is given to the peripheral. The EXREQ/ Signal is sampled on the falling edge of CLKA/, and the internal 53C80 core DMA Request (DRQ) Signal is sampled on the rising edge of CLKA/. If the EXREQ/ Signal is active before DRQ becomes active, the EXREQ/ will be serviced first and the SCSI interface request is completed in the next available memory cycle. The DMA control interleaves SCSI and peripheral transfers. The peripheral requests are synchronous to the 53C300 clock (CLKA/, Pin 8), but the SCSI interface transfers are asynchronous.

Pointer Operation in Data Transfers

General Operation

Data transfers to the external buffer are controlled by three data transfer pointers: STP (Registers 29 and 2A), RAP (Registers 2B and 2C), and WAP (Registers 2D and 2E). The STP controls the number of data bytes being transferred. The RAP contains the read address of the buffer, and the WAP contains the write address of the buffer. When a DMA transfer occurs, one source will use the RAP and the other will use the WAP. The STP determines when the transfer will stop. For example, during a target receive operation, when the WAP is equal to the STP, the End of DMA Transfer Bit (Register 25, Bit 7) will be set indicating that all data received from the SCSI Data Bus has successfully been written to the buffer memory. Both the RAP and WAP are automatically advanced after a byte has been read/written into the buffer. The direction of the transfer is determined by the value of the Direction Bit (Register 28, Bit 2). If the Direction Bit is set, then data can be read from the buffer. If the Direction Bit is reset, then data can be written to the buffer.

During initiator or target DMA send operations, data is read from the buffer address as indicated by the current value of the RAP and latched in the Output Data Register (Register 20). During initiator or target DMA receive operations, data latched in the Input Data Register (Register 26) is written to the buffer address as indicated by the current value of the WAP.

SCSI Operation

When the SCSI Data Bus is ready to transfer data, an internal 53C80 core DMA Request (DRQ) Signal will become active. The data transfers from the SCSI Data Bus to the buffer can continue until the RAP or the WAP is equal to the STP. If the RAP or WAP is equal to the STP, then the Equal Bit (Register 28, Bit 6) will be set. The internal 53C80 core End of DMA Transfer Process (EOP/) Signal will be asserted if the REOP Bit (Register 28, Bit 3) is set. If the REOP Bit is set and the Equal Bit becomes set, then this condition will set the End of DMA Transfer Bit (Register 25, Bit 7) and assert IRQ/ if the Enable EOP Interrupt Bit (Register 22, Bit 3) is set. If the EXREQ/ Signal becomes active during a SCSI transfer, the peripheral controller request will be completed during the next available memory cycle.

Peripheral Operation

When the peripheral controller is ready to transfer data, it will assert EXREQ/. EXREQ/ must satisfy the setup time with respect to the CLKA/ Signal. Any current SCSI transfers will be completed. The peripheral controller request will be completed on the next available memory cycle. The buffer will be read/written at the current address indicated by the RAP or the WAP. If the internal 53C80 core DMA Request (DRQ) Signal becomes active during a peripheral controller transfer, then the SCSI transfer will be completed on the next available memory cycle.

Microprocessor Access to Internal Registers

The microprocessor interface is compatible with Intel's 8-bit address/data multiplexed bus scheme. The microprocessor can read or write any of the internal registers to initiate SCSI Bus activity or to assert any one of the SCSI Bus signals. The microprocessor must write the desired register address to the Address/Data Bus (AD0-AD7, Pins 20-13), and then assert the PCS Signal. After the address has been latched on the falling edge of ALE, then a read or write can be executed by asserting the PRD/ Signal or the PWR/ Signal. For a register read, the data will appear on the Address/Data Bus after a read access time delay. For a register write, the data must satisfy the setup time before the rising edge of PWR/ to ensure that the data is properly written to the chip.

Microprocessor Access to the Buffer

The microprocessor can set up the buffer address pointers by accessing the internal registers as described above. Since the RAP, WAP, and STP Registers contain 16-bit addresses, the microprocessor must write to the chip twice to change the addresses. The microprocessor can monitor the DMA transfers by reading the RAP/WAP counters while the DMA is in progress. It can also advance the STP while a DMA transfer is taking place, but only by two or more. Access to the RAP/WAP/STP while a DMA transfer is in progress does not have an adverse effect on the DMA throughput rate.

The microprocessor can access a byte of data stored in the buffer by setting the Access Memory Bit (Register 28, Bit 0). The DMA Enable Bit (Register 28, Bit 1) must not be set. The RAP/WAP must be loaded with the correct address and the Direction Bit (Register 28, Bit 2) must be set for a read and not set for a write. The data can be obtained by reading/writing the Processor/Buffer Register (Register 2F). The RAP/WAP will automatically be incremented after each byte is read/written.

Electrical Characteristics

D.C. Characteristics

Absolute Maximum Stress Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
Tstg	Storage Temperature	−55	150	°C
VDD	Supply Voltage	−0.5	7.0	V
VIN	Input Voltage	VSS − 0.5	VDD + 0.5	V
*ESD	Electrostatic Discharge (For all pads except SCSI)	−3000	3000	V
*ESD	Electrostatic Discharge (For all SCSI pads)	−4000	4000	V

*Tested using the human body model—100pF at 1.5kΩ

Operating Conditions

All timings and DC characteristics shall hold across the full range of temperature and voltage states below:

SYMBOL	PARAMETER	MIN	MAX	UNIT
VDD	Supply Voltage	4.75	5.25	V
IDD	Supply Current	0	25	mA
Ta	Operating Free-Air Temperature	0	70	°C
SRF	Signal Rise/Fall Time (SCSI pins)	8.0		ns

Microprocessor Interface Signals

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIL	Input Low Voltage	VSS − 0.5	0.8	V	
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VOL	Output Low Voltage	VSS	0.4	V	IOL = 3.20mA
VOH	Output High Voltage	2.4	VDD	V	IOH = −400uA
IOL	Output Low Current	4.0		mA	VOL = 0.4V
IOH	Output High Current	−2.0		mA	VOH = VDD − 0.5V
IIH	Input High Leakage	0	10	μA	VIN = VDD
IIL	Input Low Leakage	−10	0	μA	VIN = VSS

SCSI Signals

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIL	Input Low Voltage	VSS − 0.5	0.8	V	
VIH	Input High Voltage	2.0	VDD + 0.5	V	
VOL	Output Low Voltage	VSS	0.5	V	IOL = 48mA
VHYS	Hysteresis	200		mV	
IOL	Output Low Current	48		mA	VOL = 0.5V
IIL	Input Low Leakage	−10	0	μA	VIN = VSS
IIH	Input High Leakage	0	10	μA	VIN = VDD

Address Lines A0-A7

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4.0mA
VOH	Output High Voltage	2.4	VDD	V	IOH = −400uA
IOL	Output Low Current	4.0		mA	VOUT = VSS
IOH	Output High Current	−2.0		mA	VOUT = VDD

Buffer Control and Address Lines

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VOL	Output Low Voltage	VSS	0.4	V	IOL = 4.0mA
VOH	Output High Voltage	2.4	VDD	V	IOH = −400uA
VIL	Input Low Voltage (HSD0-HSD7)	VSS − 0.5	0.8	V	
VIH	Input High Voltage (HSD0-HSD7)	2.0	VDD + 0.5	V	
IOL	Input Low Current	4.0		mA	VOL = 0.4V
IOH	Input High Current	−2.0		mA	VOH = VDD − 0.5V
IIH	Input High Leakage	0	10	μA	VIN = VDD
IIL	Input Low Leakage	−10	0	μA	VIN = VSS

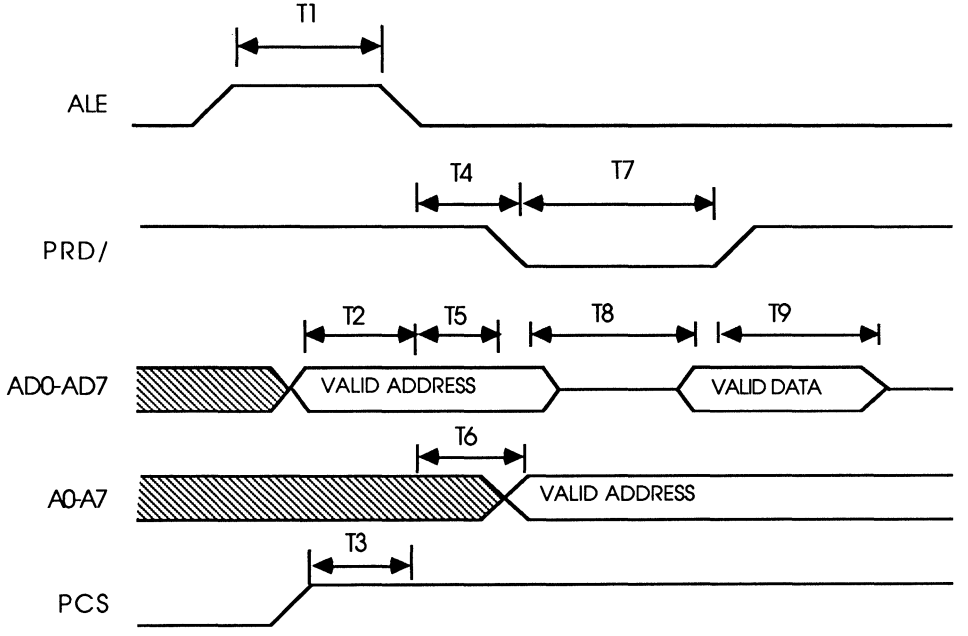
Miscellaneous Input and Output Pins

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITION
VIL	Input Low Voltage (PORST/, ID0-ID2/)	VSS − 0.5	0.8	V	
VIH	Input High Voltage (PORST/, ID0-ID2/)	2.0	VDD + 0.5	V	
VOL	Output Low Voltage (D0-D2)	VSS	0.4	V	IOL = 4mA
VOH	Output High Voltage (D0-D2)	2.4	VDD	V	IOH = −400uA
VOL	Output Low Voltage (LED/)	VSS	0.5	V	IOL = 48mA
IIH	Input High Leakage (PORST/, ID0/-ID2/, TP/)	0	10	μA	VIN = VDD
IIL	Input Low Leakage (Except ID0/ID2/, TP/)	−10	0	μA	VIN = VSS
IIL	Input Low Leakage (ID0/-ID2/, TP/)	−800	−50	μA	VIN = VSS

A.C. Characteristics

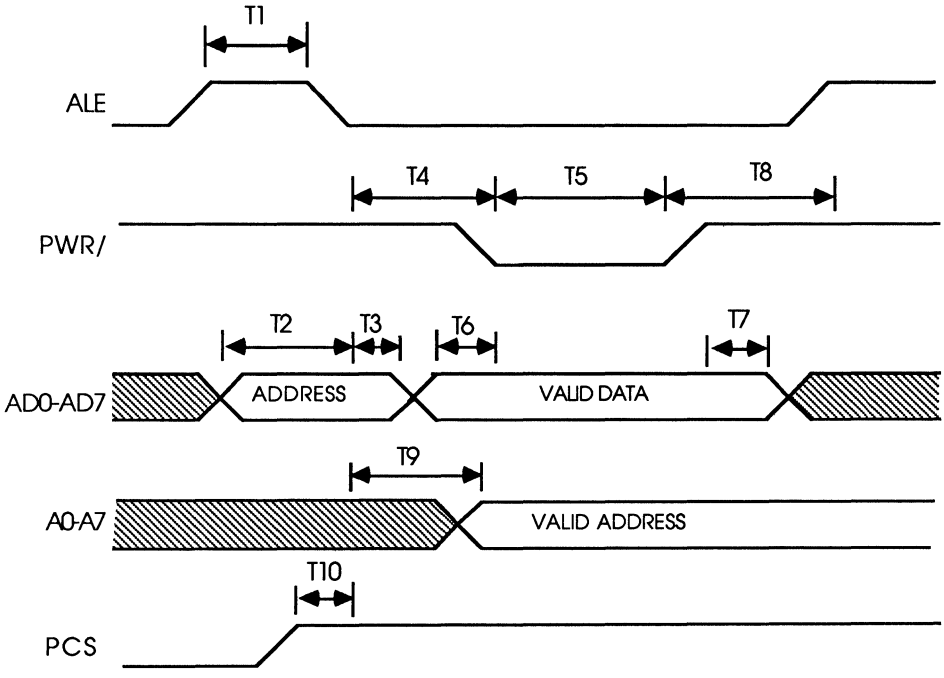
Microprocessor to Internal Register Read Timing

Symbol	Description	Min	Max	Unit
T1	ALE Pulse Width	60		ns
T2	Address Setup Time to ALE Low	5		ns
T3	PCS Setup Time to ALE Low	20		ns
T4	ALE Low to PRD/ or PWR/ Low	50		ns
T5	Address Hold after ALE Low	20		ns
T6	ALE Low to A0-A7 Address Valid		25	ns
T7	PRD/ Pulse Width	100		ns
T8	PRD/ Low to Valid Chip Data		100	ns
T9	Data Hold after PRD/	0	10	ns



Microprocessor to Internal Register Write Timing

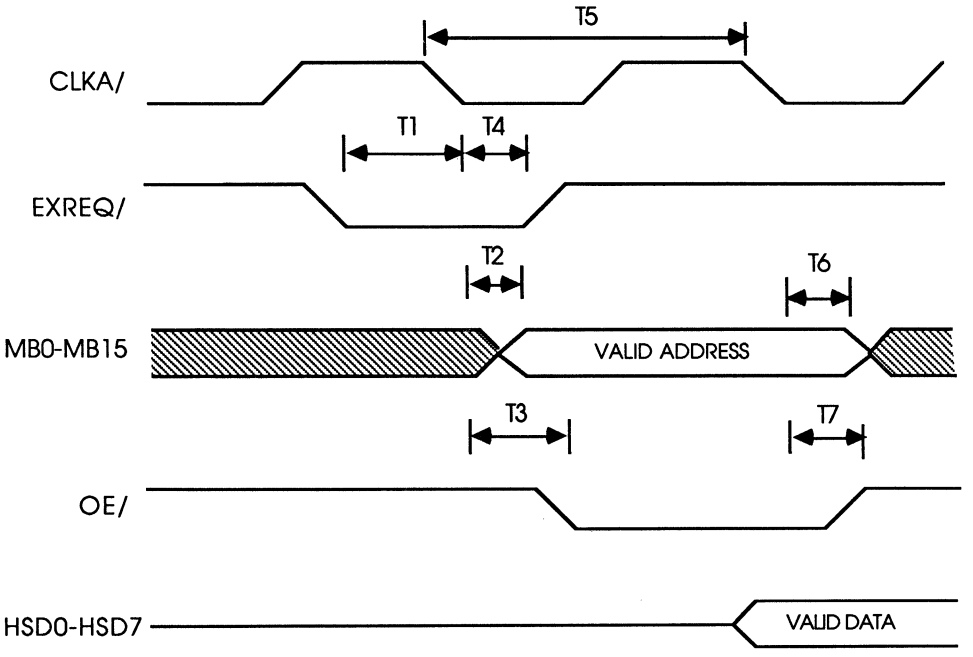
Symbol	Description	Min	Max	Unit
T1	ALE Pulse Width	60		ns
T2	Address Setup Time to ALE Low	5		ns
T3	Address Hold after ALE	20		ns
T4	ALE Low to PRD/ or PWR/ Low	50		ns
T5	PWR/ Pulse Width	100		ns
T6	Data Setup to PWR/ Transition	0		ns
T7	Data Hold after PWR/	12		ns
T8	PRD/ or PWR/ High to ALE High	20		ns
T9	ALE Low to A0-A7 Address Valid		25	ns
T10	PCS Setup Time to ALE Low	20		ns



Peripheral to Buffer Read Timing

Symbol	Description	Min	Max	Unit
T1	EXREQ/ Setup to CLKA/ Low	30		ns
T2	CLKA/ Low to New Address Valid		65	ns
T3	CLKA/ Low to OE/ Low		50	ns
T4	EXREQ/ Hold after CLKA/ Low	50	*	ns
T5	Period of CLKA/	200		ns
T6	CLKA/ Low to Address Invalid	15		ns
T7	CLKA/ Low to OE/ High	30	80	ns

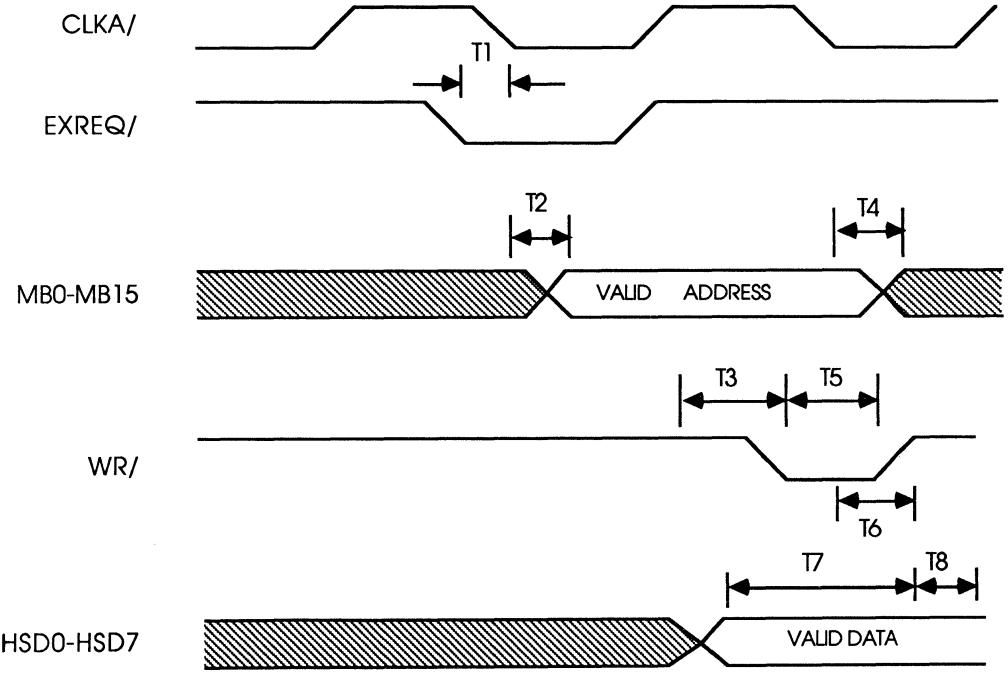
* (Period of CLKA/) - 30 ns



Peripheral to Buffer Write Timing

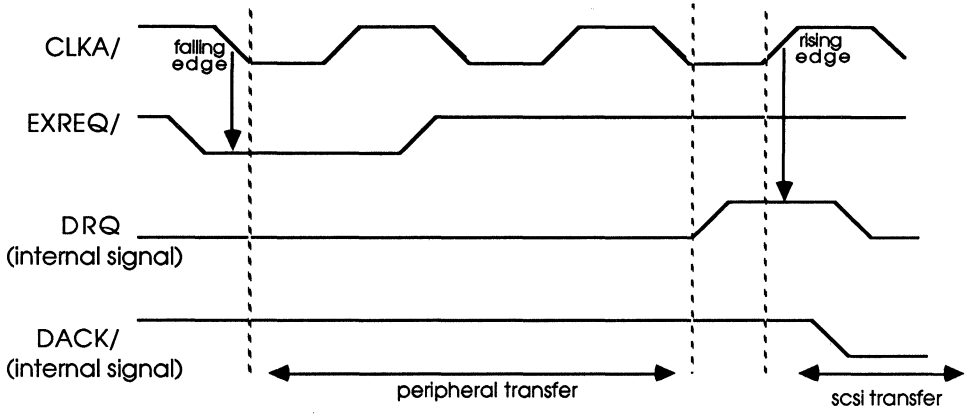
Symbol	Description	Min	Max	Unit
T1	EXREQ/ Setup Time to CLKA/ Low	30		ns
T2	CLKA/ Low to New Address Valid		65	ns
T3	CLKA/ High to WR/ Low		50	ns
T4	CLKA/ Low to Address Invalid	15		ns
T5	Width of WR/ Pulse	*	**	ns
T6	CLKA/ Low to WR/ High		50	ns
T7	Data Setup to WR/ High	***	***	ns
T8	Data Hold from WR/ High	***	***	ns

* (One-half of the period of CLKA/) - 10ns
** One half of the period of CLKA/
*** These timings depend on the characteristics of the memory device being used

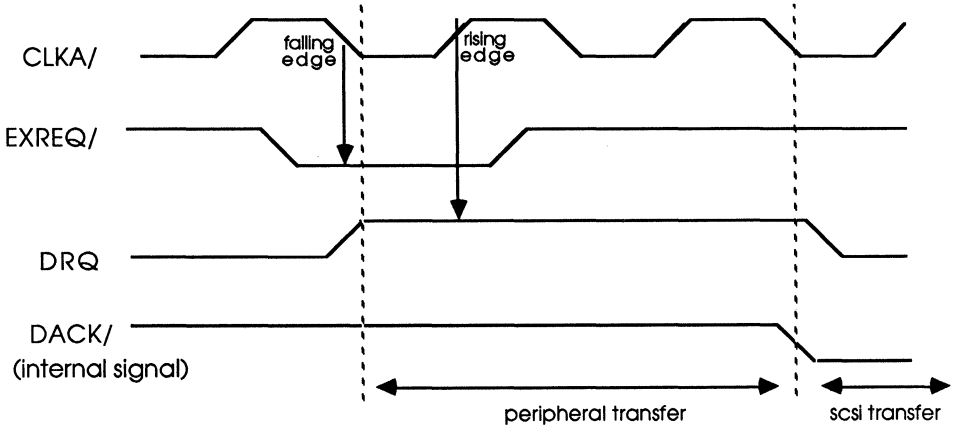


Appendix

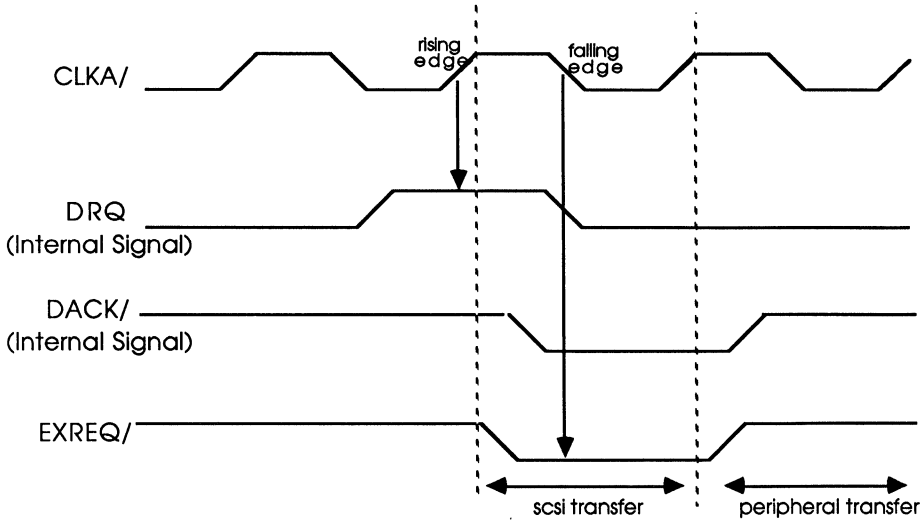
Data Transfer Request Prioritization



EXREQ/ is sampled on the falling edge of CLKA/. DRQ indicates an SCSI data transfer request. The DRQ Signal (internal to the chip) is sampled on the rising edge of CLKA/.

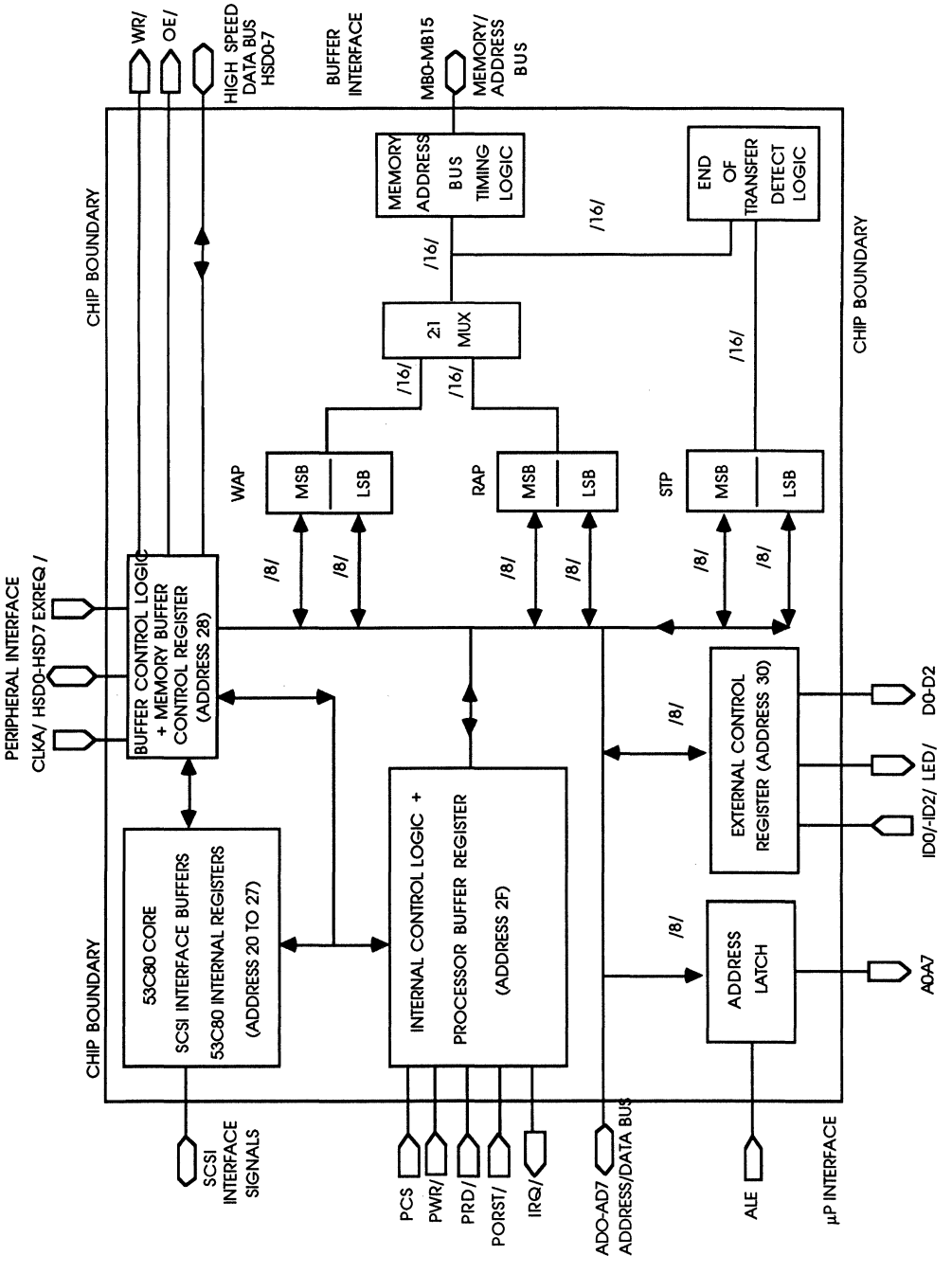


When an EXREQ/ is clocked on a falling edge of CLKA/ and a DRQ is clocked on the rising edge, priority is given to the signal that is detected first. When the EXREQ/ is given priority, the DRQ will be serviced after two clock cycles.

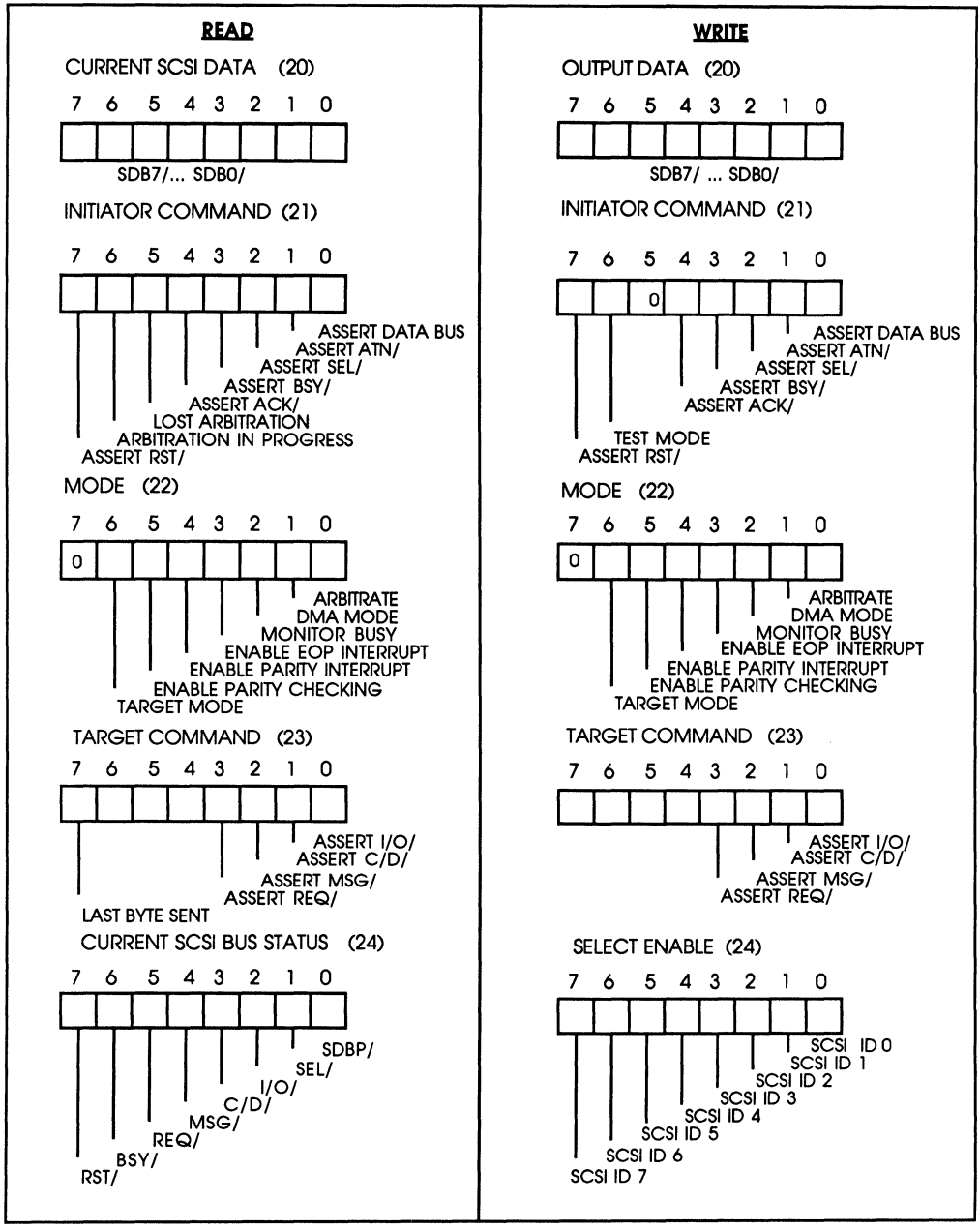


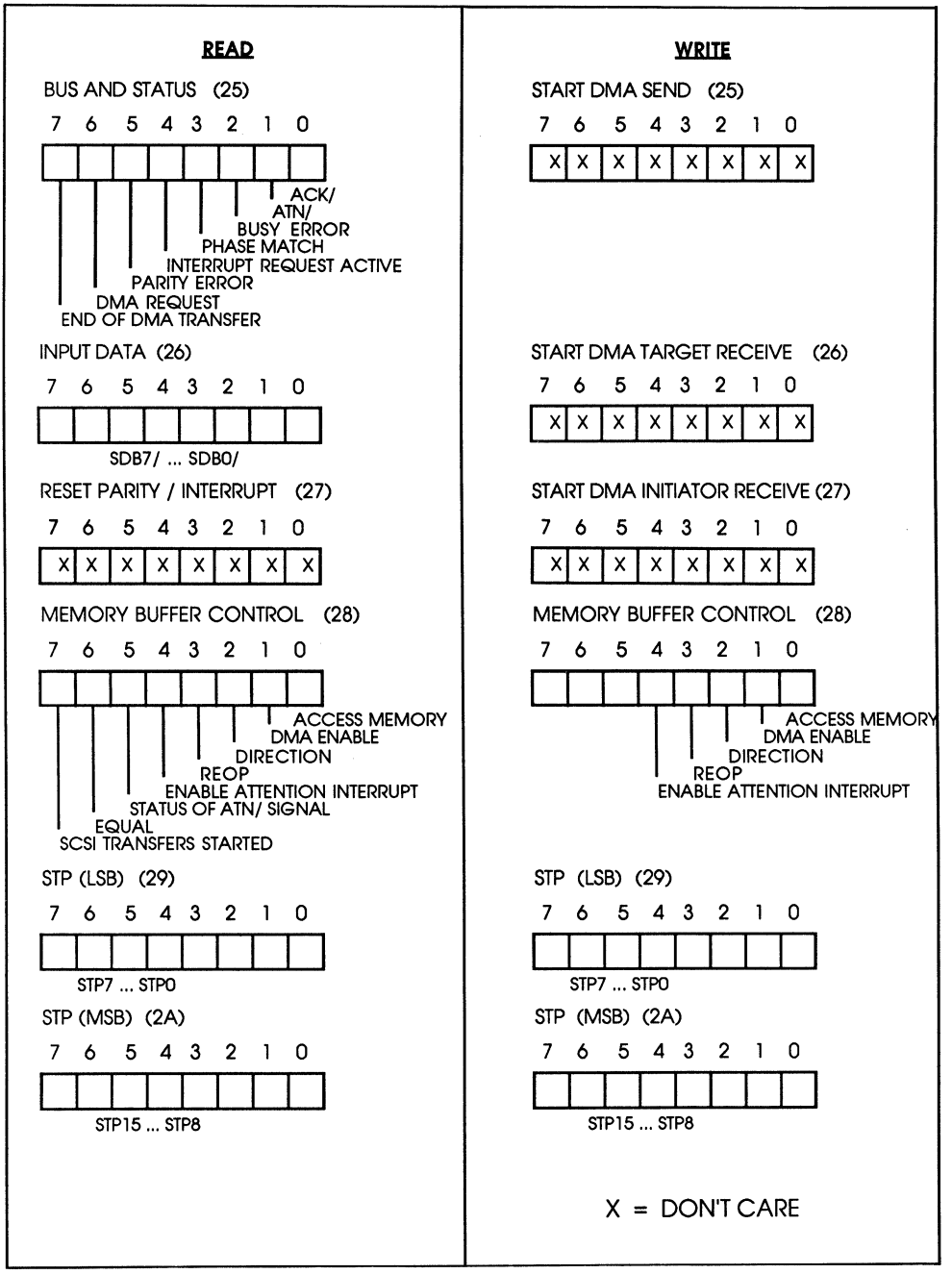
When a DRQ is clocked on a rising edge of CLKA/ and an EXREQ/ is clocked on the falling edge, priority is given to the signal that is detected first. When the DRQ is given priority, the EXREQ/ will be serviced after one clock cycle.

Data Flow Diagram



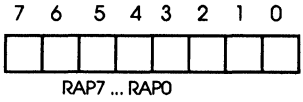
Register Summary



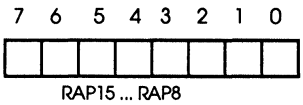

READ
BUS AND STATUS (25)
7 6 5 4 3 2 1 0
ACK/
ATN/
BUSY ERROR
PHASE MATCH
INTERRUPT REQUEST ACTIVE
PARITY ERROR
DMA REQUEST
END OF DMA TRANSFER
INPUT DATA (26)
7 6 5 4 3 2 1 0
SDB7/ ... SDB0/
RESET PARITY / INTERRUPT (27)
7 6 5 4 3 2 1 0
X X X X X X X X
MEMORY BUFFER CONTROL (28)
7 6 5 4 3 2 1 0
ACCESS MEMORY
DMA ENABLE
DIRECTION
REOP
ENABLE ATTENTION INTERRUPT
STATUS OF ATN/ SIGNAL
EQUAL
SCSI TRANSFERS STARTED
STP (LSB) (29)
7 6 5 4 3 2 1 0
STP7 ... STP0
STP (MSB) (2A)
7 6 5 4 3 2 1 0
STP15 ... STP8
WRITE
START DMA SEND (25)
7 6 5 4 3 2 1 0
X X X X X X X X
START DMA TARGET RECEIVE (26)
7 6 5 4 3 2 1 0
X X X X X X X X
START DMA INITIATOR RECEIVE (27)
7 6 5 4 3 2 1 0
X X X X X X X X
MEMORY BUFFER CONTROL (28)
7 6 5 4 3 2 1 0
ACCESS MEMORY
DMA ENABLE
DIRECTION
REOP
ENABLE ATTENTION INTERRUPT
STP (LSB) (29)
7 6 5 4 3 2 1 0
STP7 ... STP0
STP (MSB) (2A)
7 6 5 4 3 2 1 0
STP15 ... STP8
X = DON'T CARE

READ

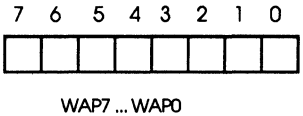
RAP (LSB) (2B)



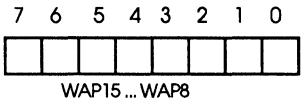
RAP (MSB) (2C)



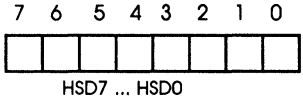
WAP (LSB) (2D)



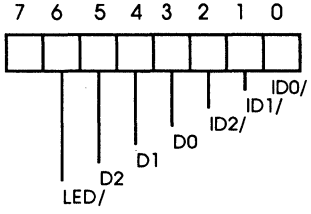
WAP (MSB) (2E)



PROCESSOR/BUFFER (2F)

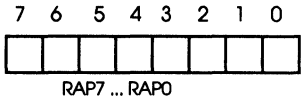


EXTERNAL CONTROL (30)

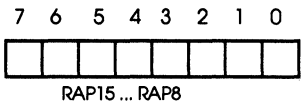


WRITE

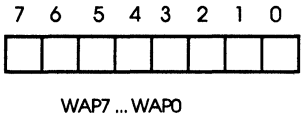
RAP (LSB) (2B)



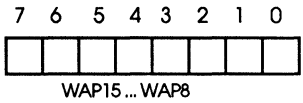
RAP (MSB) (2C)



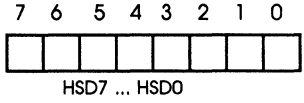
WAP (LSB) (2D)



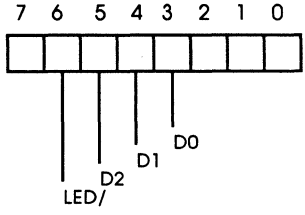
WAP (MSB) (2E)



PROCESSOR/BUFFER (2F)

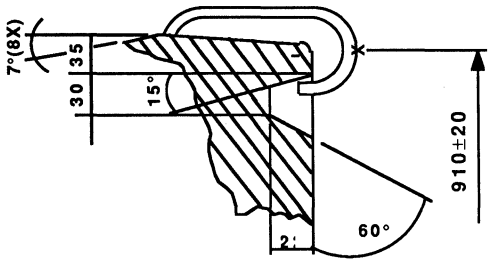


EXTERNAL CONTROL (30)



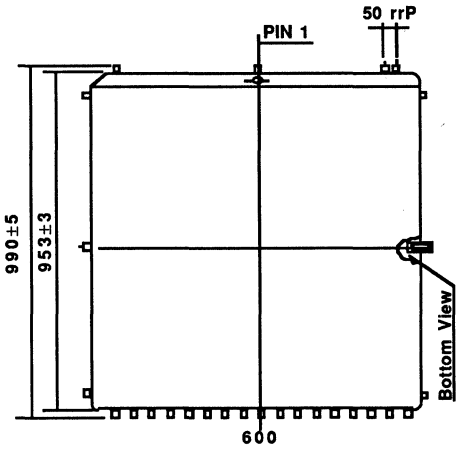
84 Pin PLCC Mechanical Drawing

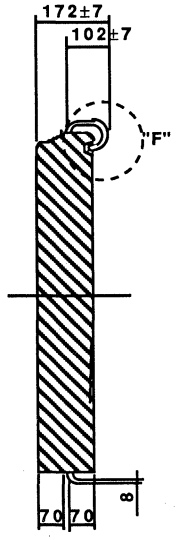
Note: All units are 1/1000 inches

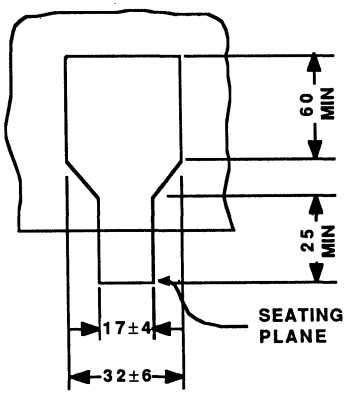


DETAIL OF "F"

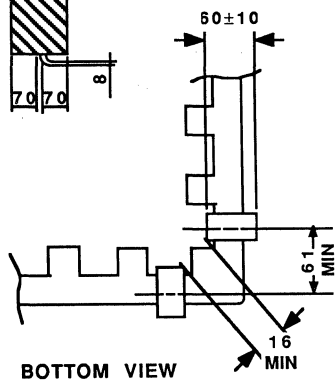
X: SURFACE MOUNT POINT







SIDE VIEW



BOTTOM VIEW

53C400 SCSI HOST ADAPTER

The NCR 53C400 SCSI Host Adapter is a 68-pin CMOS integrated circuit designed to interface the Small Computer Systems Interface (SCSI) Bus to an IBM PC, XT, AT or PS/2 Model 30 I/O channel bus. This high performance host adapter retains software compatibility with the NCR 5380/C80 SCSI chip while providing improved asynchronous bus performance, data buffering to match speeds and minimal external support logic. The 53C400 also provides special high-current output drivers capable of sinking 48 mA at 0.5 V, thereby allowing for direct connection to the SCSI Bus.

The 53C400 provides an eight-bit data interface for the IBM PC family. It communicates with the host microprocessor as a memory-mapped device and does not require the use of I/O address space. Base memory address is switch selectable between eight choices.

Data transfer is accomplished via Programmed Input/Output (PIO) operation from (to) the host memory to (from) the host adapter. DMA is not used. In order to match speeds between the SCSI and the host bus, two independent 128-byte data buffers are included in the 53C400. Therefore, simultaneous loading and unloading of these data buffers can occur from both the interfaces. Hence, data can be burst in excess of 1 Mbyte/s.

Six interrupt levels are programmable for the PC, XT and AT computers, while interrupt sharing as required by the PS/2 Model 30 system is supported. A non-interrupting operation of the host adapter is also supported.

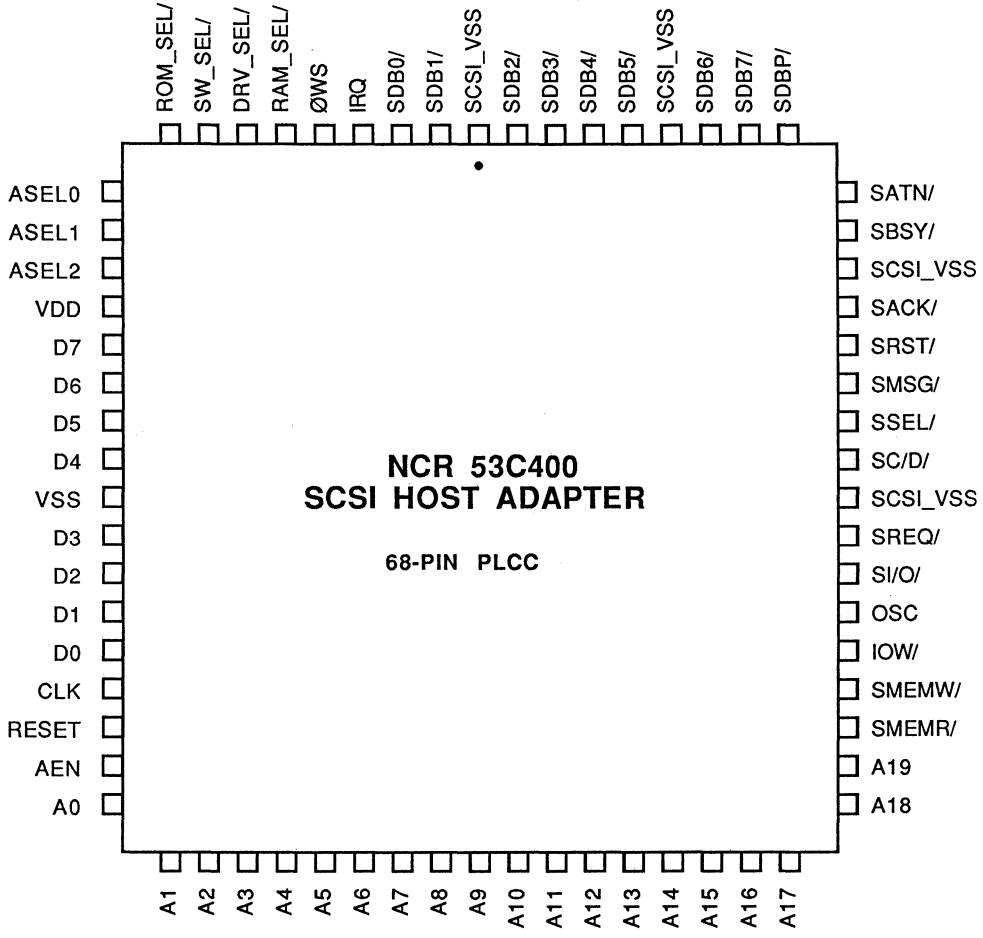
In order to ease the development of programs for the host adapter, the 53C80 register set is available. Also, a Control/Status Register, a Block Counter and a Switch Register have been included to facilitate the operation of the chip. An internal 64-byte RAM is also available for scratchpad space.

The 53C400, along with an external BIOS ROM module, can be a total solution for a host adapter. Address decoding is provided for an external static RAM up to 1.5 Kbytes. Convenient enabling signals such as RAMSEL, ROMSEL, SWSEL and DRVSEL are provided.

The 53C400 is available in a 68-pin Plastic Leaded Chip Carrier (PLCC) package.

The 53C400 Product Manual will be available the second quarter of 1988.

Pin Diagram


ROM_SEL/
SW_SEL/
DRV_SEL/
RAM_SEL/
ØWS
IRQ
SDB0/
SDB1/
SCSI_VSS
SDB2/
SDB3/
SDB4/
SDB5/
SCSI_VSS
SDB6/
SDB7/
SDBP/
ASEL0
ASEL1
ASEL2
VDD
D7
D6
D5
D4
VSS
D3
D2
D1
D0
CLK
RESET
AEN
A0
NCR 53C400
SCSI HOST ADAPTER
68-PIN PLCC
SATN/
SBSY/
SCSI_VSS
SACK/
SRST/
SMSG/
SSEL/
SC/D/
SCSI_VSS
SREQ/
SI/O/
OSC
IOW/
SMEMW/
SMEMR/
A19
A18
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17

85C20 ERROR-CORRECTING CODE CHIP

Introduction

General Description

The NCR 85C20 integrated circuit is a low-cost, high-performance, error-correcting code (ECC) IC which efficiently implements a high-performance Reed-Solomon code. This class of error-correcting codes was first discovered over twenty-five years ago, but has only recently become economical to implement. The NCR 85C20 IC represents a new step forward in implementation efficiency, translating to lower cost, which has been accomplished with proprietary techniques developed by Data Systems Technology, Corp. (DST).

The NCR 85C20 IC is targeted for optical and magnetic storage applications, as well as a wide range of communications applications. Currently, considerable interest in the NCR 85C20 IC is coming from manufacturers of 5.25 inch optical disk drives and controllers; the codes implemented in the NCR 85C20 IC have been endorsed by U.S., Japanese, and international standards committees as the recommended codes for 5.25 inch optical storage devices. Interest from the magnetic disk and tape communities is also increasing, now that the optical storage industry has shown that high-performance error correction is feasible for low-cost devices. A segment of the magnetic disk industry is expected to use the NCR 85C20 IC to cut the cost of media, testing, and support by handling higher defect densities.

Error-correction functions are divided between the NCR 85C20 IC and companion software routines. The IC generates ECC redundancy bytes (and CRC redundancy bytes, if the optional CRC code is used) during writes and ECC remainder bytes (and CRC residue bytes, if used) during reads. The software routines perform algorithms on the ECC remainder bytes to determine error locations and values and correct the errors. The CRC residue bytes (if used) are adjusted and checked for zero to verify the error locations and values.

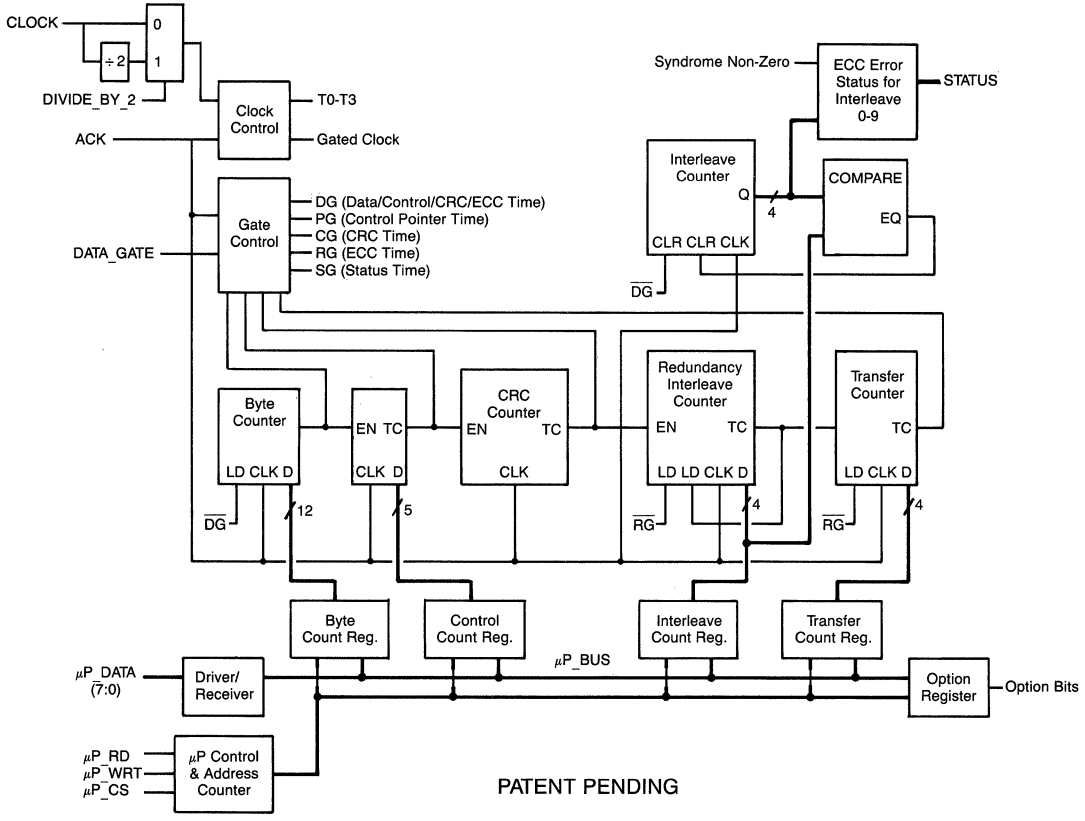
Features Summary

Some of the features of the NCR 85C20 IC are outlined below.

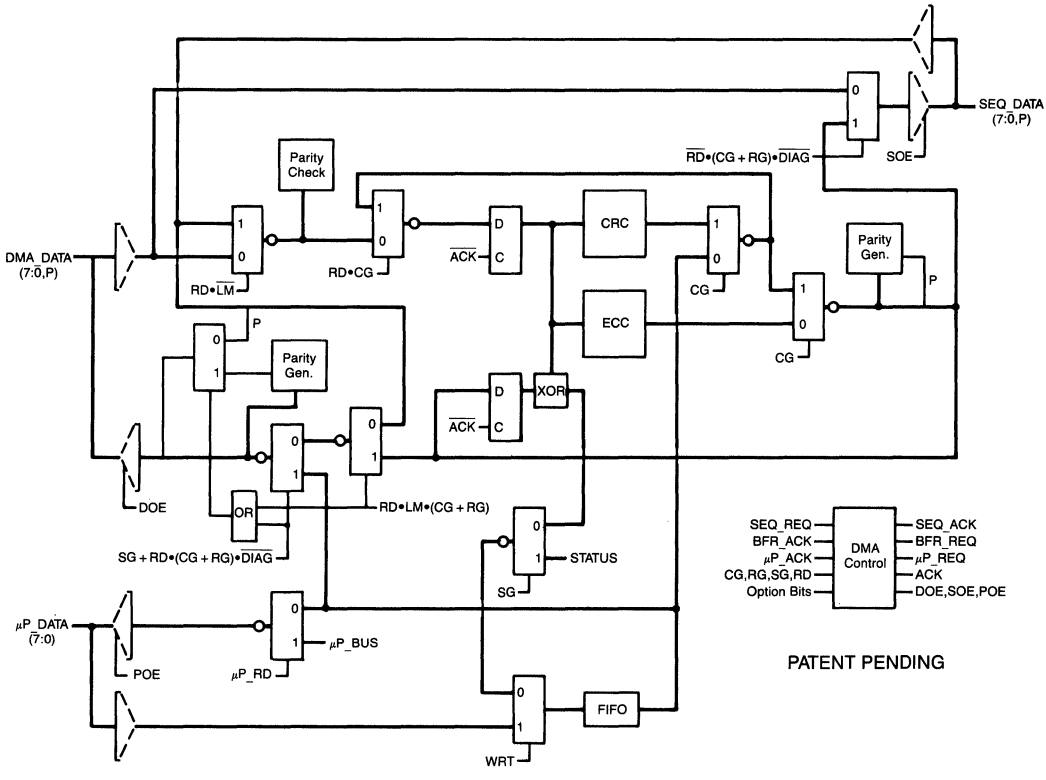
- Transforms a high raw error rate into a low corrected error rate at low cost.
- Makes feasible fast "on-the-fly" error correction using a long-distance Reed-Solomon code.
- Supports data transfer rates up to 32 megabits per second.
- Supports three to ten interleaves per physical sector.
- Implements ECC and CRC codes endorsed by U.S., Japanese, and international standards committees for optical storage applications.
- Supports variable redundancy from two to sixteen bytes per interleave in two byte increments.
- Logical sectors of arbitrary length can be made up of multiple contiguous physical sectors.
- Flexible implementation alternatives make the NCR 85C20 IC ideal for many application areas.

Block Diagram

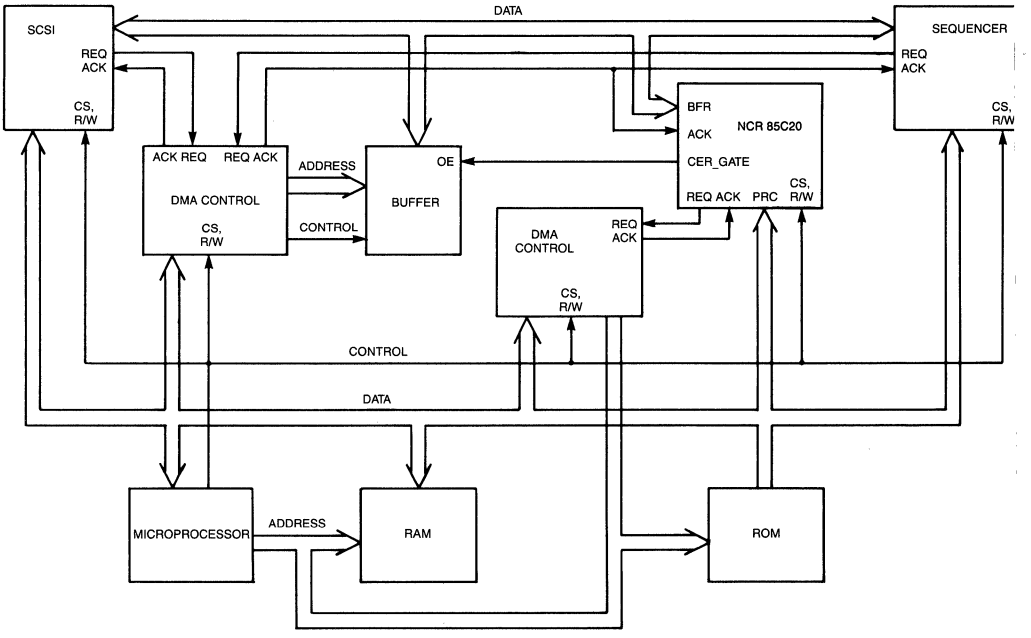
Control/Counter Logic



Data Flow

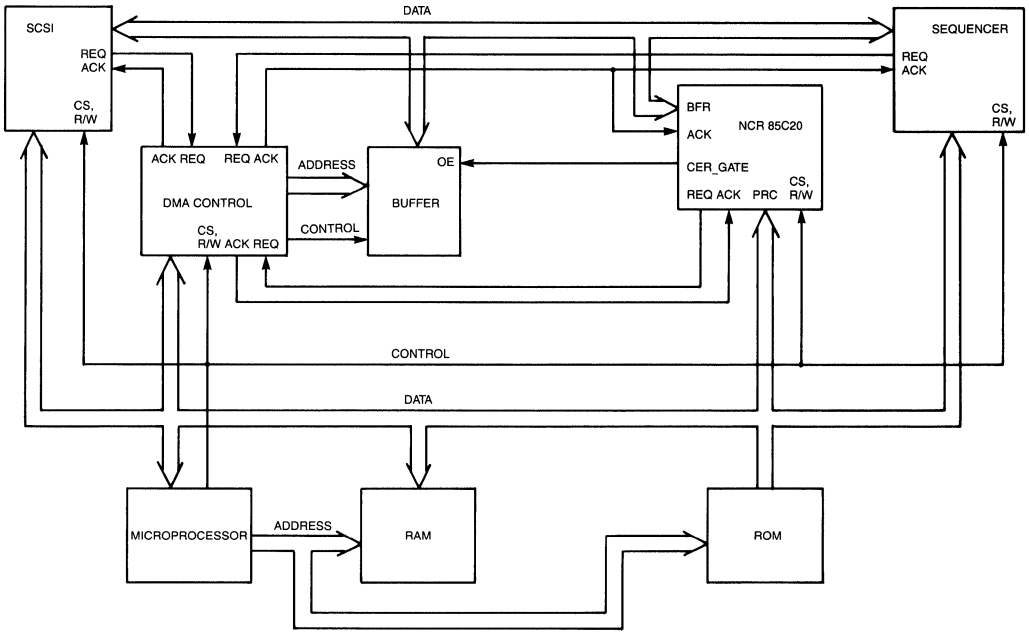

SEQ_DATA (7:0,P)
SOE
RD•(CG + RG)•DIAG
Parity Check
DMA_DATA (7:0,P)
RD•LM
RD•CG
ACK
D
C
CRC
CG
Parity Gen.
P
ECC
CG
P
Parity Gen.
XOR
ACK
DOE
OR
RD•LM•(CG + RG)
SG + RD•(CG + RG)•DIAG
STATUS
SG
μP_DATA (7:0)
POE
μP_RD
μP_BUS
FIFO
WRT
SEQ_REQ
BFR_ACK
μP_ACK
CG,RG,SG,RD
Option Bits
DMA Control
SEQ_ACK
BFR_REQ
μP_REQ
ACK
DOE,SOE,POE
PATENT PENDING

Configuration Examples



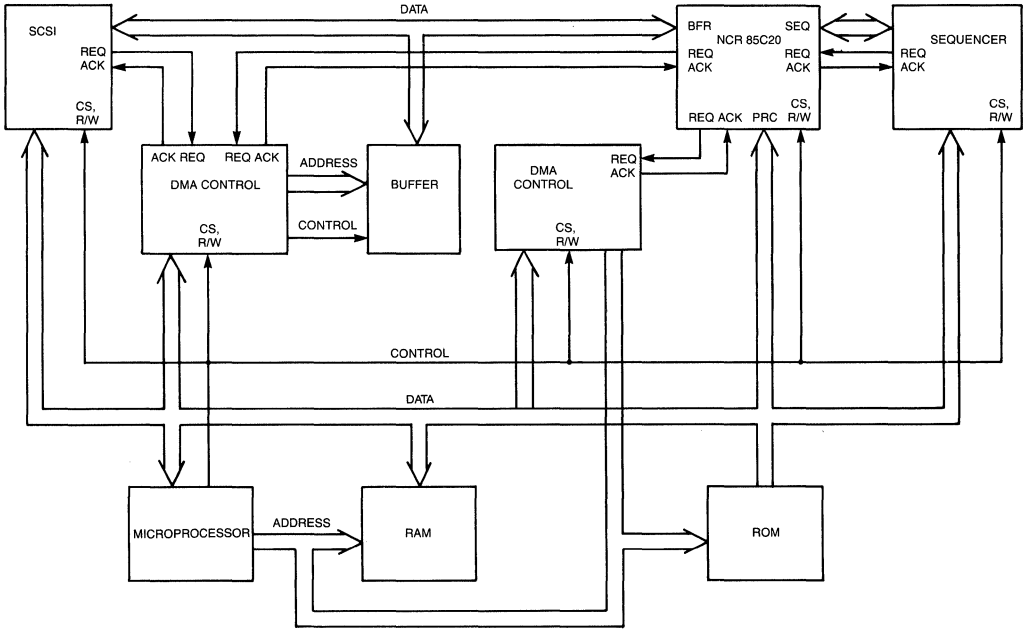
Listen Mode When Transferring Remainder/Status Bytes to Microprocessor Port

Note: For both read and write, data of consecutive sectors will be non-contiguous since requests must be made of the buffer's DMA controller to transfer the CRC/ECC residue/redundancy bytes to and from the sequencer.

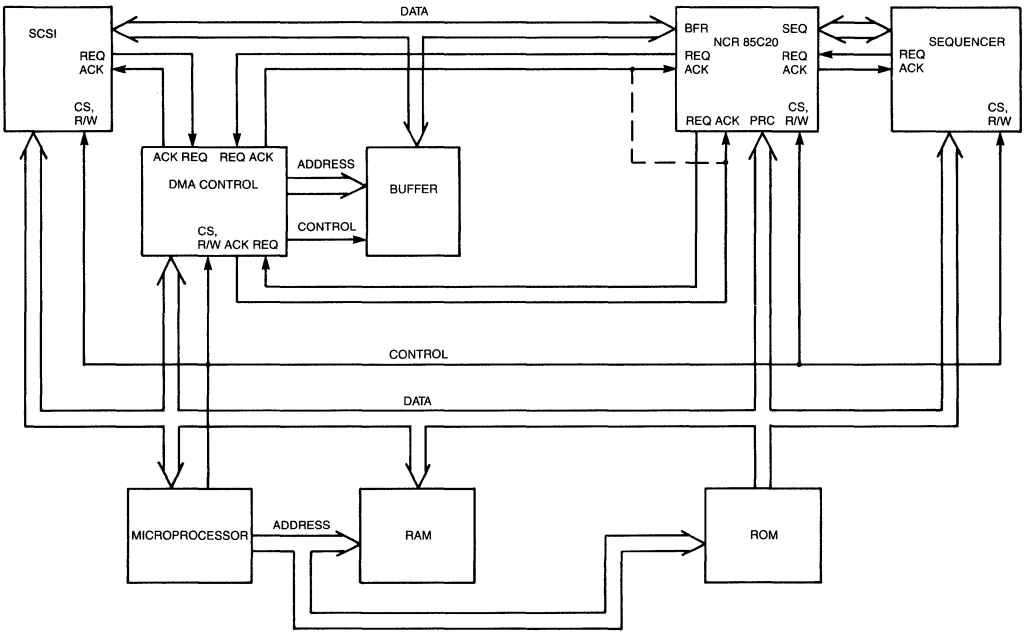


Listen Mode When Transferring Remainder/Status Bytes to Buffer Port

Note: For both read and write, data of consecutive sectors will be non-contiguous since requests must be made of the buffer's DMA controller to transfer the CRC/ECC residue/redundancy bytes to and from the sequencer. The CRC residue, ECC remainder, and status bytes will be in a separate area of buffer RAM since an independent DMA channel is used to transfer them.



Transparent Mode When Transferring Remainder/Status Bytes to Microprocessor Port



Transparent Mode When Transferring Remainder/Status Bytes to Buffer Port

Note: A variation of this configuration is shown by the dashed line and selection of XFR_REM_REM_WITH_BFR_REQ mode. This would eliminate the requirement for another DMA channel, but would result in the read data from consecutive sectors being non-contiguous in the buffer, with each sector's data bytes followed by its ECC remainder bytes.

Pin Information

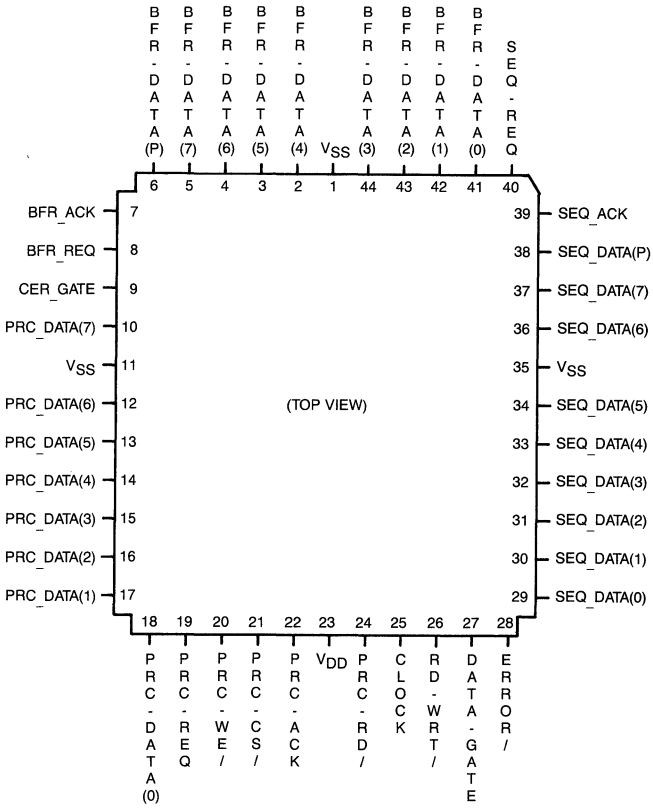
Pin Descriptions

References to CRC are understood to apply only when the optional CRC code is selected.

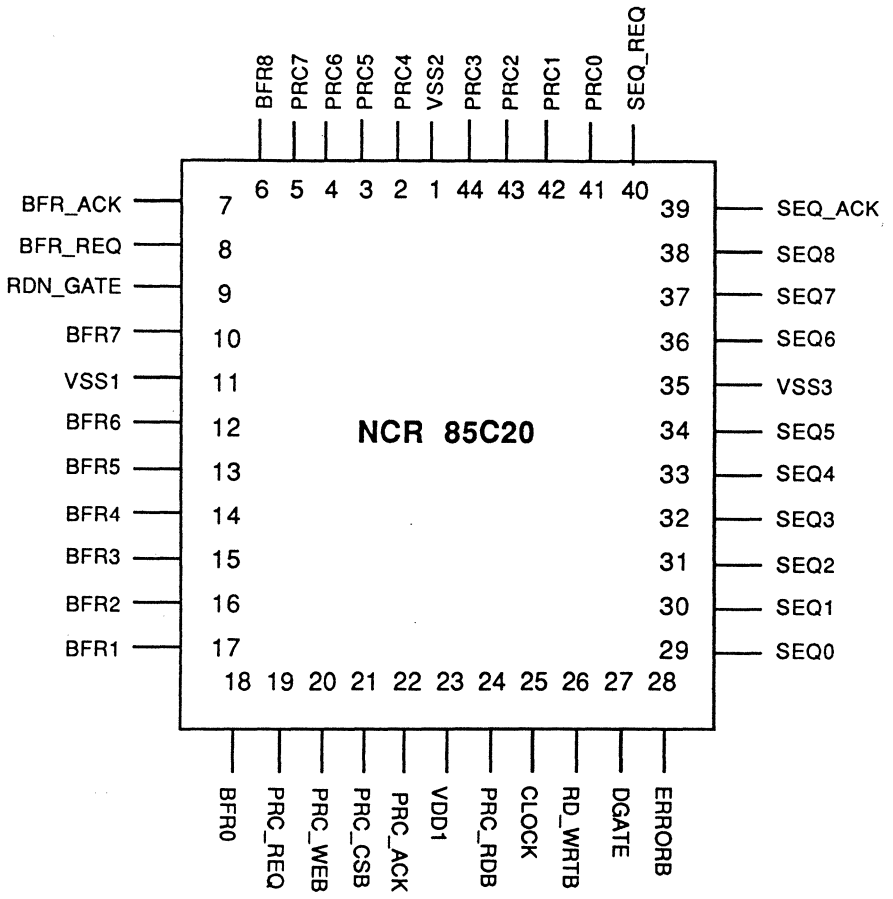
Symbol	Type	Description
CLOCK	I	The bit clock. If the divide-by-two option is chosen, the frequency of this input must be equal to or greater than the bit rate. If the divide-by-two option is NOT chosen, it must be equal to or greater than one-half the bit rate.
RD_WRT/	I	An input which places the IC in the Read Mode if high or the Write Mode if low.
DATA_GATE	I	An input which defines the beginning of the data for which the ECC redundancy bytes are to be calculated. DATA_GATE must be deasserted prior to the assertion of BFR_ACK for the last check byte of a sector. The first assertion of BFR_ACK following the assertion of DATA_GATE defines the beginning of data.
CER_GATE	O	A high-active output asserted during CRC redundancy time and ECC redundancy time. In the Listen Mode, it can be used to block the output from the buffer while the NCR 85C20 IC is outputting the write CRC/ECC redundancy bytes. It can also be used during reads to block writing the read CRC residue/ECC redundancy bytes to the buffer. During Diagnostic and Listen modes, this output is forced inactive.
ERROR/	O	A low-active output asserted if the IC detects any parity error, hardware error, ECC error, or CRC error.
BFR_DATA	I/O	A nine-bit wide port for data to and from the data buffer. Parity may optionally be generated and checked.
BFR_REQ	O	An output for requesting the transfer of data from the buffer.
BFR_ACK	I	An input to acknowledge the request and gate data onto the data bus.
SEQ_DATA	I/O	A nine-bit wide port for data to and from the sequencer. Parity may optionally be generated and checked.
SEQ_REQ	I	An input from the sequencer to request the transfer of data.
SEQ_ACK	O	An output to the sequencer to acknowledge the request and gate data onto the sequencer bus.
PRC_DATA	I/O	A eight-bit wide port for transferring configuration data to or from the microprocessor and CRC residue bytes, ECC remainder bytes, and status bytes to the microprocessor if the XFR_REM_TO_BFR bit is reset. This port also transfers control pointer bytes if the control pointer count is non-zero and the XFR_REM_TO_BFR bit is reset.
PRC_REQ	O	An output for requesting the transfer of CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) via the PRC_DATA port if the XFR_REM_TO_BFR bit is reset, or via the BFR_DATA port if the XFR_REM_TO_BFR bit is set.

Symbol	Type	Description
PRC_ACK	I	An input to acknowledge the request for CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) and to gate them onto the appropriate bus.
PRC_CS/	I	A low-active chip select input used for selecting the IC during configuration writes and reads.
PRC_WE/	I	A low-active write enable input used for writing the configuration registers.
PRC_RD/	I	A low-active read enable input used for reading the configuration registers.
V_{DD}	I	Voltage supply.
V_{SS}	I	Ground.

Pin Configuration



Pin Diagram (Prototypes)



Registers

Configuration Registers

The IC is configured by writing six bytes to the microprocessor port. This establishes the following parameters and modes of operation:

- Listen/Transparent Mode
- Normal/Diagnostic Mode
- Remainder Transfer Mode
- Enable/Disable Bus Parity Checking
- Request/Acknowledge Polarities
- Data Gate Polarity
- Normal/Divide-By-Two Clock
- Number of Data Bytes per Data Field
- Number of Control Bytes per Control Field
- Number of Interleaves
- Number of ECC bytes per Interleave
- Enable/Disable CRC Code
- Enable/Disable ECC Feedback During Redundancy Time

Complete details of the five configuration bytes follow.

Configuration Register 1

Bit

7 RESET: Clears all control and error latches.

6 RESET_ERRORS: Clears all error latches.

5 INITIALIZE: This bit must be set to allow the internal configuration register pointer to point to register 2; otherwise the pointer will remain pointing to byte 1. The state of the configuration register pointer at power-up is indeterminate; writing five consecutive hex 80 values to the microprocessor port will assure that the pointer is pointing at register 1.

4 READ_STATUS: When this bit is set, a microprocessor read sequence will read the two status registers instead of the configuration registers. The first read will input status register 1 and the second, status register 2.

Bit

3 DIAGNOSTIC_MODE: When this bit is set, during writes ECC redundancy bytes input from the buffer will be passed to the sequencer instead of those generated by the IC; during reads the ECC redundancy bytes input from the sequencer will be output instead of the ECC remainder bytes generated by the IC.

2-0 INTERLEAVE_COUNT 2-0:

INTERLEAVE COUNT	NUMBER OF INTERLEAVES
0	3
1	4
2	5
3	6
4	7
5	8
6	9
7	10

Configuration Register 2

Bit

7 LISTEN_MODE: If this bit is set, listen mode is enabled, otherwise transparent mode is enabled.

6 XFR_REM_ON_ERR: If this bit is set, the CRC residue bytes, ECC remainder bytes, and status bytes will be transferred starting when a non-zero CRC residue byte or ECC remainder byte is encountered.

5 XFR_REM_TO_BFR: If this bit is set, the CRC residue bytes, ECC remainder bytes, and the status bytes (and control pointer bytes, if used) will be output from the buffer port instead of the microprocessor port.

4 XFR_REM_WITH_BFR_REQ: If this bit is set, requests to transfer CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) during reads will be directed to the BFR_REQ output instead of the PRC_REQ output. The acknowledge will still be expected on the PRC_ACK input. The XFR_REM_TO_BFR bit must be set and the LISTEN_MODE bit must be reset or else this bit is overridden.

Bit

3 DELAY_SEQ_ACK: If this bit is set and transparent mode is being used, then during redundancy time the deassertion of SEQ_ACK is delayed for one-half to one-and-one-half internal clock periods from the deassertion of SEQ_REQ. Otherwise SEQ_ACK will be deasserted immediately upon the deassertion of SEQ_REQ.

2 BFR_PARITY_CHECK: This bit enables checking of parity of input data on the buffer port.

1 SEQ_PARITY_CHECK: This bit enables checking of parity of input data on the sequencer port.

0 NOT_USED: Must be zero.

Configuration Register 3

Bit

7 BFR_REQ_HI_ACT: Setting this bit defines the active state of the buffer request BFR_REQ as high active.

6 BFR_ACK_HI_ACT: Setting this bit defines the active state of the buffer acknowledge BFR_ACK as high active.

5 SEQ_REQ_HI_ACT: Setting this bit defines the active state of the sequencer request SEQ_REQ as high active.

4 SEQ_ACK_HI-ACT: Setting this bit defines the active state of the sequencer acknowledge SEQ_ACK as high active.

3 PRC_REQ_HI_ACT: Setting this bit defines the active state of the microprocessor request PRC_REQ as high active.

2 PRC_ACK_HI_ACT: Setting this bit defines the active state of the microprocessor acknowledge PRC_ACK as high active.

1 DATA_GATE_HI_ACT: Setting this bit defines the active state of the DATA_GATE input as high active.

0 ENABLE_FEEDBACK: Setting this bit enables feedback during ECC redundancy time of a read.

Configuration Register 4

Bit

7 SYNCHRONOUS_ACK: If this bit is set, the BFR_ACK signal must be synchronous with the input clock. If this bit is reset, BFR_ACK is internally synchronized with the input clock resulting in a one-half to one-and-one-half clock delay in processing the data byte.

6 NO_CRC: If this bit is set, no CRC bytes will be calculated or output by the IC.

5 BURST: If this bit is set, burst transfers to/from the FIFO will be enabled. PRC_REQ or BFR_REQ, as appropriate, will remain asserted as long as the FIFO is not full/empty. Assertion of the respective ACKnowledge will not cause the REQuest to be deasserted.

4-0 CONTROL_COUNT 4-0: This field is the five bits of the control pointer byte count. The value loaded is the length of the control pointer field, zero to thirty-one. The control pointer field is appended to the end of the data field and prior to the CRC/ECC field. It is transferred with the same REQuest-ACKnowledge signals that are selected to transfer the CRC/ECC field.

Configuration Register 5

Bit

7 DIVIDE_BY_2: If this bit is set, the input clock will be divided by two.

6-4 TRANSFER_COUNT 2-0: This field allows less than the maximum number of sixteen ECC redundancy bytes per interleave to be transferred.

TRANSFER COUNT	NUMBER OF ECC REDUNDANCY BYTES TRANSFERRED
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16

Bit

3-0 BYTE_COUNT 11-8: This field is the four high-order bits of the data byte count. The value loaded is the four high-order bits of (N-2) where N is the number of data bytes, not including any control pointer bytes.

Configuration Register 6

Bit

7-0 BYTE_COUNT 7-0: The eight low-order bits of the data byte count. The value loaded is the eight low-order bits of (N-2) where N is the number of data bytes, not including any control pointer bytes.

Status Registers

Following the transfer of the ECC remainder bytes during a read, two bytes of status information are transferred.

Status Register 1

Bit

7-0 ECC_ERROR: One or more of the ECC remainder bytes was non-zero for interleaves 7 to 0 respectively.

Status Register 2

Bit

7 SUMMARY_ERROR: Indicates that at least one other bit of one or both of the status registers is non-zero.

6 SHIFT_REGISTER_ERROR: A hardware error occurred in one of the internal shift registers.

5 TRANSFER_OVERRUN: During a write, an attempt was made to read from the FIFO but it was empty. During a read, an attempt was made to write to the FIFO but it was full.

4 SEQ_PARITY_ERROR: A parity error was detected on data being input on the sequencer port.

3 BFR_PARITY_ERROR: A parity error was detected on data being input on the buffer port.

Bit

2 CRC_ERROR: One or more of the CRC residue bytes was non-zero.

1-0 ECC_ERROR: One or more of the ECC remainder bytes was non-zero for interleaves 9 to 8 respectively.

Functional Description

IC Functional Specifications

Data rate: Maximum 32 megabits per second.

Interfacing: Parallel I/O to buffer, sequencer, and microprocessor.

ECC Code type: Interleaved Reed-Solomon code of degree sixteen.

CRC Code type: Degree four Reed-Solomon code operating on EXCLUSIVE-OR sum of bytes across interleaves.

Number of interleaves: Three to ten.

Number of errors corrected per interleave:

Guaranteed: one-half the number of ECC redundancy bytes used; one to eight. Many error events with more symbol-errors (up to the number of ECC redundancy bytes used) may be corrected if special software algorithms are used.

Sector sizes accommodated:

Up to the number of interleaves used multiplied by 239, minus four if the optional CRC code is used. Arbitrarily large logical sectors may be generated by writing multiple physical sectors back-to-back.

Raw error event rate accommodated:

Using 512-byte, three-interleave sectors, up to 1.E-4 events per bit, depending on burst-length distributions and randomness of error events. Some configurations can accommodate a raw error event rate greater than 1.E-4.

Corrected error rate:

Excluding hardware failures, less than 1.E-13 uncorrectable events per bit for 512-byte, three-interleave sectors and a raw error rate of 1.E-4 events per bit in which each error event affects a single random byte.

Undetected error rate:

Using the ECC code and the CRC code, approximately 1.9E-32 undetected erroneous data events per bit, given the same sector size, raw error rate, and random distribution and excluding hardware failures.

IC Operation

General IC Operation

During writes, the NCR 85C20 IC computes the check bytes (CRC and ECC redundancy bytes) while the data bytes are being written and then appends the bytes at the end the data field. For reads, the check bytes are recomputed and then compared with the check bytes read from the device. The difference (CRC residue and ECC remainder bytes, also called syndromes), if any, is output with the status bytes so that the errors can be located, evaluated, and corrected by the software algorithms.

The NCR 85C20 IC has two modes that it operates in, transparent and listen. For the transparent mode the NCR 85C20 IC is placed between the data buffer and the sequencer and functions similar to a transceiver. The transfer of check bytes and syndromes is isolated from the data buffer bus thereby allowing multiple sectors of data to be placed contiguously in the data buffer.

For the listen mode the NCR 85C20 IC, the data buffer, and the sequencer are all connected to the same bus. The IC monitors the transfer of the data bytes and then outputs the check bytes and syndromes at the appropriate time.

The data field is treated by the NCR 85C20 IC as two sub-fields, system data followed by control pointer data. The system data are stored in the data buffer while the control pointer data, syndromes and status may be stored either in the controller's microprocessor memory or in a different area of the data buffer.

IC Interface

The NCR 85C20 IC's interface is comprised of three byte-wide data ports with associated controls, three input control lines and two output status lines. The three data ports connect to the buffer, sequencer, and microprocessor.

The buffer port receives data from the data buffer (during writes) and outputs data to the data buffer (during reads).

The sequencer port outputs data to the sequencer (during writes) and receives data from the sequencer (during reads). The sequencer port is not used in the listen mode.

The microprocessor port is written to by the microprocessor to configure the IC and may be read to verify the configuration registers or read the status registers. The microprocessor port outputs the CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) during reads if the XFR_REM_TO_BFR mode is not used; if the XFR_REM_TO_BFR mode is used, the buffer port outputs the CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used).

Control Pointer Field

The control pointer field can be zero to thirty-one bytes in length. It is placed immediately following the system data and is protected by the CRC code (if used) and the ECC code. The same port and DMA controls that transfer the syndromes and status bytes also transfer the control pointer field bytes. Note that the data byte count does not include control pointer bytes.

FIFO

An internal twelve-byte FIFO is used to prefetch the control pointer bytes during writes and to buffer the control pointer bytes, syndromes and status bytes during reads.

Since the FIFO is dynamic, the prefetch of the control field is delayed until there are sixty-three or fewer bytes of system data remaining to be written.

Read Mode operation is internally latched until the FIFO is empty even when the read is completed. Therefore the FIFO must be fully emptied or the IC reset prior to the next read or write.

Transparent/Listen Modes

When used in the transparent mode, the NCR 85C20 IC is interposed between the data buffer and the sequencer such that all data to and from the sequencer flows through the IC. During writes, data bytes are input to the IC from the buffer port, output from the IC to the sequencer port and the check bytes are calculated. At the end of the data the check bytes are then output from the IC to the sequencer port. For reads, the data bytes are input to the IC from the sequencer port, output to the buffer port and CRC redundancy bytes and ECC redundancy bytes are calculated. During redundancy time the CRC residue bytes and ECC remainder bytes are formed as the EXCLUSIVE-OR sum of the CRC/ECC redundancy bytes read and CRC/ECC redundancy bytes calculated from the read data. The CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) are directed to either the microprocessor port or the buffer port, depending on the selected XFR_REM_TO_BFR remainder transfer mode.

When used in the listen mode, the data buffer is connected directly to the sequencer, with the NCR 85C20 IC monitoring data as it flows between the buffer and sequencer. During writes, data bytes are input from the buffer port with the CRC redundancy bytes and ECC redundancy bytes being calculated so that during redundancy time they can be output to the buffer port to be appended to the data. During reads, data bytes are similarly monitored and during redundancy time, CRC residue bytes and ECC remainder bytes are formed as the EXCLUSIVE-OR sum of the CRC/ECC redundancy bytes read and CRC/ECC bytes calculated from the read data. The CRC residue bytes, ECC remainder bytes, and status bytes (and control pointer bytes, if used) are directed to either the microprocessor port or the buffer port, depending on the selected XFR_REM_TO_BFR transfer mode.

Diagnostic Mode

During diagnostic mode writes the NCR 85C20 IC is internally placed in a read mode while externally the data flows in the same manner as during normal writes. Thus both the data field and check bytes are fetched from the data buffer and passed to the sequencer. When the check bytes are fetched the IC compares them to the computed values and asserts the ERROR/ output if a miscompare occurs. If the XFR_REM_TO_BFR mode is not selected then the syndromes and status bytes (and control pointer bytes, if used) will also be output to the microprocessor.

For diagnostic mode reads the NCR 85C20 IC allows the check bytes to be stored in the data buffer. During listen mode the only difference is that CER_GATE is not activated. During transparent mode all the bytes are passed through the IC to the data buffer. In all modes except XFR_REM_WITH_BFR_REQ the syndromes and status (and control pointer bytes, if used) are also transferred.

DIAGNOSTIC WRITE

MODE	Input to IC	Output to SEQ	Output to μP
Transp-XFR TO μP	data,ctrl,chk	data,ctrl,chk	ctrl,synd,stat
XFR TO BFR	data,ctrl,chk	data,ctrl,chk	---
XFR W/BFR_REQ	data,ctrl,chk	data,ctrl,chk	---
Listen-XFR TO μP	data,ctrl,chk	---	ctrl,synd,stat
XFR TO BFR	data,ctrl,chk	---	---

DIAGNOSTIC READ

MODE	Input to IC	Output to BFR	Output to μP
Transp-XFR TO μP	data,ctrl,chk	data,ctrl,chk	ctrl,synd,stat
XFR TO BFR	data,ctrl,chk	data,ctrl,chk	ctrl,synd,stat
XFR W/BFR_REQ	data,ctrl,chk	data,ctrl,chk	---
Listen-XFR TO μP	data,ctrl,chk	---	ctrl,synd,stat
XFR TO BFR	data,ctrl,chk	---	ctrl,synd,stat

Companion Software Specifications

Memory Requirements

The companion software used with the NCR 85C20 IC requires from 4K bytes to 10K bytes of memory, depending on the processor employed in the controller and on correction performance and correction time requirements. Implementation of extended correction algorithms may require more memory.

Correction Times

Assembly-language companion software is complete for several processors, including the Z8, 8088, 68HC11, and 68000. Average correction times are given in the table below. These times assume an architecture that allows the processor to run at full speed during correction using memory-mapped remainder and data buffers. If the status bytes output from the IC indicate no errors are present, the companion software need not be called. When errors are detected, the total correction time for a sector consists of the sum of the overhead figure and the correction times for each interleave.

The correction times specified below assume a fixed sector size, a fixed number of interleaves, a single number of ECC redundancy bytes per interleave, and that no processor registers need be saved/restored. Correction times will increase depending on the number of configurations (combinations of sector lengths, numbers of interleaves, and numbers of ECC redundancy bytes per interleave) which the correction algorithm must support.

AVERAGE CORRECTION TIMES (IN MICROSECONDS) PER INTERLEAVE
VS NUMBER OF ERRORS PER INTERLEAVE, OPERATING ON A 512 BYTE,
3 INTERLEAVE SECTOR WITH 16 ECC REDUNDANCY BYTES PER INTERLEAVE

# OF ERRORS	8088 (@8 MHz)[1]	Z8 (@12 MHz)[2]	68000 (@8 MHz)[3]
OVERHEAD[4]	50	25	60
0[5]	6	15	20
1	140	190	120
2	460	540	420
3[6]	1900	3500	3500
4	5300	4500	4250
5	8900	8000	6900
6	11600	10000	9150
7	14500	12500	11400
8	17000	15000	13700

Notes: 1 Measured on 8088-based computer
2 Measured on Z8 development system
3 Measured on 68000-based computer
4 Only those sectors in error incur a time penalty
5 Only those zero-error interleaves before the last interleave in error incur a time penalty
6 8088 time reflects an available software option which decreases three-error correction time

Software Verification

DST has verified its proprietary software algorithms with a combination of high-level language and assembly language test routines. Millions of test cases have been run against the Z8 and 8088 versions of the software. All cases of one to eight random byte-errors in an interleave have been corrected properly, and measured miscorrection probability matches that calculated for cases of nine or more random byte-errors in an interleave. Software for any new processors will undergo the same degree of testing.

Companion Software Operation

The companion software for the NCR 85C20 IC developed by DST implements a proprietary set of fast algorithms. This software normally resides in the instruction memory space of the controller and is executed by its processor. The assembly-language software is complete for several processors, including the Z8, 8088, 608HC11, and 68000. DST licenses this software to NCR 85C20 IC customers.

The traditional steps used to decode a Reed-Solomon code are listed below:

- Compute individual syndromes (or composite remainder)
- Generate error-locator and error-evaluator polynomials
- Find roots of the error-locator polynomial
- Compute error locations as logs of these roots
- Compute error values using the error-evaluator polynomial

Algorithms for performing these steps can be found in most texts on error-correcting codes. However, the most efficient algorithms are generally proprietary. DST has developed a proprietary set of particularly efficient algorithms which accomplish error correction by faster, more direct methods.

Electrical Characteristics

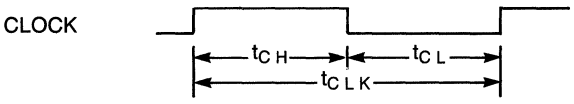
DC Characteristics

Absolute Maximum Stress Ratings				
Storage temperature		−55	+150	°C
Operating temperature		0	+70	°C
Supply voltage		−0.3	+7.0	V
Input voltage		−0.3	+7.0	V
Operating Ratings				
		(Min)	(Max)	
Supply high voltage	V_{DD}	4.75	5.25	V
Input high voltage	V_{IH}	2.0	V_{DD}+0.3	V
Input low voltage	V_{IL}	0.8	V_{SS}−0.3	V
Output high voltage	V_{OH}	V_{SS}+2.4	V_{DD}+0.3	V
Output low voltage	V_{OL}	V_{SS}+0.4	V_{SS}−0.3	V
Output high current ($@V_{OH}$ = 2.4 V)	I_{OH}		−400	μA
Output low current ($@V_{OL}$ = 0.4 V)	I_{OL}	2.0		mA
Input leakage current	I_{IL}	−10.0	+10.0	μA
Tristate leakage current	I_{TL}	−10.0	+10.0	μA
Capacitance of inputs	C_I		10.0	pF
Capacitance of outputs	C_O		10.0	pF
Capacitance of tristates	C_T		10.0	pF
Pullup current	I_{PU}	−600	−50	μA
Supply current	I_{DD}		25	mA

AC Characteristics

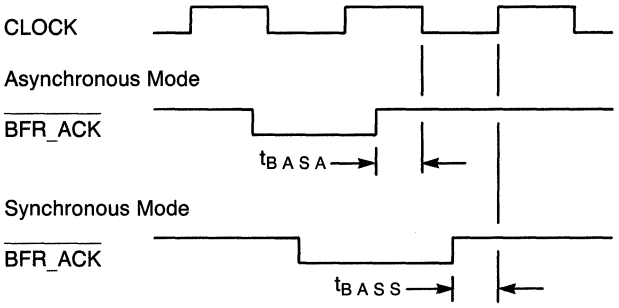
Units of all timings are nanoseconds

CLOCK TIMING



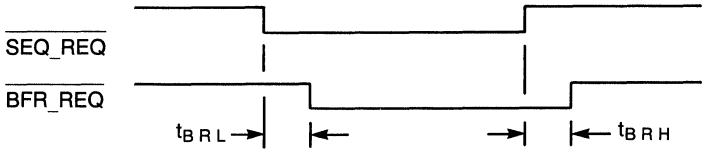
		NOT DIVIDE_BY_2			DIVIDE_BY_2		
Name	Description	Min	Typ	Max	Min	Type	Max
t_{CLK}	Clock Period	62			31		
t_{CH}	Clock High	28			14		
t_{CL}	Clock Low	28			14		

BUFFER ACKNOWLEDGE TIMING



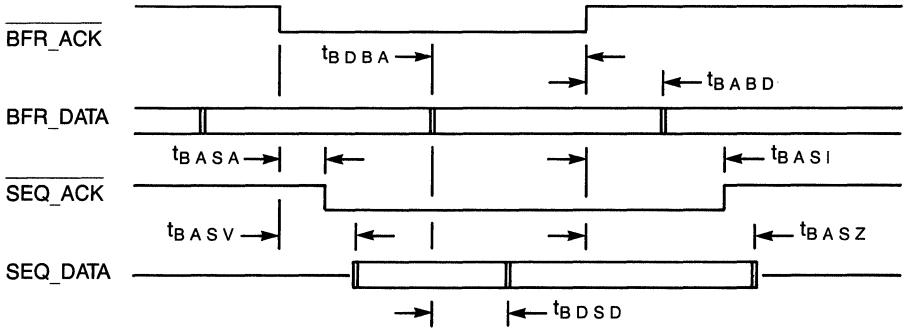
Name	Description	Min	Typ	Max
t_{BASA}	Buffer ACK Setup (Asynchronous)	20		
t_{BASS}	Buffer ACK Setup (Synchronous)	25		

SEQUENCER/BUFFER REQUEST TIMING



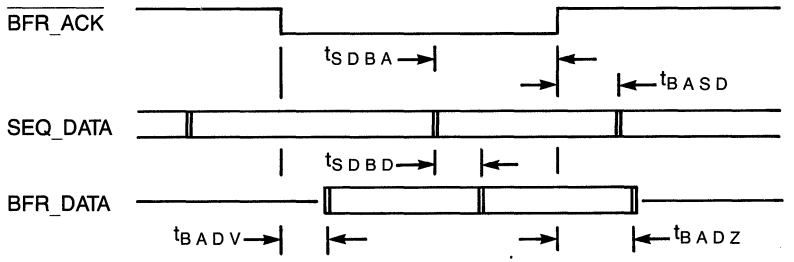
Name	Description	Min	Typ	Max
t_{BRL}	Buffer REQ Low Delay			30
t_{BRH}	Buffer REQ High Delay			30

INPUT FROM BUFFER TIMING (WRITE DATA, LISTEN MODE READ)



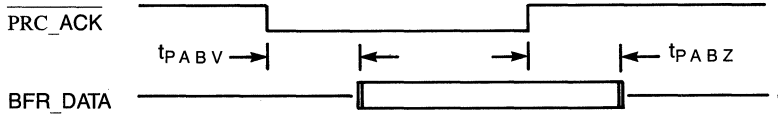
Name	Description	Min	Typ	Max
t_{BDBA}	Buffer Data to Buffer ACK Setup	5		
t_{BABD}	Buffer ACK to Buffer Data Hold	25		
t_{BASA}	Buffer ACK to Seq. ACK Active Delay			30
t_{BASI}	Buffer ACK to Seq. ACK Inactive Delay			30
t_{BASV}	Buffer ACK to Seq. Data Valid Delay			40
t_{BASZ}	Buffer ACK to Seq. Data High-Z Delay	10		50
t_{BDSD}	Buffer Data to Seq. Data Valid Delay			35

OUTPUT TO BUFFER TIMING
(LISTEN MODE WRITE CHECK, TRANSPARENT READ)



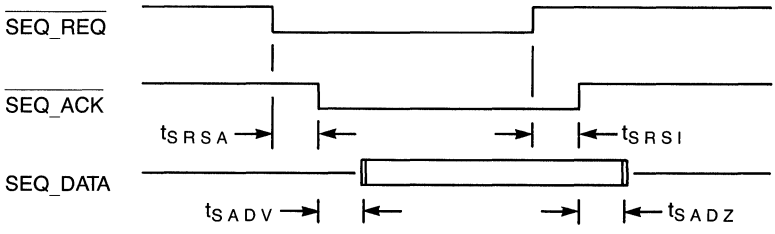
Name	Description	Min	Typ	Max
t_{SDBA}	Sequencer Data to Buffer ACK Setup	5		
t_{SDBD}	Sequencer Data to Buffer Data Delay			35
t_{BASD}	Buffer ACK to Sequencer Data Hold	25		
t_{BADV}	Buffer ACK to Data Valid Delay			60
t_{BADZ}	Buffer ACK to Data High-Z Delay	10		50

OUTPUT TO BUFFER TIMING (READ CHECK WITH XFR_REM_TO_BFR)



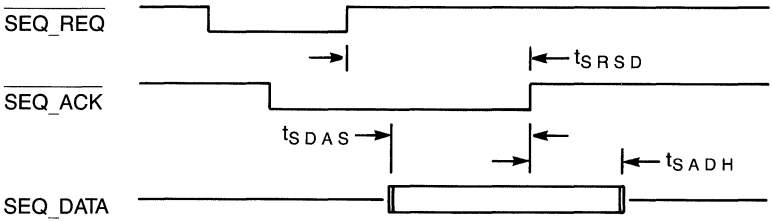
Name	Description	Min	Typ	Max
t_{PABV}	Processor ACK to Buffer DATA Valid			45
t_{PABZ}	Processor ACK to Buffer DATA High-Z	10		50

OUTPUT TO SEQUENCER TIMING (TRANSPARENT WRITE CHECK)



Name	Description	Min	Typ	Max
t_{SRSA}	Seq. REQ to Seq. ACK Active Delay			40
t_{SRSI}	Seq. REQ to Seq. ACK Inactive Delay			40
t_{SADV}	Seq. ACK to Seq. DATA Valid Delay			65
t_{SADZ}	Seq. ACK to Seq. DATA High-Z Delay	10		50

INPUT TO SEQUENCER TIMING
(TRANSPARENT READ CHECK WITH DELAY_SEQ_ACK)

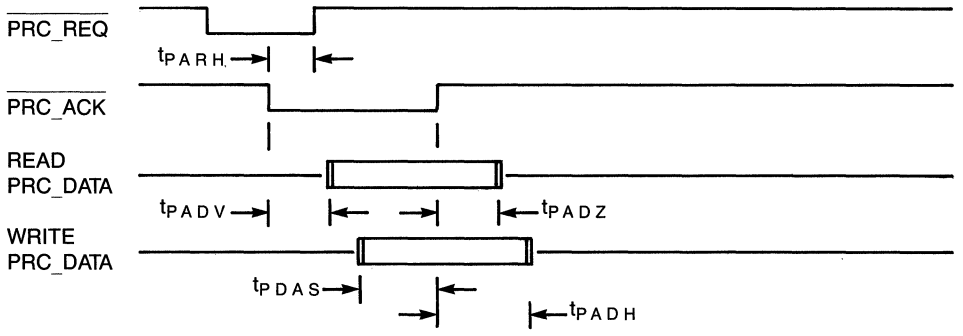


Name	Description	Min	Typ	Max
t_{SRSD}	Seq. REQ to Delayed Seq. ACK Inactive	*		**
t_{SDAS}	Seq. DATA to Seq. ACK Setup	5		
t_{SADH}	Seq. ACK to Seq. DATA Hold	25		

When using DIVIDE_BY_2: * $t_{SRSI} + t_{CLK}$ ** $t_{SRSI} + t_{CLK} * 3$

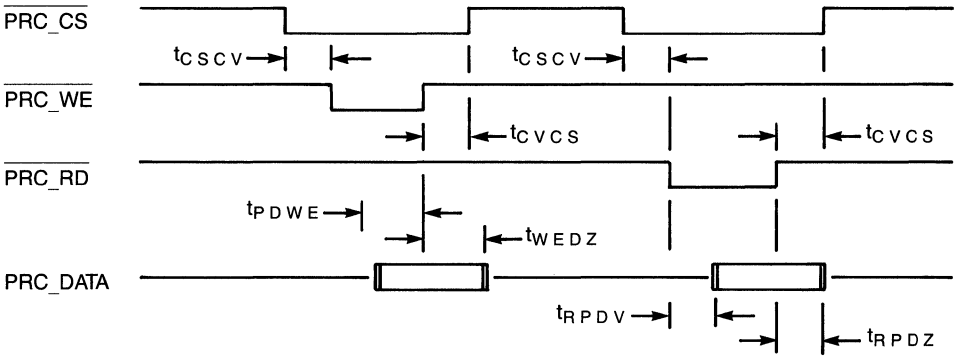
Not using DIVIDE_BY_2: * $t_{SRSI} + t_{CLK} * 0.5$ ** $t_{SRSI} + t_{CLK} * 1.5$

MICROPROCESSOR REQUEST/ACKNOWLEDGE TIMING



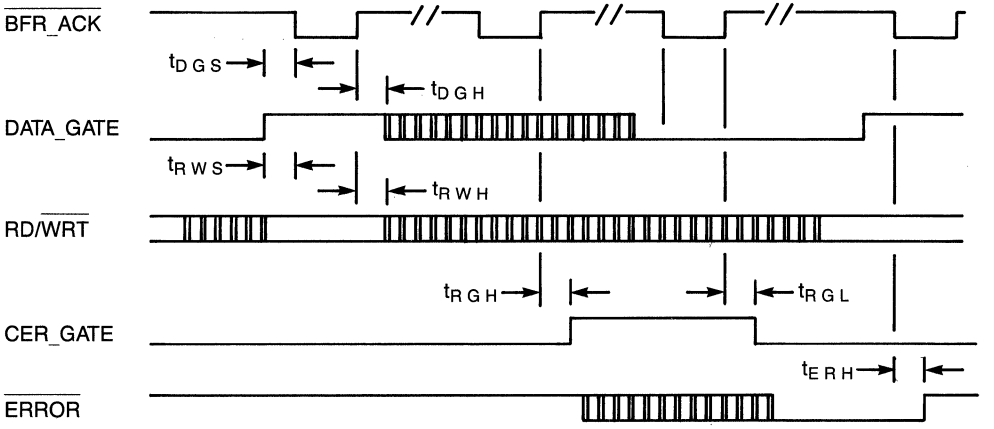
Name	Description	Min	Typ	Max
t_{PARH}	Processor ACK to REQ High Delay			35
t_{PADV}	Processor ACK to Data Valid Delay			65
t_{PADZ}	Processor ACK to Data High-Z Delay	10		50
t_{PDAS}	Processor DATA to ACK Active Setup	5		
t_{PADH}	Processor ACK to Data High-Z Hold	25		

MICROPROCESSOR I/O TIMING



Name	Description	Min	Typ	Max
t_{CSCV}	Chip Select to Command Valid Setup	0		
t_{CVCS}	Command Valid to Chip Select Hold	0		
t_{PDWE}	Processor Data to Write Enable Setup	25		
t_{WEDZ}	Write Enable to Proc. Data High-Z Hold	20		
t_{RPDV}	Read to Processor Data Valid Delay			60
t_{RPDZ}	Read to Processor Data High-Z Delay	10		50

CONTROL SIGNAL TIMING

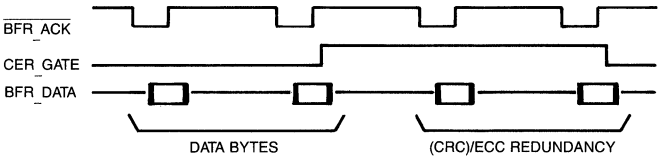


Name	Description	Min	Typ	Max
t_{DGS}	Data Gate Setup	0		
t_{DGH}	Data Gate Hold	15		**
t_{RWS}	Read/Write Setup	0		
t_{RWH}	Read/Write Hold	30		
t_{RGH}	CRC/ECC Redundancy Gate High Delay			60
t_{RGL}	CRC/EDD Redundancy Gate Low Delay			60
t_{ERH}	Error High Delay			85

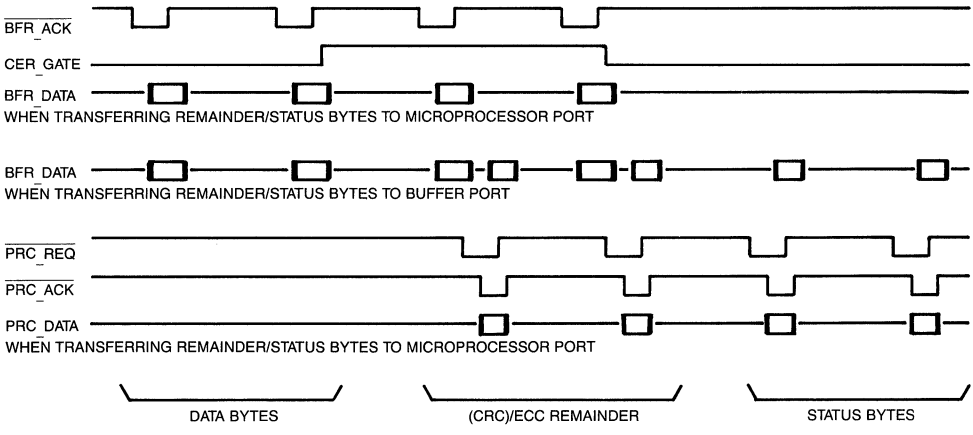
* BFR_ACK for the last check byte of a sector.

** DATA_GATE must be deasserted prior to the assertion of BFR_ACK for the last check byte of a sector.

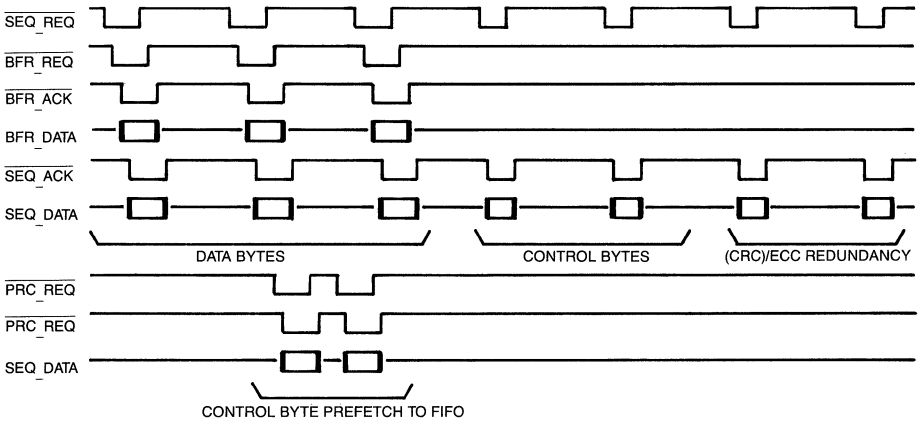
WRITE: LISTEN MODE



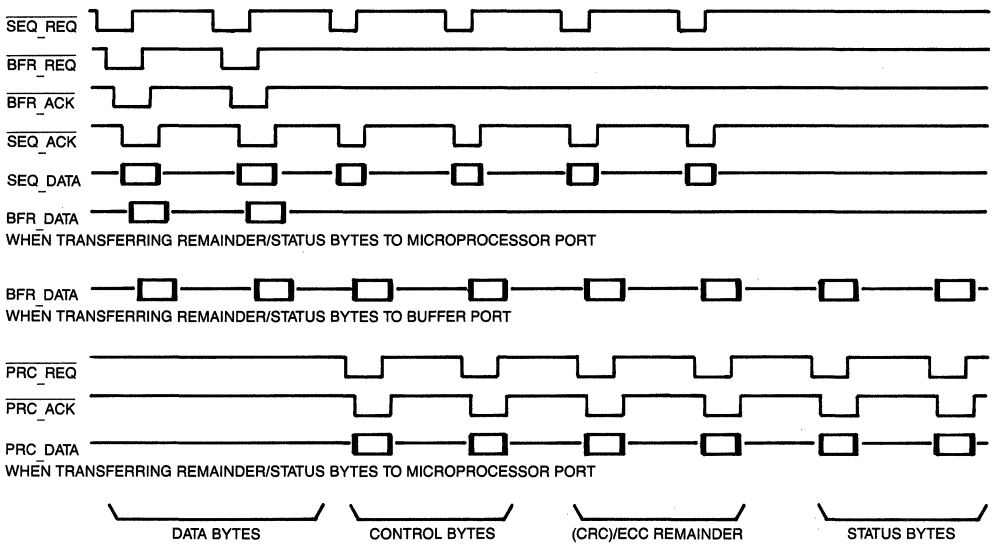
READ: LISTEN MODE



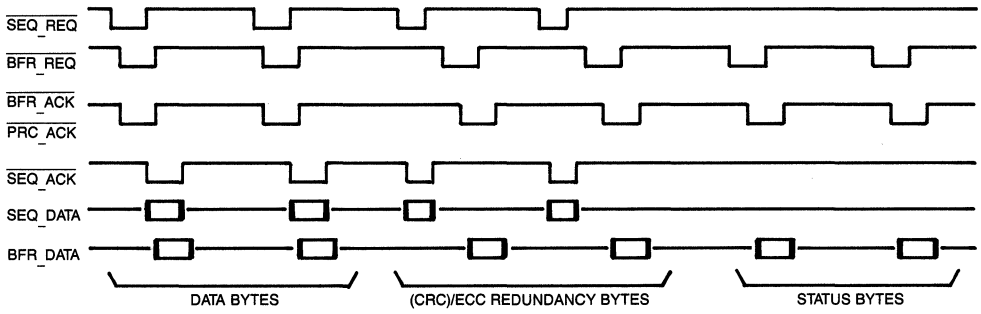
WRITE: TRANSPARENT MODE WITH CONTROL FIELD



READ: TRANSPARENT MODE WITH CONTROL FIELD WHEN TRANSFERRING REMAINDER/STATUS BYTES WITH PRC_REQ



READ: TRANSPARENT MODE WHEN TRANSFERRING REMAINDER/STATUS BYTES WITH BFR_REQ



Applications

This section defines some of the factors to be considered and sets forth strategies that can be used in implementing the NCR 85C20 IC to achieve maximum performance and error tolerance. The discussion was written with 5.25 inch optical WORM disk in mind. However, much of the information is also applicable to high error rate magnetic media devices.

Extending IC Capability

The NCR 85C20 IC can be used to correct higher raw error rates if error pointer information is available from some external source. External pointer sources include modulation-code run-length violations, marginal timing, and marginal amplitude. If signal dropout is the predominant type of error and if the burst length distribution shows a high probability of long defects, modulation-code run-length violations can be an excellent pointer source. When a block-modulation code is used with byte or nibble boundaries, run-length violation pointers will accurately identify bytes in error. When a 2,7-like code is used, a run-length violation pointer may flag a byte adjacent to the byte in error. This error location uncertainty can be overcome to some extent in the decoding algorithms. The capability of the IC can be extended without external pointers by using error locations from adjacent interleaves as pointers. Either method of pointer correction increases software complexity and byte count.

For storage devices, there is a simple method for transferring pointers from the device to the controller. Implement a special read command that places pointer flags on the data line (or bus). These flags replace data, control pointer, CRC and ECC bytes, but not gap or framing bytes - these will be transferred as for a normal read command. When the correction algorithm encounters an uncorrectable sector, it returns to the calling routine with a flag requesting that pointers be read. The calling routine executes the special read command for the required sector and pointer flags are placed on the data line (or bus) and bufferred at the controller. The calling routine returns control to the correction routine, which uses the pointers to assist correction. No special hardware is required at the controller to support this technique. If modulation-code run-length violations are the only pointer source, the only support hardware required at the drive is a multiplexer to switch between data and an invalid-decode line from the modulation decoder.

It is also possible to handle higher raw error rates by reducing the number of bytes per interleave, at the price of increased redundancy. Some applications (e.g., high-density data recorders), will handle long bursts by performing extensive interleaving external to the IC.

Logical sector lengths greater than those allowed by the error-correcting code can be handled by constructing a longer logical sector from multiple contiguous physical sectors.

The NCR 85C20 IC makes it possible to reduce the cost of some existing discrete Reed-Solomon code implementations. It may be necessary to use a PROM (or sequential circuitry) to map from one finite-field representation to another.

Extending Code Capability

The previous heading discussed extending the NCR 85C20 IC's capability without decoding beyond the basic guarantees of the Reed-Solomon code. It is also possible to extend capability by using algorithms that do decode beyond these basic guarantees. Examples of error situations which, though not guaranteed to be handled by extended decoding techniques, do have a certain probability of being handled include:

- A number of random byte-errors in an interleave exceeding the basic guarantees of the error-correcting code.
- A long burst in combination with random byte errors, where the total number of bytes in error in an interleave exceeds the basic guarantees of the error-correcting code.
- A single long burst in an interleave, whose total length approaches the number of ECC redundancy bytes.

When such techniques are used, a tradeoff must be made between correctability and detectability. If, in increasing correctability, detectability is sacrificed to the point that it falls below that required, the NCR 85C20 IC's optional CRC code can be used.

Pointer and adjacent interleave information can be used in conjunction with extended decoding algorithms to further extend correctability and/or detectability and/or to reduce correction time.

Error-Tolerant Track Formats

Achieving error tolerance in the track format is a major consideration when architecting a storage device and controller for high error rate media. All special fields and all special bytes of the track format must be error-tolerant. This includes but is not limited to sync fields, sync marks, header fields, sector marks, and index marks.

Experience shows that designing an error-tolerant track format (one that does not dominate the uncorrectable sector event rate) to support high defect densities can be even more difficult than implementing a high performance error-correcting code over data.

SYNCHRONIZATION

For high defect rate devices, it is essential that the device/ controller architecture include a high degree of tolerance to defects that fall within sync marks. There are several synchronization strategies that achieve this. The selection will be influenced by the nature of the device (e.g., magnetic vs. write-once optical vs. erasable optical) and the nature of defects (e.g., length distribution, growth rate, etc). Both false detection and detection failure probabilities must be considered.

One method for achieving tolerance to defects that fall within sync marks is to employ error-tolerant sync marks. Error-tolerant sync marks have been used in the past that can be detected at the proper time even if several random bits of the mark are in error. Other error-tolerant sync marks have been used that can be detected at the proper time if there are one or more random burst-errors anywhere in the mark. When selecting the pattern for an error-tolerant sync mark, false detection requires as much attention as detection failure. The best results are often achieved when the pattern is selected to be decoded on the device-side of the recording code decoder and when clock phasing and byte synchronization are established simultaneously on detection of the mark. Contact DST [(303) 466-5228] for more information on selecting and implementing error-tolerant sync marks.

Another strategy is to replicate sync marks with some number of bytes between. The number of bytes between replications is determined by the maximum defect length to be accommodated. A different code is used for each replication so that the detected code identifies the true start of data. The number of replications required is selected to achieve a high probability of synchronization for the given rate and nature of defects. Mark lengths, codes, and detection qualification criteria are selected to achieve an acceptable rate of false sync mark detection.

A strategy that is applicable to write-once and read-only optical technology is to architect the device and controller such that no single sync mark is required to be detected. If the bit timing is correct, read is initiated whether or not the sync mark is detected. This is possible only if periods of timing uncertainty between sectors, analogous to those caused by a write splice in magnetic recording, are avoided. Some optical device/ controller architectures allow many consecutive sync marks to be in error without causing a loss of data.

Even if the architecture avoids timing uncertainty between sectors, there may still be some timing uncertainty between the last sector of a track and the first sector of a track. In this case, special care must be taken so that a defect in the first sync mark of a track does not cause a loss of data.

Some track formats currently being proposed for optical media have a sensitivity to the loss of the first sync mark or the first few sync marks of a track. It has been suggested that recovery for this type of error could be accomplished by reading pit patterns into RAM within the controller and searching for recognizable sync marks to establish a reference for decoding pit patterns. Once the reference was established, data would be decoded from the pit patterns with software, the decoded data would be input to the EDAC circuitry, and operation would continue. This technique has been given the name "heroic recovery." Heroic recovery has merit provided that the frequency with which it is invoked is low, in order to minimize performance impact. Some of the other techniques discussed above can be used to reduce the frequency of the need to perform heroic recovery.

One final point on synchronization: if preambles precede sync marks, they must also be error-tolerant, and if there are several steps involved in synchronization, each must be error-tolerant.

MAINTAINING SYNCHRONIZATION THROUGH LARGE DEFECTS

Obviously, it is desirable to maximize the defect length that the PLL can flywheel through without losing synchronization. Engineers responsible for defect handling strategy will want to influence the device's rotational speed stability and PLL flywheeling characteristics. One technique that has been used to extend the length of bursts the PLL can flywheel through is to coast the PLL through defects by using some criteria (run-length violation, loss of signal amplitude, etc.) to temporarily shut off updating of the PLL's frequency and phase memory.

FALSE SYNC MARK DETECTION

The false detection of a sync mark can result in synchronization failure. The probability of false mark detection must be kept low by careful selection of mark lengths, codes, and qualification criteria. In some architectures, once data acquisition has been achieved, sync mark detection is qualified with a timing window in order to minimize the probability of false detection. In such an architecture, it is desirable to generate the timing window from the reference clock; if the timing window is generated from the data clock and the PLL loses sync while clocking over a large defect in a known defective sector, the following good sector may be missed due to the subsequent mispositioning of the timing window.

RESYNC FIELDS

Many storage devices (both optical and magnetic) that support high error rate media use resync fields. There are two distinctly different approaches used for resync: guaranteed synchronization and synchronization recovery.

For guaranteed synchronization, the philosophy is to make synchronization so tolerant of defects that the probability of synchronization loss is less than the device's specified uncorrectable error rate. Frequently when this approach is used, all clocking information is derived from the resync fields and no clocking is derived from data.

When synchronization recovery is used, the philosophy is to regain any synchronization loss at the next resync field. Resync fields are spaced to limit the burst length associated with loss of sync to that which is correctable by the ECC employed. When this approach is used, clocking is generally derived from data. Associated with this method is a specification for the maximum number of slipped cycles (plus or minus) that the resync fields can tolerate. The correct number of bytes must be supplied to the EDAC circuitry whether or not cycles have been slipped.

ERROR-TOLERANT INDEX AND SECTOR MARKS

If index and sector marks are recorded on the media, they must be error-tolerant so that they are detected with very high probability, or the device/controller architecture must be insensitive to their loss. In addition, the probability of false detection of these marks must be kept low.

HEADERS

For high error-rate devices, header strategy is influenced by defect event rates, growth rates, length distributions, performance requirements, and write prerequisites. Consider an optical device with eight-byte header fields, a defect event rate of 1.0E-4 defect events per bit, and twenty-five sectors per track; roughly one of six tracks would contain a defective header. Clearly in this case some form of header defect insensitivity is required.

One header strategy requires replication. A number of contiguous headers with CRC are written, then on read one copy must be read error-free. Another strategy is to allow a data field to be recovered even if its header is in error. This requires that headers consist solely of address information such as track and sector number. If a header is in error, such information can be generated from known track orientation. Some optical devices combine this strategy with header replication in order to minimize the frequency at which address information is generated rather than read. In any case, devices using high error-rate media must be insensitive to defects falling into the headers of several consecutive sectors. When address information is generated rather than read, the data field can be further qualified by subsequent headers.

Many optical devices also include address information within the highly protected data field to use as a final check that the proper data field has been recovered. This check must take place after error correction. The best time to perform it may be just before releasing the sector for transfer to the host.

Some devices avoid headers altogether by including address information only in the protected data field. It is difficult to implement this strategy in architectures where correction is allowed to lag read by several sectors when processing high error content data.

SERVO SYSTEMS

In many devices, the ability to handle large defects is limited by the servo system(s). Engineers responsible for defect handling strategy must understand the limits of the servo system(s) relative to defect tolerance. In particular, any testing of defect handling capabilities should include the servo system(s).

MODULATION CODES

The modulation code selected will affect EDAC performance by influencing noise-generated error rates, the extension of error bursts, the ability to acquire synchronization, the ability to hold synchronization through defects, the ability to generate erasure pointers, and the resolution of erasure pointers.

Buffer Considerations

Storage subsystem architectures that employ high-performance EDAC schemes must minimize the effect of error correction on throughput. With high error rates, it is impractical to lose a revolution each time an error occurs. Therefore, some form of real-time correction is necessary. Before discussing a specific architecture, it is important to understand the difference between computer-data applications and real-time applications.

COMPUTER-DATA APPLICATIONS

For computer-data applications, high throughput is important but synchronous data transfer is not essential. To achieve high through-put, data delay must be short most of the time, but longer data delays due to a longer than normal seek time, rotational latency time, retired sector processing time, or error correction time may occur periodically. Less frequently, even longer data delays are encountered due to a maximum length seek, a maximum rotational latency time, a long retired sector processing time or the correction of a significant error event.

These longer-than-normal data delays are acceptable provided that their frequency of occurrence is such that high through-put is maintained. Generally, the longest delays are associated with the least frequent events.

REAL-TIME APPLICATIONS

Real-time applications are those that require a storage device to accept or deliver data at a constant rate. A seek and rotational latency may precede data transfer, but once data transfer commences it must continue at a constant rate.

It seems inconceivable in a real-time application that the transfer rate of a storage device will precisely match that of the application. Therefore real-time applications can be subdivided into two groups: those where the application is faster and those where the storage device is faster.

APPLICATION FASTER THAN STORAGE DEVICE

If the real-time application is faster than the storage device, the device cannot keep up with the application, so the controller must buffer at least part and perhaps all of the requested data before data transfer begins. Thus, a large buffer is required.

STORAGE DEVICE FASTER THAN APPLICATION

When the storage device is faster than the application, a large buffer is required for speed matching. Since the storage device is faster, the buffer will be filling faster than it is emptying. If the block of data being transferred is sufficiently large, the buffer will overflow and a revolution must be lost for speed matching. The buffer must be large enough to store at least one track of data in order for servicing of the real-time application to continue while the revolution is being lost.

It should be clear from the discussion above that a large buffer is required for all real-time applications. Independently of the real-time application issue, most storage device controllers under development today will incorporate large buffers due to other performance considerations. This use of large buffers to enhance performance is being encouraged in part by low RAM costs.

Since a large buffer is likely to be employed anyway, it can be used to smooth out error correction delays in the case where the storage device is faster than the application. The buffer may need to be larger than one track, so that when a revolution is lost for speed matching there will still be enough data remaining in the buffer to handle a significant error correction delay when buffer filling re-commences. It may also be necessary to provide a capability for delaying the start of data transfer until some amount of data has been bufferred and corrected so that an error correction delay that occurs early in a transfer can be smoothed out as well.

BEST-CASE AND WORST-CASE OPTIMIZED EDAC SYSTEMS

Best-case optimized EDAC systems strive for minimum data delay and fastest correction for those error cases that occur most frequently. Worst-case optimized EDAC systems strive for minimum worst-case correction times. It is not uncommon for worst-case optimized EDAC systems to have a higher average data delay and therefore less through-put than best-case optimized EDAC systems.

Most EDAC implementations employing the NCR 85C20 IC are best-case optimized. When there are no errors, data is delayed one sector time between device and host. When there is a single byte in error, data is delayed one sector time plus the very short time required to correct the single byte in error. Long error correction delays are occasionally encountered, but the longer the delay, the more rare is the error case causing it.

Correction times in systems using Reed-Solomon codes are determined by the implementation alternatives selected and the speed of the processor performing the correction algorithm. The number of errors correctable without losing a revolution depends on correction time, data transfer rate, and buffer strategy. The buffer strategy outlined below can be very effective in minimizing the effects of correction time on through-put. This and other similar buffer strategies are expected to be widely used in the future as the storage-device industry moves to high-performance error correction. Best-case optimized EDAC implementations employing the NCR 85C20 IC are a good match for computer-data applications and for most real-time applications as well, provided that an appropriate buffer architecture such as the one described below is employed.

BUFFER ARCHITECTURE FOR THE NCR 85C20 IC

A sequencer handles the buffering of data bytes, control pointer bytes, CRC residue bytes, ECC remainder bytes, and status bytes for each sector. The sequencer continues transfer from the device when errors occur, and orientation is not lost. A processor checks the status bytes for each sector and performs the correction algorithm when required. After correction, a sector is released for transfer to the host. This architecture requires that the transfer of data from the device to the buffer and from the buffer to the host be asynchronous; the reading of sectors by the sequencer and the correction of sectors by the processor are independent.

If there are no errors, read data will be delayed by one sector-time, as is the case today with most 5.25 inch magnetic disk controllers. When errors occur, data will be additionally delayed by the time required for error correction. However, the proposed architecture minimizes the effect of this additional delay. If transfer to the host is faster than transfer from the device, a multiple-sector transfer may end at the same time regardless of whether the transfer was error-free or there was a significant error event within the transfer. After an error, data transfer to the host will be delayed by more than one sector-time, but since transfer to the host is faster than transfer from the device, it is possible for transfer to the host to catch up before the multiple-sector transfer ends. For single-sector reads, the average correction time per sector is overshadowed by access and latency times.

The buffering of ECC remainder bytes should be memory-mapped to minimize the time required to access them. The ECC remainder bytes for each interleave are not stored consecutively in the remainder buffer but are spaced apart by the number of interleaves, and therefore are not easily or quickly accessed via DMA transfers.

In summary, to minimize the effects of correction time on device performance, use the following guidelines when designing the buffer:

- Use a large data buffer.
- Memory-map the ECC remainder buffer.
- Do not allow ECC errors on read to interrupt device-to-buffer transfers.
- Make buffer loading and unloading asynchronous.
- Have the processor check status and perform correction if necessary before sectors are released to the host.

When the buffer strategy outlined above is used, error occurrences affect performance by introducing data delay. To put this delay into proper perspective, Tables 6.1 to 6.3 are included below.

Table 6.1 shows the rates at which various numbers of byte errors occur in an interleave vs. various raw error event rates. Table 6.2 shows the average contribution to data delay made by each number of errors in an interleave, and the average delay from all errors, vs. various raw error event rates, using the correction times of the 8088 processor shown in Companion Software Specifications on page 230. In generating Tables 6.1 and 6.2, it was assumed that each error event affects a single byte and that error events occur at random intervals in a three-interleave, 512-byte sector.

The 8088 processor correction times from page 230 were also used to compute average correction time per sector against the defect distribution assumed by the X3B11 EDAC Ad Hoc Committee (Hitachi model: M=4, L=2, truncation at 100 bits, and a defect event rate of 1.E-4 events per bit). These results are shown in Table 6-3.

Diagnostic Considerations

The classical "write long", "read long" diagnostic approach for storage device error correction does not apply to high error-rate devices, since a sector read under diagnostic testing may contain real errors along with the simulated errors. In the case of optical WORM disks, the classical approach has the additional disadvantage of using up media. For these reasons, a diagnostic mode has been incorporated into the NCR 85C20 IC.

During a diagnostic write, the buffer supplies data, CRC redundancy bytes (if the optional CRC code is used), and ECC redundancy bytes to be transferred to the sequencer. The IC computes its own CRC bytes and ECC redundancy bytes and compares them to those supplied by the buffer, asserting the ERROR/ signal if any do not match. The data-plus-redundancy passed through the IC may or may not be actually written to a device, depending on the design of the controller.

During a diagnostic read, the CRC residue bytes and ECC redundancy bytes input from the sequencer are transferred to the buffer instead of the internally-generated CRC residue bytes and ECC remainder bytes. The status bytes are generated and transferred as for a normal read. The CRC residue bytes and ECC remainder bytes are still computed and the ERROR signal will be asserted if any are non-zero. A diagnostic read allows the host access to the actual CRC residue bytes and ECC redundancy bytes for any data field.

The diagnostic mode allows the IC to be checked for proper operation without reading from or writing to the device. Tests should include a simulated error-free sector as well as cases that simulate one interleave in error at a time. The diagnostic mode also provides the capability to perform classical "write long" and "read long" operations, if so desired.

To distinguish between NCR 85C20 IC failures and software failures, test the software first: test all tables in software, then test the software itself against sets of ECC remainder bytes for which proper responses are known.

Design Support From DST

Design support is available from Data Systems Technology, Corp., the developer of the technology implemented in the NCR 85C20 IC. DST can provide assistance with system integration, defect distribution analysis, error-tolerant track-format strategies, buffer strategies, and processor selection.

The performance of an EDAC system depends on the nature of defects as well as their rate of occurrence. DST has developed software models for computing uncorrectable error rate from given burst length distributions and has results for several distributions measured on 5.25 inch optical media.

High-performance error correction of data will be to little avail if the effects of a high defect rate overwhelm other aspects of the recording system. DST can provide assistance in selecting track formats, synchronization strategies, and recording codes for high error-rate environments.

The processor selected for the controller will greatly influence correction times. DST has evaluated a number of processors to determine relative merit as candidates for implementing the error-correction algorithms. When incorporating the NCR 85C20 IC into your designs, please consult DST [(303) 466-5228)] regarding processor selection.

For a nominal fee plus expenses, DST will send an engineer to your site to make a presentation and to answer your questions.

TABLE 6.1 PROBABILITIES OF ONE TO EIGHT ERRORS PER INTERLEAVE
IN UNITS OF (INTERLEAVES WITH GIVEN NUMBER OF ERRORS)
PER INTERLEAVE IN A 3-INTERLEAVE, 512-DATA-BYTE SECTOR
GIVEN RANDOM ONE-BYTE-ERROR DISTRIBUTION

Errors Per	Raw Error Event Rate (Random Byte-Errors Per Bit)				
Interleave	5.0E-4	2.0E-4	1.0E-4	1.0E-5	1.0E-6
1	3.5E-1	2.2E-1	1.3E-1	1.5E-2	1.5E-3
2	1.3E-1	3.3E-2	9.6E-3	1.1E-4	1.1E-6
3	3.3E-2	3.3E-3	4.7E-4	5.4E-7	5.5E-10
4	6.1E-3	2.4E-4	1.7E-5	2.0E-9	2.0E-13
5	8.9E-4	1.4E-5	5.1E-7	5.8E-12	5.9E-17
6	1.1E-4	6.9E-7	1.2E-8	1.4E-14	1.4E-20
7	1.1E-5	2.8E-8	2.6E-10	2.9E-17	3.0E-24
8	1.0E-6	1.0E-9	4.6E-12	5.3E-20	5.3E-28

TABLE 6.2 AVERAGE CORRECTION TIMES
IN UNITS OF MICROSECONDS PER INTERLEAVE USING 8088
ON 3-INTERLEAVE, 512-DAT-BYTE SECTORS
GIVEN RANDOM ONE-BYTE-ERROR DISTRIBUTION

Errors Per	Raw Error Event Rate (Random Byte-Errors Per Bit)				
Interleave	5.E-4	2.E-4	1.E-4	1.E-5	1.E-6
1	49	31	18.2	2.1	0.2
2	60	15	4.4	0.5	—
3	63	6	0.9	—	—
4	32	1	0.9	—	—
5	8	—	—	—	—
6	1	—	—	—	—
7-8	—	—	—	—	—
Total	213	54	24.4	2.6	0.2

TABLE 6.3 AVERAGE CORRECTION TIMES
IN UNITS OF MICROSECONDS PER SECTOR USING 8088
GIVEN (M=4, L=2, T=100, DER=1.E-4) ERROR DISTRIBUTION

SECTOR SIZE (BYTES)	NUMBER OF INTERLEAVES	AVERAGE CORRECTION TIME (MICROSECONDS)
512	3	115
512	5	115
1024	5	225
1024	10	230

APPENDIX A. THE CODES IMPLEMENTED

Finite Field Definition

Let β^i represent elements of a finite field generated by the following polynomial over GF (2)

$$x^8 + x^5 + x^3 + x^2 + 1$$

Elements of the finite field employed by the codes implemented in the NCR 85C20 IC are given by

$$a^i = (\beta^i)^{88}$$

Error-Correcting Code Description

The error-correcting code implemented is a long-distance, interleaved, Reed-Solomon code operating on eight-bit symbols. ECC redundancy bytes are inverted (EXCLUSIVE-OR-ed with hex FF) before writing. In a shift register implementation, the shift register is initialized to zero.

ECC GENERATOR POLYNOMIAL

Degree sixteen, distance seventeen, self-reciprocal, with coefficients from GF (256):

$$G(x) = \prod_{i=120}^{135} (x + \alpha^i)$$

COEFFICIENTS OF THE ECC POLYNOMIAL (IN DECIMAL)

1	92	160	86	11	68	2	1	167	1	2	68	11	86	160	92	1

COEFFICIENTS OF THE ECC POLYNOMIAL (IN HEXADECIMAL)

01	5C	A0	56	0B	44	02	01	A7	01	02	44	0B	56	A0	5C	01

LOGS BASE α OF THE ECC POLYNOMIAL COEFFICIENTS (IN DECIMAL)

0	180	20	42	16	179	142	0	11	0	142	179	16	42	20	180	0

LOGS BASE β OF THE ECC POLYNOMIAL COEFFICIENTS (IN DECIMAL)

0	30	230	126	133	197	1	0	203	0	1	197	133	126	230	30	0

The NCR 85C20 IC implements a single Reed-Solomon code of fixed degree, but the number of ECC redundancy bytes used is programmable. High error-rate devices such as optical disks will use all sixteen ECC redundancy bytes to enable correction of up to eight symbol-errors per interleave; lower error-rate devices which do not require this amount of correction power may use fewer ECC redundancy bytes.

The number of ECC redundancy bytes per interleave is programmable from two to sixteen inclusive, in increments of two. The total number of ECC redundancy bytes per sector is the number of ECC redundancy bytes per interleave multiplied by the number of interleaves, which is programmable from three to ten inclusive. Regardless of the number of ECC redundancy bytes per interleave used, the maximum number of data bytes per sector must be less than or equal to the number of interleaves multiplied by 239, minus four if the optional CRC code is used.

When used on 512-data-byte physical sectors with sixteen redundancy bytes per interleave, a minimum of three interleaves is required, with 171 data bytes in two of the interleaves and 170 data bytes in the third interleave, as shown below.

171 DATA BYTES IN INTERLEAVES 0 AND 1	D0	D1	D2	170 DATA BYTES IN INTERLEAVE 2
	D3	D4	---	
	---	---	---	
	---	D508	D509	
	D510	D511	R0	
16 REDUNDANCY BYTES IN EACH INTERLEAVE	R1	R2	---	
	---	---	---	
	---	R44	R45	
	R46	R47		

When the optional CRC code is used, the error-correcting code covers 172 bytes in each of the three interleaves:

172 BYTES COVERED BY ECC IN EACH INTERLEAVE	D0	D1	D2	
	D3	D4	---	
	---	---	---	
	---	D508	D509	
	D510	D511	CRC0	4 CRC BYTES
	CRC1	CRC2	CRC3	
16 REDUNDANCY BYTES IN EACH INTERLEAVE	R0	R1	---	
	---	---	---	
	---	R43	R44	
	R45	R46	R47	

Error-Correcting Code Performance

This appendix defines the basic capability of the Reed-Solomon error-correcting code implemented in the NCR 85C20 IC and its companion software when all sixteen ECC redundancy symbols are used to correct up to eight symbol-errors.

PROBABILITY OF UNCORRECTABLE ERROR

Probabilities for uncorrectable error when the error-correcting code is used to correct up to eight symbol-errors are shown below, for an assumed physical sector size of 512 data bytes. For these calculations, error bursts are assumed to occur at random intervals and each burst is assumed to affect a single symbol (one byte). The raw burst error rate is the ratio of burst error events to total bits transferred. The uncorrectable error rate is the ratio of uncorrectable error events to total bits transferred.

RAW BURST ERROR RATE DEFECT EVENTS/BIT	UNCORRECTABLE ERROR RATE (UCE) UNCORRECTABLE INTERLEAVE EVENTS/BIT
1.0E-3	1.5E-8
5.0E-4	5.6E-11
2.0E-4	2.2E-14
1.0E-4	4.7E-17
1.0E-5	5.4E-26
1.0E-6	5.4E-35

The formula used for these calculations is given below:

$$UCE = \frac{\text{UNCORRECTABLE INTERLEAVE EVENTS}}{\text{BIT}} = \frac{1}{n*k} \sum_{i>e}^{n} i \binom{n}{i} p^i (1-p)^{n-i}$$

Where

- n = Interleave length in symbols
- k = Symbol width in bits (k = 8 for byte symbols)
- e = Maximum number of symbol errors correctable per interleave (i.e., the number of ECC redundancy symbols divided by two)
- p = Raw symbol error probability (in units of symbol errors per symbol)

$$\binom{n}{r} = \frac{n!}{r!\,(n-r)!} = \prod_{j=0}^{r-1} \frac{(n-j)}{(r-j)}$$

Note that for our assumption of small bursts,

- p = k*RAW BURST ERROR RATE, where units of RAW BURST ERROR RATE is burst error events per bit

If error bursts cluster or if error bursts span more than one symbol, the actual uncorrectable error rate will be greater than that shown above. Figures A.1 through A.3 show uncorrectable error rate as a function of raw burst error rate for several simple cases.

To determine the performance of the error-correcting code for a particular application, the burst length distribution must be known. Initial defects that are handled by some form of media retirement such as defect skipping or alternate sector assignment would not be included in this burst length distribution.

MISCORRECTION PROBABILITY

When the number of errors in an interleave exceeds the correction power of a code, it is possible for the ECC remainder bytes produced to be identical to those produced by some corrrectable set of errors. This is called miscorrection, since the correction algorithm will leave the real errors in place and "correct" the apparent set of errors. Using DST's error-correction algorithm, the miscorrection probability of the error-correction code is given by:

$$P_{mc} = \frac{\text{\# OF VALID REMAINDER BYTES}}{\text{\# OF POSSIBLE REMAINDER BYTES}} = \sum_{i=0}^{e} \frac{n \; i \; 255^i}{256^{(e+i)}}$$

The longest interleave of a 512 data-byte sector is 187 bytes in length (171 data bytes and 16 ECC redundancy bytes). When the error-correcting code is used to correct up to eight symbol-errors the miscorrection probability is

$$P_{mc} = \sum_{i=0}^{8} \frac{187 \; i \; 255^i}{256^{(8+i)}} = 1.7E\text{-}6$$

in units of miscorrected interleaves per uncorrectable interleave.

PROBABILTY OF UNDETECTED ERRONEOUS DATA

As noted above, only those error events which cause the number of symbol-errors in any single interleave to exceed e are subject to miscorrection. Thus the probability of transferring an undetected erroneous interleave is the product of the probability of having more than e symbol-errors in one interleave and the miscorrection probability of the error-correcting code when it is used to correct up to e errors in one interleave. The probability of occurrence of an undetected erroneous sector is the same as that of an undetected erroneous interleave on an events per bit basis, so the probability of transferring an undetected erroneous sector is the product of the probability of occurrence of an uncorrectable interleave and the miscorrection probability of the error-correcting code:

$$P_{ued} = UCE * P_{mc}$$

Using values from above, the probability of transferring undetected erroneous data under a random one-byte error distribution with a raw burst error rate of 1.E-4 per bit for 512 data-byte, three-interleave sectors using all sixteen ECC redundancy bytes is

$$P_{ued} = 4.7E\text{-}17 * 1.7E\text{-}6 = 8.0E\text{-}23$$

in units of undetected erroneous sectors per bit, excluding hardware failure and track-format-induced errors.

When the optional CRC code is used, the probability of transferring undetected erroneous data is the product of the probability of occurrence of an uncorrectable interleave, the miscorrection probability of the error-correcting code, and the statistical misdetection probability of the CRC code:

$$P_{ued} = UCE * P_{mc} * P_{md}$$

Using the misdetection probability of the CRC code from below, the probability of transferring undetected erroneous data under a random one-byte error distribution with a raw burst error rate of 1.E-4 per bit for 512 data-byte, three-interleave sectors using all sixteen ECC redundancy bytes and the optional CRC code is

$$P_{ued} = 4.7E\text{-}17 * 1.7E\text{-}6 * 2.3E\text{-}10 = 1.8E\text{-}32$$

in units of undetected erroneous sectors per bit, excluding hardware failure and track-format-induced errors.

CRC Code Description

The optional CRC code implemented is a Reed-Solomon code operating on one-byte symbols which are the EXCLUSIVE-OR sum of data bytes across interleaves. There are four CRC redundancy bytes per sector, regardless of the number of interleaves. CRC redundancy bytes are not inverted, are written immediately after the data bytes, and are covered by the error-correcting code. In a shift register implementation, the shift register is initialized to zero.

The CRC code is specially constructed so that its residue can be adjusted as correction occurs; when correction is complete, the residue should have been adjusted to zero.

CRC GENERATOR POLYNOMIAL

Degree four, distance five, with coefficients from GF(256);

$$G(x) = \prod_{i=136}^{139} (x + \alpha^i)$$

COEFFICIENTS OF THE CRC POLYNOMIAL (IN DECIMAL, HIGH ORDER FIRST)

1 232 194 86 198

COEFFICIENTS OF THE CRC POLYNOMIAL (IN HEXADECIMAL)

01 E8 C2 23 C6

LOGS BASE ALPHA OF THE CRC POLYNOMIAL COEFFICIENTS (IN DECIMAL)

0 97 228 117 40

LOGS BASE BETA OF THE CRC POLYNOMIAL COEFFICIENTS (IN DECIMAL)

0 121 174 96 205

CRC Code Performance

The statistical misdetection probability of the CRC code is given by

$P_{md} = 2^{-32} = 2.3E\text{-}10$

in units of undetected uncorrectable sectors per uncorrectable sector.

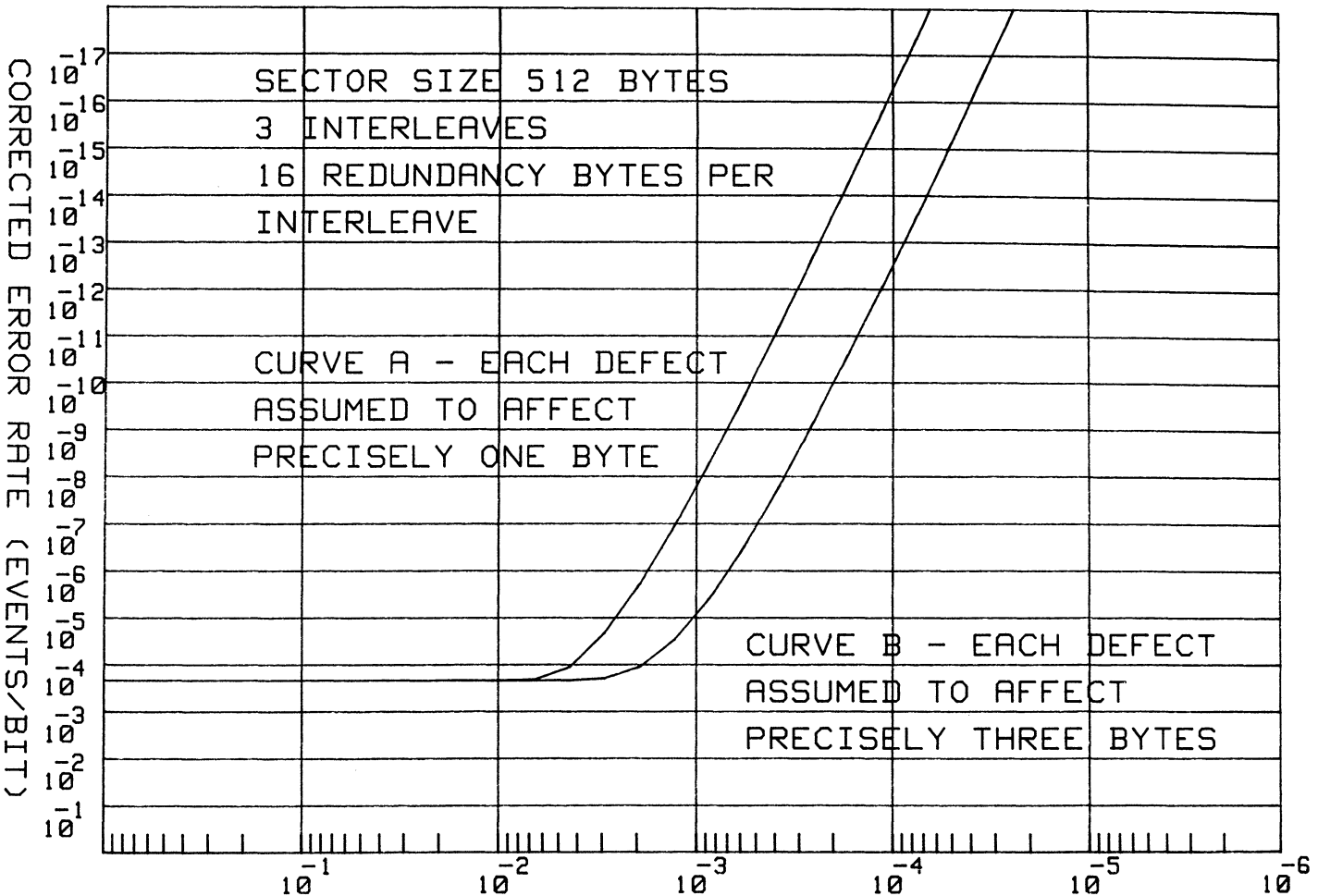


FIG. A.1 RAW BURST ERROR RATE (EVENTS/BIT)

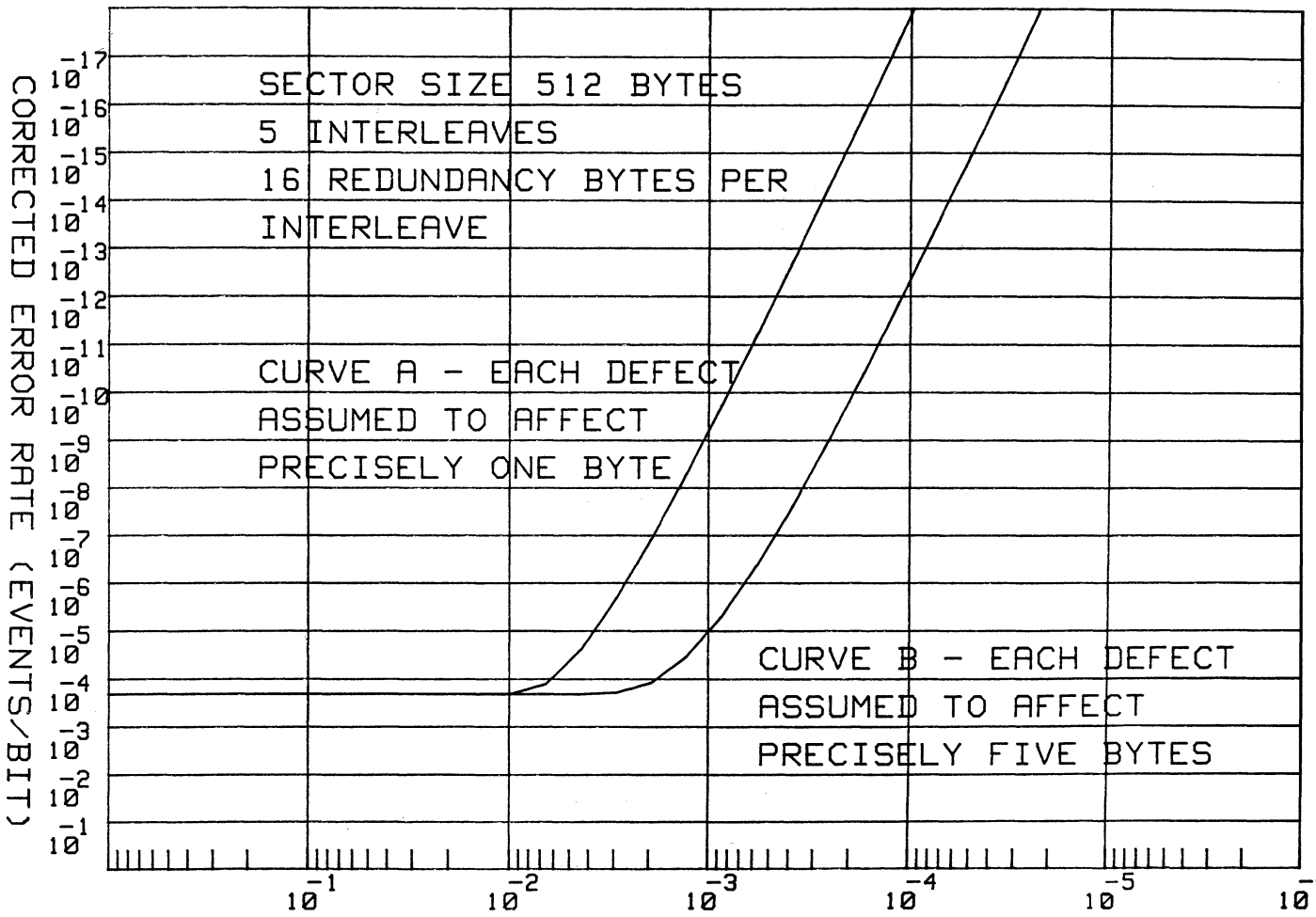


FIG. A.2 RAW BURST ERROR RATE (EVENTS/BIT)

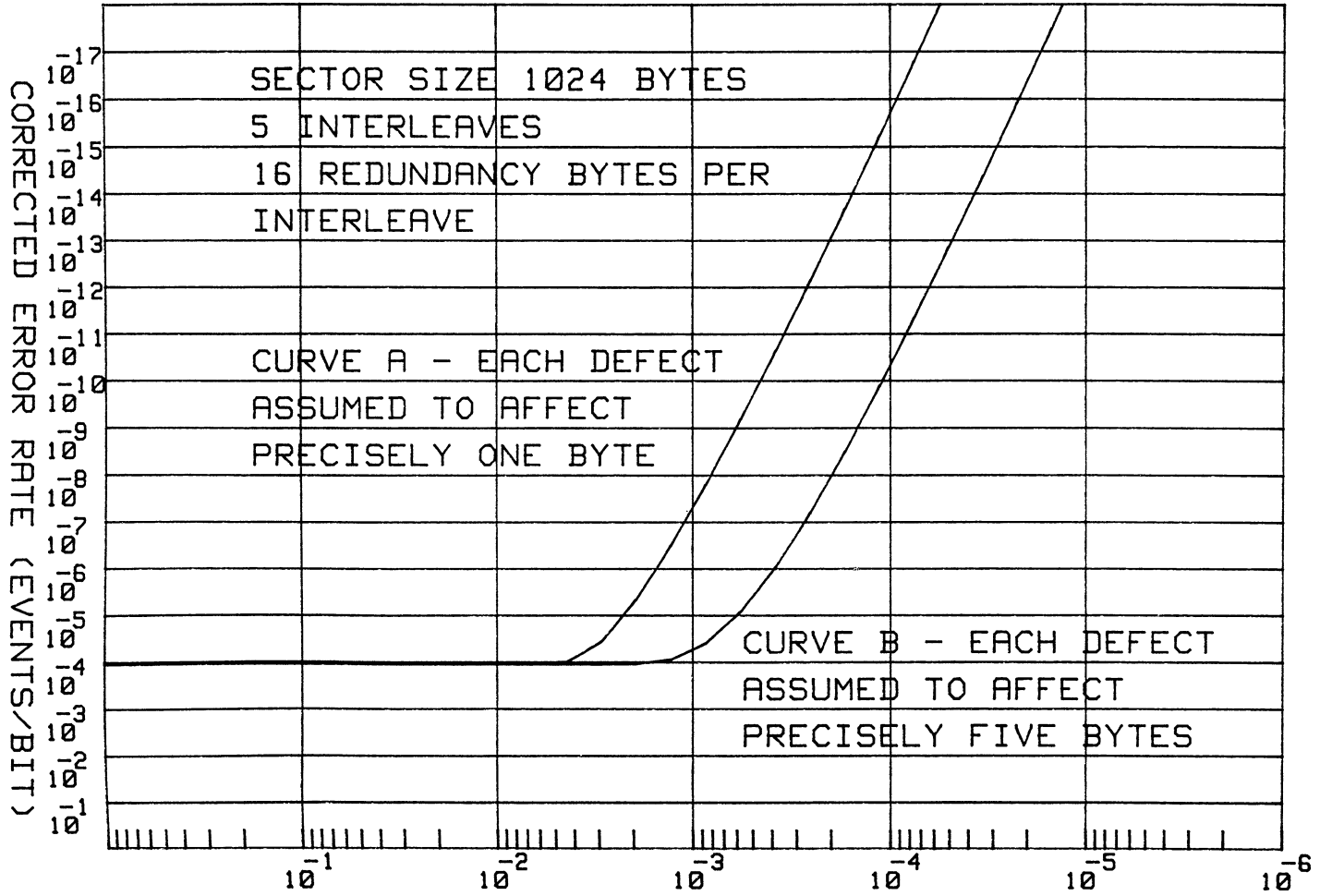


FIG. A.3 RAW BURST ERROR RATE (EVENTS/BIT)

REFERENCES

Neal Glover
Practical Error Correction Design for Engineers, 1982, Data Systems Technology, Corp., Broomfield, CO. (303) 466-5228

Neal Glover
"An overview of Error Detection and Correction Technology for Storage Devices," 1986, Data Systems Technology, Corp., Broomfield, CO. (303) 466-5228

Neal Glover and Trent Dudley
Recording Codes and Error Tolerance for Magnetic and Optical Storage, 1987, Data Systems Technology, Corp., Broomfield, CO. (303) 466-5228

X3B11 86-064
"Proposed EDAC Scheme for 5.25 Inch Optical Storage Devices"

X3B11 86-153
"Encoding Specification for the X3B11-Approved EDAC Code for 5.25 Inch Optical Storage Devices"

GRAPHICS PRODUCTS

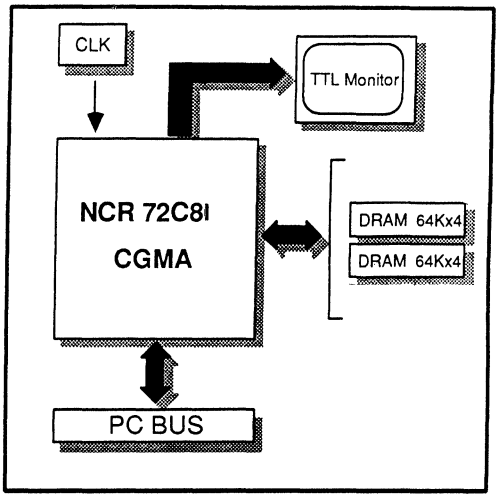
72C81 CGMA

72C81 CGMA

Introduction

Features:

- 100% IBM CGA Compatible
- 100% IBM MDA Compatible
- 100% Hercules Compatible
- Hi-Definition CGA mode
- On chip 6845 CRTC
- On chip character ROMS
- Jumper and Software Mode Selection
- Flashless Screen Updates
- Direct Interfacing
- 84 pin PLCC



System Environment

Summary

The NCR 72C81 CGMA (Color Graphics and Monochrome Adapter) is a single chip CMOS device which provides hardware compatibility with CGA (Color Graphics Adapter), MDA (Monochrome Display Adapter) and HGA (Hercules Graphics Adapter) display controllers. In addition, a high resolution CGA Mode (Hi-Def) provides a 640 × 400 display featuring high quality text and line doubled graphics. Special mapper circuitry intercepts I/O accesses to the internal 6845 and inserts parameters that are correct for high resolution operation. In this way, a high resolution display is produced and 100% compatibility with CGA software is retained.

The NCR 72C81 integrates a 6845 CRT Controller, character ROMs, plus PC bus, monitor and frame buffer interface circuitry. To produce a display adapter, all that is needed is the CGMA, two DRAMs and a clock signal. The display buffer can be written at any time (without producing distortions or snow on the screen) while wait states are minimized.

Operation Overview

The 72C81 CGMA is a single chip CMOS VLSI CRT Controller that supports standard IBM display modes plus a 100% CGA compatible Hi-Def Mode.

In color alphanumeric (A/N) modes, two bytes are used to define character images. A character byte is used to look up character images stored in ROM. The ROM data is serialized and shifted out to the color encoder. Attribute bytes control the encoder and define the color of the character, the background and the border color if CGA mode is active. The attribute byte also controls character blinking, intensity and reverse video display.

In color graphics modes, the display memory is organized in a packed pixel format with each byte representing four or eight pixels, depending on the mode selected. In 320 × 200 CGA graphics mode, each byte represents four pixels with two bits per pixel. Data read from the display memory is shifted as bit pairs to the color encoder. The color of the displayed images is specified through the Color-Select Register (3D9h).

Monochrome A/N modes operate in a similar manner to color modes except that the 8 × 14 pixel cell

character ROM is selected and an additional horizontal pixel is added for each character, resulting in a 720 pixel wide display (80 characters). Additionally, the color encoder outputs video data on the Green output and intensity control on the Intensity output with the available attributes being blink, blank, underline, reverse video and intensify. Vertical sync is inverted in monochrome modes.

In monochrome graphics mode, data from the display memory is serialized and displayed in a 720 × 348 format.

In order to set up a specific display, video is blanked by writing to the Mode Control Register (3D8h), the appropriate timings are loaded to the 6845 CRTC, the display memory is loaded with data and video is enabled. This process is performed through BIOS or through application programs which write directly to the display adapter.

Modes of Operation

Selecting Modes

The display adapter mode (CGA, MDA, Hi-Def or disable) can be selected through the M1 and M2 inputs during power-up reset or by software control. The mode is read via bits 6 and 7 of the Extended Control Register (address 25Ah/35Ah read). M1 and M2 are decoded to select the modes as shown in Table 2.2.

M1	M2	Mode
0	0	CGA
0	1	Hi-Def
1	0	MDA/HGA
1	1	Disable

Table 2.2

The M1/M2 Control Register (address 25Bh/35Bh write only) is used for software selection of display modes. Bit 6 is M1, and bit 7 is M2. The other bits in this register are reserved and should be written as zeros for compatibility with possible future versions. The procedure for changing modes through software is:

- Set M1 and M2 bits of the M1/M2 Control Register for the desired mode.
- Write the Extended Control Register (25Ah/35Ah) with the desired control parameters, plus bit 7 high to reset the 72C81. A software reset does not affect the values stored in the M1/M2 Control Register.
- Re-write the Extended Control Register with bit 7 low to remove the reset condition.
- Reload the 6845 or other parameters as desired.

Mode	Appearance and Display Data	Alpha Format	Charac. Cell	Screen Resolution	Buffer Start	Colors	Page Size
CGA	Color-Alpha	40 x 25	8 x 8	320 x 200	B8000	16	2048
CGA	Color-Alpha	80 x 25	8 x 8	640 x 200	B8000	16	2048
Hi-Def	Color-Alpha	40 x 25	8 x 16	320 x 400	B8000	16	2048
Hi-Def	Color-Alpha	80 x 25	8 x 16	640 x 400	B8000	16	2048
MDA	Mono-Alpha	80 x 25	9 x 14	720 x 350	B0000	3	4096
CGA	Color-Graphic	40 x 25	8 x 8	320 x 200	B8000	4	16384
CGA	Color-Graphic	80 x 25	8 x 8	640 x 200	B8000	2	16384
Hi-Def	Color-Graphic	40 x 25	8 x 16	320 x 400 *	B8000	4	16384
Hi-Def	Color-Graphic	80 x 25	8 x 16	640 x 400 *	B8000	2	16384
HGA	Mono-Graphic	80 x 25	9 x 14	720 x 348	B0000/ B8000	2	32768**

* Scan Doubled Display ** 2 page support in Hercules and Text Mode

Table 2.1 CGMA Operating Modes

Hardware mode selection is accomplished as follows. When RESET (pin 53) is active (high level), the logic levels on pins 13 (M1) and 12 (M2) are read and stored in the M1/M2 Control Register. When RESET is not active, M1 and M2 drive out the values stored in the M1/M2 Control Register, enabling external decode for clock selection for multi-mode applications. Resistors (minimum value 680 ohms) are used to place levels on M1 and M2. M1 and M2 should not be connected directly to the power supply if software control is to be used.

CGMA Disable Mode

This mode tristates the R, G, B, I, H, and V outputs and also prevents the 72C81 from responding to monochrome or color I/O and memory addresses. The purpose of this mode is to disable 72C81 operation if another graphics source is installed in the system.

Monochrome Alpha Mode

The monochrome mode of operation is an 80 × 25 character text mode using 7 × 9 characters in a 9 × 14 character cell. The display memory, starting at address B0000h, is organized as a linear array with two bytes per character. Four pages of text are possible. The first (even) byte is the character code (same as color alphanumeric modes) while the second (odd) byte is the character attribute byte. There are 256 character codes available for the character set. Display memory wraps at address B4000h such that a memory access to address B4000h retrieves the character stored at B0000h.

The attribute byte selects normal video, reverse video, intensify, underline, and blink as shown below. Multiple attributes may be selected.

Monochrome Graphics Mode

Monochrome Graphics Mode is a 720 × 348 pixel display and is selected by setting the mode switches to MONO and controlling bits one and four of the Mode Control Register (3B8h). The CRT controller registers (3B0h-3B7h) must be set for 90 active bytes (45 active characters) per scan line with four lines per row and appropriate blanking. Horizontal and Vertical Sync timing must be set for an 18.4 KHz monitor. Pixel bits are the same as in color modes with 1 bit per pixel at 720 × 348 resolution. Memory organization is set up as one or two pages with page 1 starting at address B0000h and page 2 starting at address B8000h. Each page is divided into four segments as shown below.

Scan Line	Segment Offset
0, 4, 8, ...	x000
1, 5, 9, ...	x200
2, 6, 10...	x400
3, 7, 11...	x600

The Memory Mode Register (3BFh) is used to set the wrap-around and display areas.

Color Alphanumeric Modes

The alphanumeric color modes of operation support both a 40 × 25 and an 80 × 25 character display using the 8 × 8 character cell (normal mode) or the 8 × 16 character cell (Hi-Def mode). The two character sets are ROM based and are integrated within the 72C81 CGMA. The 8 × 8 cell size character set has a 5 × 7 font with a one line descender. The 8 × 16 cell character set has 7 × 9 characters with a two line descender. To produce the 40 × 25 mode display, the 14.318 MHz dot clock is internally divided by two. For Hi-Def mode, a 20 MHz dot clock is used and twice as many lines are displayed.

In the alphanumeric modes, the 16K word display memory is arranged as a linear array with two bytes per character. This provides eight screens of 40 × 25 text and four screens of 80 × 25 text. The first (even)

Attribute Expressed	B7	B6	B5	B4	B3	B2	B1	B0
No Display	x	0	0	0	x	0	0	0
Blink	1	x	x	x	x	x	x	x
Intensify	x	x	x	x	1	x	x	x
Reverse Video	x	1	1	1	x	0	0	0
Underline	x	x	x	x	x	0	0	1
Normal Video	All other combinations							

x = Don't care

byte is the character code and the second (odd) byte is the character attribute. There are 256 character codes for each character set. Display memory from B8000h to BFFFFh can be written to but only B8000h to BBFFFh is displayed. This allows a non-blanking scroll to be performed as in some PCs. To achieve a fully IBM compatible memory wrap, disconnect PC bus address line A14 from the CGMA and connect the CGMA's A14 input to ground.

The attribute byte selects the color of the character, the background color within the cell, and blinking. Each character cell can have one of 16 foreground colors and one of 16 background colors. If bit 5 of the Mode Control Register is set, background color selection is reduced to one of 8 colors and bit 7 of the attribute byte is used for character foreground blink instead of background intensity.

Colors (I,R,G,B)

Colors displayed are produced from a combination of I, R, G and B as shown.

I	R	G	B	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	Light Gray
1	0	0	0	Dark Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	White

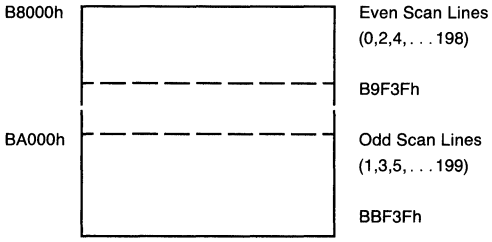
Color Mapping for Alphanumeric Mode

Bit	B7	B6	B5	B4	B3	B2	B1	B0
Color	I/B	R	G	B	I	R	G	B
Function	Blink	Background			Foreground			

Color Graphics Modes

The graphics modes for the 72C81 support a 320 × 200 and a 640 × 200 pixel display. The Hi-Def mode results in each line being displayed twice (double dot), which provides for better resolution and provides a fuller, more solid appearing display.

In graphic modes, the display buffer is split into two 8K byte blocks. Even scan lines (0, 2, 4, ... 198) use the first half of memory and odd scan lines (1, 3, 5, ... 199) use the bottom half of memory. Address B8000h is in the top left of the display while address BA000h is in the top left corner on the next line down. This is illustrated below.



In the 640 × 200 mode, each byte in the display buffer contains the information for 8 pixels (one bit per pixel). The background color is always black and the foreground color is one of eight colors determined by the Color Select Register (3D9h).

In the 320 × 200 mode, each byte in the display buffer contains information for four pixels (two bits per pixel). The background color is one of sixteen colors specified via the Color Select Register (3D9h). The foreground color of the pixel depends on palette selection (Bit 4 and Bit 5 of 3D9h) and the value of the two bits for the pixel.

Pixel Bits		Palette 0 Color	Palette 1 Color
0	0	Background Color	Background Color
0	1	Green	Cyan
1	0	Red	Magenta
1	1	Yellow	Light Grey

An alternate color palette is available in 320 × 200 graphics mode by setting bit 2 of the Mode Control Register (address 3D8h) to a 1 value. This bit is used to inhibit color burst output (to select monochrome output) in systems which have composite video output. If an RGBI monitor is connected, the output is in color and the following palette is available:

Pixel Bits		Alternate Palette	Intensified Palette
1	0	Background Color	Background Color
0	1	Cyan	Light Cyan
1	0	Red	Light Red
1	1	Light Gray	White

When operating in this mode, bit 5 of the Color Select Register (3D9h) does not switch palettes. The intensified version of the palette is selected through bit 4 of the Color Select Register.

Device Communications

The 72C81 responds to both Memory and I/O accesses. I/O accesses (IOR/ and IOW/) are used to specify the manner in which data in the frame buffer is displayed. Memory operations (MEMW/ and MEMR/) are used to write and retrieve data in the frame buffer. I/O operations take place through a group of registers. These registers are described below.

Color Modes

Address (Hex)	Register
3D4	6845 Index Register
3D5	6845 Data Register
3D8	Mode Control Register
3D9	Color Select Register
3DA	CRT Status Register
3DB	Clear Light Pen Latch
3DC	Preset Light Pen Latch

Monochrome Modes

Address (Hex)	Register
3B4	6845 Index Register
3B5	6845 Data Register
3B8	Mode Control Register
3B9	Preset Light Pen
3BA	CRT Status Register
3BB	Clear Light Pen
3BF	Memory Mode Register

Special Registers

Address (Hex)	Register
25A/35A	Extended Control Register
25B/35B	M1/M2 Control Register

The addresses for the special registers is controlled by the 35Xh/25Xh input. Two addresses are available in the event of an I/O address conflict with these special registers.

Selection of modes through the M1/M2 Control Register changes the mapping of the I/O Registers depending on whether the selected mode is color or monochrome.

Registers

Mode Control Register - I/O address 3D8h (color), 3B8h (mono) (Write only)

Bit 0 — 0 = All other modes
1 = 80 × 25 alphanumeric mode

Bit 1 — 0 = Alphanumeric mode
1 = Graphics Mode

Bit 2 — 0 = Normal operation
1 = Enable alternate palette (320 × 200 graphics)

Bit 3 — 0 = Blank display
1 = Enable video

Bit 4 — 0 = Other modes
1 = 640 × 200 graphics

Bit 5 — 0 = Use bit 7 of the alphanumeric attribute byte for an intensified set of background colors.
1 = Blink alphanumeric foreground if bit 7 of the attribute byte is set.

Bit 6 — 0 = Not Used

Bit 7 — 0 = Normal
1 = Display page 2 of monochrome graphics if memory mode register indicates that two pages are available.

Color Select Register - I/O address 3D9h (Write only)

Bits 0-3 — Sets the border color in alphanumeric modes.
Sets the background color in 320 × 200 mode.
Sets the foreground color in 640 × 200 mode.

Color Components:		
	Bit 0	Blue
	Bit 1	Green
	Bit 2	Red
	Bit 3	Intensity

Bit 4 — 1 = Intensified background colors in alphanumeric mode and intensified foreground colors when in 320 × 200 graphics modes.

Bit 5 — 0 = Palette 0
1 = Palette 1 (320 × 200 mode)

Bit 6 — Not used

Bit 7 — Not used

Status Register - I/O address 3DAh (color), 3BAh (mono) (Read only)

Bit 0 — (Active High) - Blanking Interval: Indicates that the display is in a blanking interval.

Bit 1 — (Active High) - Light Pen Signal: Indicates that a light pen signal has occurred. This bit is reset by writing to I/O address 3DBh.

Bit 2 — (Active Low) - Light Pen Switch: A low value indicates that the switch is on.

Bit 3 — VSYNC pulse in color mode, video in monochrome mode.

Bit 4 — Always low

Bit 5 — Always low

Bit 6 — Always low

Bit 7 — Always high in color mode, negative VSYNC in monchrome mode.

Clear Light Pen - I/O address 3DBh (color), 3BBh (mono) (Write only)

Writes to this address are used to clear the light pen latched condition.

Preset Light Pen - I/O address 3DCh (Write only)

This address is written to set up the light pen signal latched condition for testing purposes when in color mode. If monochrome mode is selected, this function is performed by I/O address 3B9.

Memory Mode Register - I/O address 3BFh (Write only)

The Memory Mode Register is active only in monochrome modes and is used to set the wraparound and display areas.

Bit 0 0 = Alphanumeric (4K byte wrap)
1 = Graphics

Bit 1 0 = Page 1 (32K byte wrap)
1 = Page 1 & 2 (64K byte wrap)

Bit 2-7 Not Used.

Extended Control Register - I/O address 25Ah/35Ah (Write only)

Writes to the Extended Control Register are decoded as shown below.

Bit 0 Text Cell Size Selection
0 = 8 × 8
1 = 8 × 16
Bit 0 controls which character set is used for text display. Monochrome and Hi-Def modes require the 8 × 16 cell size, while 8 × 8 is used for standard color mode.

Bit 1 Line Doubler Mode
0 = Single Line
1 = Double Line
Line doubling is used for the Hi-Def and monochrome modes. This results in each scan line being displayed twice so that graphic images have a higher resolution. It is also used to blank the display borders.

Bit 2 Mapper
0 = No Map
1 = Mapper Enabled
The mapper is a translation circuit which causes substitute parameters to be sent to the 6845 style CRTC controller logic so that normal parameters for standard color operation are changed to parameters for Hi-Def operation. These include the selection of the 8 × 16 cell size, higher scanning rate and 400 active display lines. The mapper allows standard application packages to be displayed using Hi-Def mode, providing higher quality text and fuller graphics without software change.

Bit 3 Monochrome
0 = Color
1 = Monochrome
This bit controls monochrome operational differences between color and monochrome. If software is used to set this bit, rather than the M1/M2 selection inputs, then bit 0 (8 × 16 cell size) and bit 1 (Line Doubler) must also be set. Bit 3 controls the following:

- Changes the 72C81 registers so they respond to I/O address 3BXh rather than 3DXh.
- Changes the frame buffer addressing to B0000h from B8000h.
- Changes the Light Pen preset from 3DCh to 3B9h.
- Changes the basic timing from 8 clocks per character to 9 clocks per character in Text mode.
- Enables reverse video and underline attributes for text mode and inverts the VSYNC pulse.
- Automatically sets the 40/80 line width bit in register 3B8h to 80 for text operations.
- Changes the memory mapping in graphics operations.
- Disables the Red and Blue video outputs (held low) and outputs monochrome video on the Green channel.
- Automatically sets bit 4 at 3B8h for 640 × 200 graphics.

Bit 4 CGMA Disable
0 = Normal
1 = Disable

Bit 5 Reserved

Bit 6 Test Mode Select

Bit 7 Software Reset

Extended Status Register - address 25Ah/35Ah (Read only)

This register allows software visibility of modes selected by M1/M2 at power-up plus those attributes set by the Extended Control Register.

Bit 0 0 = 8 × 8 Text Cell Size
1 = 8 × 16 Text Cell Size

Bit 1 0 = Normal Graphics
1 = Line Doubled Graphics

Bit 2 0 = No Mapper
1 = Mapper Enabled

Bit 3 0 = Color
1 = Monochrome

Bit 4 0 = 72C81 Active
1 = 72C81 Disabled

Bit 5 Always low

Bit 6 M1 input value

Bit 7 M2 input value

M1/M2 Control Register - address 25Bh/35Bh (Write only)

This register is used for software selection of the adapter operating mode (CGA, MDA/ HGA, or Hi-Def).

Bit 0-5 Reserved - always written as zeros

Bit 6 M1

Bit 7 M2

M1	M2	Mode
0	0	CGA
0	1	Hi-Def
1	0	MDA/HGA
1	1	Reserved

CRT Controller

Internal to the 72C81 is a CRT controller which is software compatible with a 6845 CRT Controller and performs the same functions except for interlace capability. To address the controller's pointer register, use an even address (3D0h, 3D2h, 3D4h, or 3D6h). To store or retrieve a value from a particular data/control register, use an odd I/O address (3D1h, 3D3h, 3D5h, or 3D7h). The data/control registers inside the 72C81 CRT controller are ordered the same as the 6845 and use the same individual bit programming. Refer to a Motorola 6845 data sheet for detailed information.

CRT Controller Timing Illustration

The following chart indicates normal register programming for the CRT controller corresponding to the various modes. The values shown for the Hi-Def mode are for reference only. These values are substituted for standard color values by the mapper circuitry. These values are in Hexadecimal.

	MDA/HGA (MONO)		CGA			HI-DEF		
	Mono Text	Mono Grfx	40 Col Text	80 Col Text	Grfx Mode	40 Col Text	80 Col Text	Grfx Mode
R0	61	35	38	71	38	31	63	31
R1	50	2D	28	50	28	28	50	28
R2	52	2E	2D	5A	2D	28	50	28
R3	0F	07	0A	0A	0A	4	4	4
R4	19	5B	1F	1F	7F	1A	1A	6B
R5	06	02	06	06	06	6	6	6
R6	19	57	19	19	64	19	19	64
R7	19	57	1C	1C	70	1A	1A	67
R8	02	02	02	02	02	2	2	2
R9	0D	03	07	07	01	F	F	3
R10	0B	00	06	06	06	#	#	#
R11	0C	00	07	07	07	#	#	#

# R10 = CGA R10 X 2
R11 = (CGA R11 X 2) + 1

6845 Register Reference

Register Address	Register Number	Register Type	Units	Read/Write
0	R0	Horizontal Total	Character	Write Only
1	R1	Horizontal Display	Character	Write Only
2	R2	Horizontal Sync Position	Character	Write Only
3	R3	Horizontal Sync Width	Character	Write Only
4	R4	Vertical Total	Character Row	Write Only
5	R5	Vertical Total Adjust	Scan Line	Write Only
6	R6	Vertical Displayed	Character Row	Write Only
7	R7	Vertical Sync Position	Character Row	Write Only
8	R8	Interlace Mode		Write Only
9	R9	Maximum Scan Line Address	Scan Line	Write Only
A	R10	Cursor Start	Scan Line	Write Only
B	R11	Cursor End	Scan Line	Write Only
C	R12	Start Address (High)		Write Only
D	R13	Start Address (Low)		Write Only
E	R14	Cursor Address (High)		Read/Write
F	R15	Cursor Address (Low)		Read/Write
10	R16	Light Pen (High)		Read Only
11	R17	Light Pen (Low)		Read Only

System Interface

Monitor Outputs

The outputs to the monitor are comprised of Red, Green, Blue, Intensity and the Horizontal and Vertical Sync pulses. These outputs are TTL/CMOS compatible with high drive capability and are tri-stateable via the CGMA Disable Mode.

In Monochrome Mode, Red and Blue are at a logic low level while Green and Intensity contain the video information. The Horizontal sync is a positive pulse with a frequency (line rate) of 18.4 KHz. The Vertical sync is a negative polarity pulse output of 50 Hz which allows compatibility with standard monochrome PC monitors with high persistence phosphors (P39).

In standard color mode, Red, Green, Blue, Intensity signals are provided. The Horizontal sync signal is a positive polarity pulse with a scanning frequency of 15.75 KHz. The Vertical sync pulse is of positive polarity and has an output frequency of 60 Hz, providing compatibility with standard color PC monitors.

In Hi-Def color mode, signals are the same as in standard color mode except that the horizontal rate is 25 KHz and the vertical rate is 57 Hz.

CGMA Monitor Parameters

The frequencies listed below apply only if the standard video rate frequencies are used.

PC Interface

The PC interface consists of the Data Bus: D0 - D7, the Address bus: A0 - A19 and control signals AEN, MEMR/, MEMW/, IOR/, IOW/, RESET, and IORDY for PC wait state control. These signals allow the 72C81 to be interfaced directly to the PC bus.

Light Pen

The light pen circuitry consists of two input signals, the light pen switch and the light pen input. The light pen switch only goes back to the host processor as a status bit (a zero indicates the switch is made). The light pen input is latched and goes to the host as a status bit but also causes the coincident raster point to be stored for retrieval by the host. The latched signal is reset under program control.

> Note: For software compatibility, the light pen should be used in non-Hi-Def modes of operation.

Display Buffer Interface

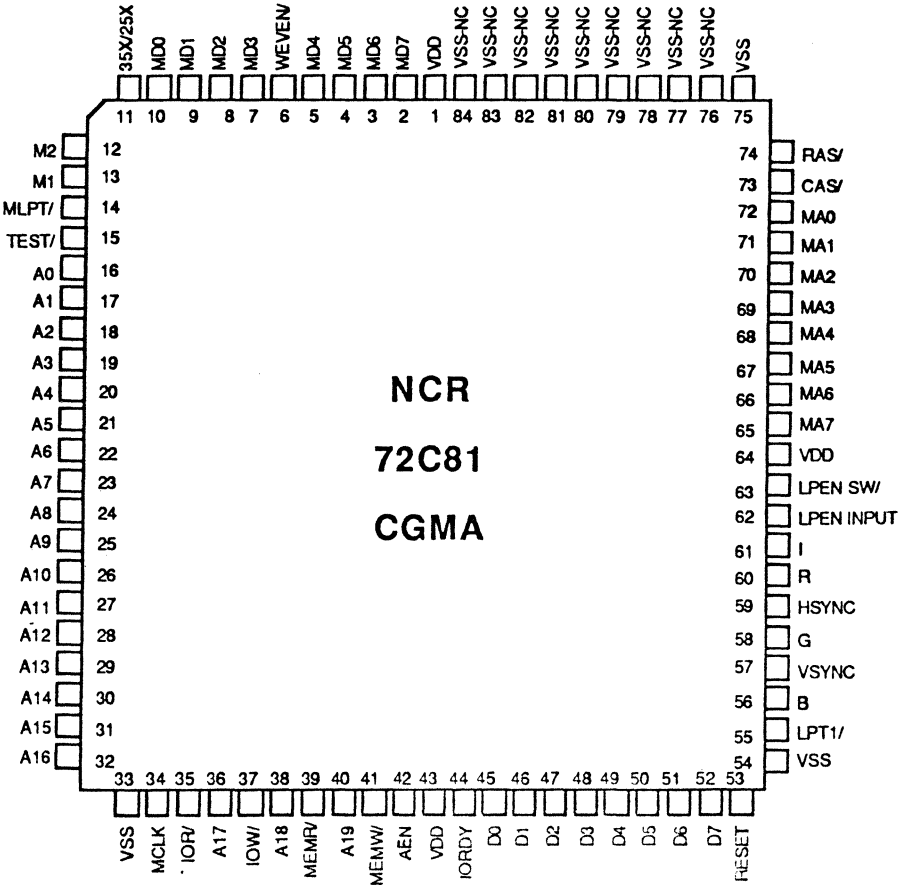
The display buffer interface is comprised of 8 address, 16 data, and 4 control lines. The Memory Address lines, MA0 - MA7, are multiplexed lines which provide row and column addresses which are clocked by RAS/ and CAS/ signals. The memory data lines, MD0 - MD15, are the bidirectional bus between the 72C81 and the frame buffer. WEVEN/ is low when data is written to the frame buffer and is high when data is read from the frame buffer. Memory timings are compatible with 16K × 4 and 64K × 4 DRAMs. The 64K × 4 parts are required if two page monochrome mode is used.

> Note: Standard color (CGA) mode requires DRAMs with 120 ns access, while Hi-Def and monochrome modes require 90 ns access.

	Monochrome	CGA Mode	Hi-Def Mode
Horizontal scan rate	18.432 KHz	15.750 KHz	25.000 KHz
Vertical scan rate	50 Hz	60 Hz	57 Hz
Video rate	16.257 MHz	14.318 MHz	20.000 MHz
Displayable colors	3	16	16
Character size	7 x 9 pixels	5 x 7 pixels	7 x 9 pixels
Character cell size	9 x 14	8 x 8	8 x 16
Maximum resolution	720 x 350	640 x 200	640 x 400

CGMA Monitor Parameters

84 Pin PLCC Diagram



PC Bus Interface Signals

SIGNAL	TYPE	DESCRIPTION
D0-D7	I/O	Data bits 0 to 7 are used for data transfers between the processor, memory, and I/O devices.
A0-A19	I	Address bits 0 to 19 address memory and I/O devices.
RESET	I	This input is used to initialize the 72C81 upon power-up.
IORDY	O	This output is pulled low to lengthen memory and/or I/O cycles for slow devices.
IOR/	I	I/O Read: This active-low signal instructs the 72C81 (or other addressed I/O devices) to drive its data onto the data bus.
IOW/	I	I/O Write: This active-low signal instructs the 72C81 (or other addressed I/O devices) to read the data on the data bus.
MEMR/	I	Memory Read: This active-low signal instructs the addressed memory to drive its data onto the data bus.
MEMW/	I	Memory Write: This active-low signal instructs the addressed memory to store the data present on the data bus.
AEN	I	Address Enable: This active-high signal indicates that a DMA transfer is taking place.

Monitor Interface Signals

SIGNAL	TYPE	DESCRIPTION
R	O	TTL/CMOS-level Red monitor drive - active for color modes of operation, held low for monochrome modes.
G	O	TTL/CMOS-level Green monitor drive - active for color modes, video for monochrome modes.
B	O	TTL/CMOS-level Blue monitor drive - active for color modes of operation, held low for monochrome modes.
I	O	TTL/CMOS-level Intensity monitor drive - active for color and monochrome display modes.
HSYNC	O	Horizontal Sync - initiates horizontal retrace.
VSYNC	O	Vertical Sync - initiates vertical retrace; polarity is inverted for monochrome modes.

Frame Buffer Interface Signals

MD0-MD15	I/O	Bidirectional memory data lines for data transfer between the frame buffer and the 72C81.
MA0-MA7	O	Multiplexed address lines for frame buffer DRAM row and column addresses.
RAS/	O	Row Address Strobe - used to strobe row addresses present on the memory address bus to the DRAMs.
CAS/	O	Column Address Strobe - strobes column addresses on the memory address bus to the DRAMs.
WODD/	O	Write Odd - Write enable signal for odd bank of the frame buffer. Used because writes to the frame buffer are eight bits wide while reads are 16 bits wide for flashless screen updates.
WEVEN/	O	Write even - write enable signal for even bank of the frame buffer.

Clock, Control and Power Signals

SIGNAL	TYPE	DESCRIPTION
MCLK	I	MCLK is the 72C81's pixel clock input and is equal to 14.318 MHz for standard color mode, 16.257 MHz for monochrome modes, and 20 MHz for Hi-Def mode. Refer to table, section 7.1.1.
TEST/	I	Test mode control. For normal operation, this pin is connected to the positive supply, VDD.
35X/25X	I	This input selects the I/O mapping of the Extended Control/Status Register and the M1/M2 Control Register. If this input is low, the Extended Control/Status Register is mapped at I/O address 25Ah and the M1/M2 Control Register is mapped to I/O address 25Bh. If this input is high, these registers are mapped to I/O addresses 35Ah and 35Bh respectively.
M1	I/O	Mode select line 1. (see M2 description)
M2	I/O	Mode select line 2. The mode lines M1 and M2 are bidirectional. When the RESET input is active, the logic levels on these pins are stored in the M1/M2 Control Register. When RESET is inactive, the values stored in the M1/M2 Control Register are driven out on these pins. This allows for external clock selection.
CSOUT/	O	This output is driven low when an I/O access is made at 25Eh to 25Fh, or 35Eh to 35Fh depending on the value of the 35Xh/25Xh input. This output can serve as a control for external devices.
LPEN SW/	I	The sense of this input is reselected in bit 2 of the Status Register (I/O address 3DAh).
LPEN INPUT	I	A high to low transition on this line causes the current refresh address to be latched into registers 16 and 17 of the 6845. The trigger for the light pen is reset by writing to address 3DBh. No specific data value is required to perform the reset, just the IOR/ access to 3DBh.
VSS	I	Power supply ground.
VDD	I	Positive five-volt supply input.

Parallel Port Select Signals

SIGNAL	TYPE	DESCRIPTION
MLTP/	O	This output is driven low when an I/O access is made at 3BCh to 3BEh and is used as a chip select for parallel port interface chips. This allows a printer port to be included with monochrome display adapter configurations.
LPT1/	O	This output is driven low when an I/O access is made at 378h to 37Fh and is used as a chip select for parallel port interface chips. This output is used for color display adapter configurations that include a printer port.

Electrical Specifications

D.C. CHARACTERISTICS (Ta = 0°C to 70°C, Vcc = 5V ±5%, Vss = 0V)

Characteristic		Symbol	Min.	Max.	Unit
Max. Power Supply Current		Idd		100	mA
Input Voltage					
A0-A19, IOR/, MEMR/, MEMW/, AEN, RESET, LPEN INPUT/, LPEN SW/, M2, M1, MCLK, 35X/25X, TEST/,MD0-MD15, D0-D7		ViL	-0.3	0.8	V
		ViH	2.0	VDD + 0.3	V
Input Current					
A0-A19, IOR/,IOW/, MEMR/, MEMW/, AEN, RESET, M2, M1, MCLK, ViH =VDD, ViL= 0		Iin	-10	10	µA
35x/25x, TEST/, LPEN SW/, LPEN INPUT		Iin	-400	10	µA
Output Voltage					
R,G,B,I, HSYNC, VSYNC MA0 - MA7, RAS/, CAS/, WODD/, WEVEN/	(IoL = 2mA IoH = -1mA)	VoL		0.4	V
		VoH	2.4		V
MD0-MD15	(IoL = 2mA IoH = -1mA)	VoL		0.4	V
		VoH	2.4		V
D0-D7	(IoL = 4mA IoH = -2mA)	VoL		0.4	V
		VoH	2.4		V
Output Load Capacitance					
R,G,B,I, HSYNC, VSYNC		CL		150	pf
MA0 - MA7, RAS/, CAS/, WODD/, WEVEN/		CL		40	pf
MD0-MD15		CL		50	pf
D0-D7		CL		100	pf

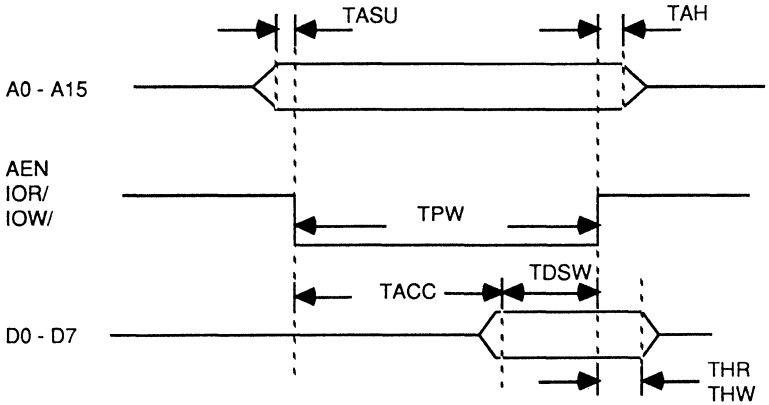
A.C. Characteristics

All timings are taken from the 10% or 90% points with respect to the specified VoL and VoH levels of the waveforms.

Data Bus Timings

Item	Symbol	Min	Typ	Max	Unit
Data Port Read:					
Select Enable	TPW	200	500		ns
Data Valid	TACC		100	80	ns
Data Hold - Read	THR	10	25	40	ns
Data Port Write:					
Select Enable	TPW	200	500		ns
Data Setup	TDSW	150			ns
Data Hold - Write	THW	50			ns
Address Setup/Hold :					
Before Write*	TASU	10	0		ns
After Write*	TAH	10	0		ns

*Note: Address must be stable during select times



Frame Buffer Read Timings

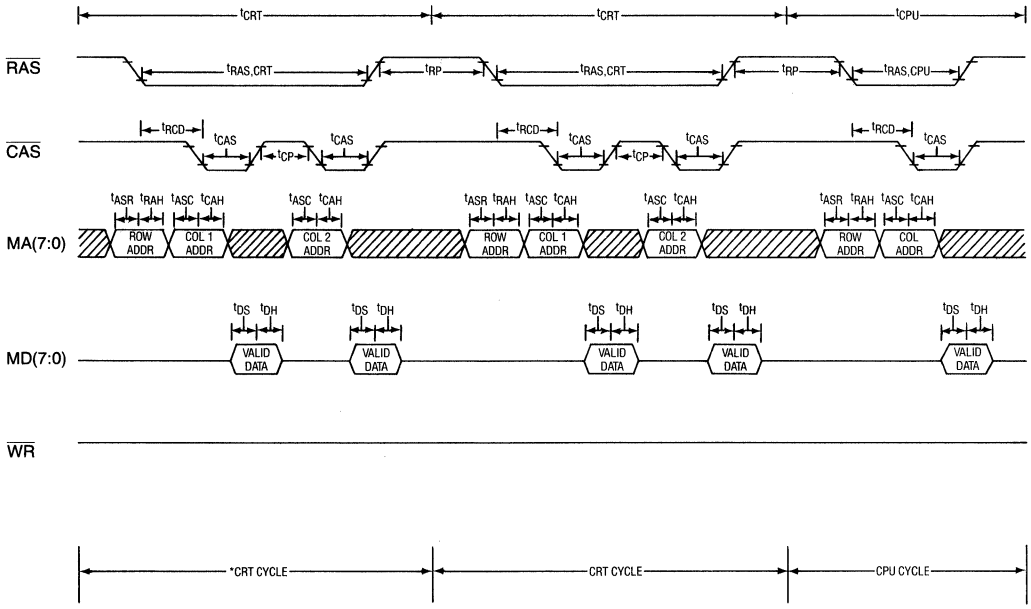
PARAMETER	SYMBOL	MIN	MAX
CRT Cycle Time	t_{CRT}	6T	6T
CPU Cycle Time	t_{CPU}	4T	4T
$\overline{RAS}$ Precharge Time	t_{RP}	2T – 3	2T + 3
$\overline{RAS}$ CRT Pulse Width	$t_{RAS,CRT}$	4T – 3	4T + 3
$\overline{RAS}$ CPU Pulse Width	$t_{RAS,CPU}$	2T – 3	2T + 3
$\overline{CAS}$ Precharge Time	t_{CP}	T – 3	T + 3
$\overline{CAS}$ Pulse Width	t_{CAS}	T – 3	T + 3
$\overline{RAS}$ to $\overline{CAS}$ Delay	t_{RCD}	T – 3	T + 3
Row ADDR Setup	t_{ASR}	1/2T – 10	—
Row ADDR Hold	t_{RAH}	1/2T	—
Col ADDR Setup	t_{ASC}	1/2T – 10	—
Col ADDR Hold	t_{CAH}	1/2T	—
Data in Setup	t_{DS}	5	—
Data in Hold	t_{DH}	0	—

T = MCLK PERIOD (ns)

Migration from 72C80 to 72C81

The NCR 72C80 chip normally requires a Display Buffer of four 64K×4 DRAM chips, while the newer NCR 72C81 needs only two 64K×4 DRAMs. Both of these versions are referred to as "CGMA." The 72C81 Display Buffer has an 8-bit data bus and uses Page-mode accesses during CRT cycles to gain equivalent 72C80 performance. The 72C80 and the 72C81 are functionally identical, as far as the user is concerned. On the 72C81, nine signals described below are labeled "VSS-NC." All other pins are identical between the two parts.

Frame Buffer Read Timings (CRT & CPU Reads)



NOTE: all timings are referenced to V_{IL} and V_{IH} levels.

* This CRT Cycle only occurs during Alphanumeric 80 column modes.

COMMUNICATIONS PRODUCTS

90C26 ARCNET CONTROLLER

Introduction

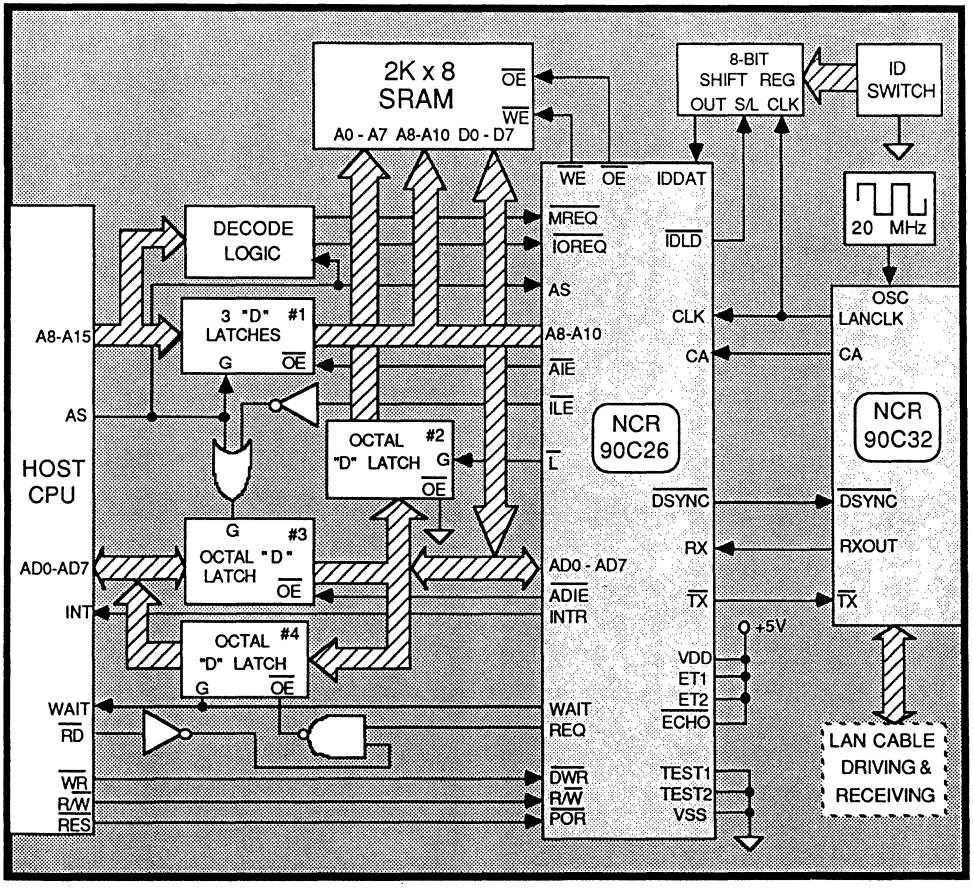
Arcnet is a registered trade mark of Datapoint Corporation. The NCR 90C26 AC (ARCNET Controller) and the NCR 90C32 AT (ARCNET Transceiver) together comprise all logical function of an ARCNET RIM (Resource Interface Module), in low power CMOS implementations. ARCNET is a popular token passing Local Area Network (LAN) scheme developed by Datapoint Corporation.

ARCNET is an ideal LAN for use in PC Networks, small office automation and factory automation.

Features

- Data transfers at 2.5 Megabits per second
- Supports up to 255 Local Area Network nodes
- Controls external SRAM containing host messages
- Token Passing Protocol gives predictable transfer intervals
- Interfaces with a variety of microprocessors
- Can transmit broadcast messages to all other nodes on LAN

Typical ARCNET Node

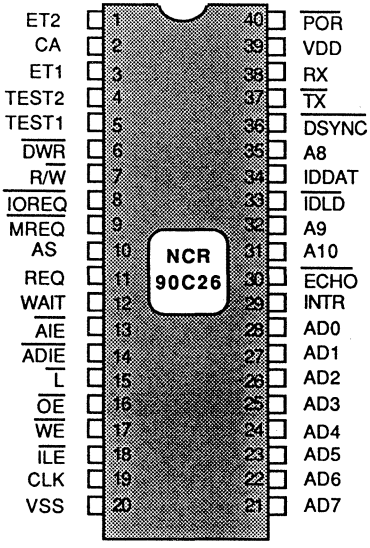


General Description

The NCR 90C26 AC (ARCNET Controller) is a powerful token-passing communications chip that handles all data transfer functions for a node on an ARCNET Local Area Network (LAN). With a flexible microprocessor interface, the AC accepts commands from its host regarding message packets to be sent or received over the LAN. These packets are stored in an external Static RAM that may be accessed jointly by the host or the AC. When a node is either added to or removed from the network, the active nodes will automatically "reconfigure" as directed by each AC. The diagram on the previous page shows the detailed connections for the AC in an ARCNET node. The NCR 90C32 ARCNET Tranceiver is a 16-pin companion chip that passes data between the AC and the actual LAN cable circuitry.

Pin Assignment

PIN ASSIGNMENT



90C26 Pin Descriptions

Symbol	Pin No.	I/O	Name and Function
A9, A10	32, 31	O	PAGE SELECT: The two most significant address bits to the RAM buffer. These pins reflect the page value specified in the ENABLE TRANSMIT amd ENABLE RECEIVE commands. For a 1K buffer they are connected to A8 & A9 of the RAM, and for a 2K buffer system, connect to A9 & A10 respectively.
A8	35	O	ADDRESS 8: Connects to RAM pin A8 in a 2K buffer system . This pin is left unconnected in systems with a 1K buffer.
AD0 - AD7	28 - 21	I/O	ADDRESS-DATA BUS: The eight-bit data path to the AC, and the lower eight address bits of the RAM buffer. In addition, AD0 is used for command decoding of host status/control commands to the AC.
$\overline{\text{IOREQ}}$	8	I	I/O REQUEST: This signal is sampled by the AC on the falling edge of AS, and indicates that the host wishes to transfer status/control information over the Address-Data bus to the AC.
$\overline{\text{MREQ}}$	9	I	MEMORY REQUEST: This signal is sampled by the AC on the falling edge of AS, and indicates that the host plans to transfer data to or from the RAM buffer.
R/$\overline{\text{W}}$	7	I	READ / WRITE: Indicates that the pending host access request is either a read or a write cycle. R/W is sampled on the falling edge of AS.
AS	10	I	ADDRESS STROBE: Used by the AC to sample the state of R/$\overline{\text{W}}$, $\overline{\text{MREQ}}$, and $\overline{\text{IOREQ}}$. If necessary, the AC will initiate arbitration for the Address-Data bus on the falling edge of AS.
REQ	11	O	REQUEST: A feedback signal acknowledging that the AC has sampled a Memory or I/O request by the host.
WAIT	12	O	WAIT: This signal is asserted by the AC to indicate that it is not ready to transfer data, at the beginning of host access requests. WAIT is returned low when the AC is ready for the host to complete its transfer.
$\overline{\text{DWR}}$	6	I	DELAYED WRITE: A signal that informs the AC that valid data is set on the Address-Data bus, for host write cycles. $\overline{\text{DWR}}$ should be grounded, if the host will always be able to meet write data setup time.
INTR	29	O	INTERRUPT: The AC asserts this signal to indicate to the host that an enabled interrupt condition has occurred. INTR will return low upon the resetting of the interrupting status condition or the corresponding mask bit.
$\overline{\text{ILE}}$	18	O	INTERFACE LATCH ENABLE: The AC brings this signal low to gate the host's valid output data for host write cycles.
$\overline{\text{ADIE}}$	14	O	ADDRESS-DATA INPUT ENABLE: Tri-states Address-Data Bus pins in the AC. Enables the host to drive the Address-Data bus.
$\overline{\text{AIE}}$	13	O	ADDRESS INPUT ENABLE: Tri states pins A8 - A10 of the AC and enables the host to drive these signals.

90C26 Pin Descriptions (Cont'd)

Symbol	Pin No.	I/O	Name and Function
$\overline{L}$	15	O	LATCH: Gates the Address-Data bus into an external latch. This latch in turn is normally connected to A0 - A7 of the RAM buffer.
$\overline{WE}$	17	O	WRITE ENABLE: This signal strobes data off the Address-Data bus into the RAM buffer. Data is taken by the RAM at the rising edge of WE.
$\overline{OE}$	16	O	OUTPUT ENABLE: Enables the RAM buffer to drive the Address-Data bus on all RAM read cycles.
$\overline{IDLD}$	33	O	ID LOAD: When active, this signal synchronously loads the state of the ID switches into an external shift register.
IDDAT	34	I	ID DATA IN: This signal is the serialized ID switch setting from the ID switch register. The ID data is shifted into the AC MSB first, and with high level indicating logic 1.
ET1, ET2	3, 1	I	EXTENDED TIMEOUT FUNCTIONS: The levels on these pins indicate the timeout durations of the AC in checking responses from the other nodes on the LAN. These pins should be tied high for normal operation.
$\overline{TX}$	37	O	TRANSMIT DATA: Serial output data to the Transceiver chip.
RX	38	I	RECEIVE DATA: Serial input data from the Transceiver chip.
TEST1, TEST2	5, 4	I	TEST PINS 1 & 2: Used to set up special diagnostic tests. For normal operation, these pins are grounded.
$\overline{ECHO}$	30	I	ECHO DIAGNOSTIC ENABLE: When low, this signal enables a re-transmit of messages whose length is less than 254 bytes, for level testing. This pin is normally tied high.
CLK	19	I	CLOCK: A free-running 5 MHz clock used for timing of AC bus cycles and other various chip functions.
CA	2	I	AUXILIARY CLOCK: A gated 5 MHz clock used to control operation of the AC sequencer. CA is periodically halted by $\overline{DSYNC}$ and restarted by RX.
$\overline{DSYNC}$	36	O	DELAYED SYNC: This signal is asserted by the AC to cause the clock generator circuit to halt the CA clock.
$\overline{POR}$	40	I	POWER ON RESET: Sets the AC sequencer counter to zero and initializes various internal control and status signals. Also sets the POR status bit which causes the INTR pin to be asserted.
VDD	39		+5 VOLT SUPPLY.
VSS	20		GROUND.

Description of Line Protocol

The NCR 90C26 ARCNET Controller implements an isochronous line protocol, with each character byte consisting of: 2 clock units of mark (logic 1), 1 clock unit of space (logic 0), and 8 clock units of data (the character). A single clock unit is 200 nanoseconds in duration, so a byte of data will be transmitted every 4.4 microseconds. Thus, the time to transmit any message can be determined exactly. All transmissions start with an ALERT BURST, which is 6 clock units of mark. The line idles in a spacing condition. The five types of ARCNET transmissions are now described:

Invitations To Transmit:

An ALERT BURST followed by three characters: one EOT (End Of Transmission) and two repeated DID (Destination IDentification) characters. This message passes control (the "token") from one node to another.

Free Buffer Enquiries:

An ALERT BURST followed by three characters: one ENQ (ENQuiry) and two repeated DID characters. This message asks another node if it is able to accept a message packet.

Packets:

An ALERT BURST followed by from 8 to 260 characters:

- one SOH (Start Of Header) character
- one SID (Source IDentification) character
- two repeated DID characters
- an inverse COUNT character = 250 −N, for N data characters to be sent
- from 1 to 253 data characters
- two CRC (Cyclic Redundancy Check) characters

Acknowledgements:

An ALERT BURST followed by a single ACK (ACKnowledgement) character. This message is used as a positive to FREE BUFFER ENQUIRIES, and also to acknowledge the valid reception of a PACKET.

Negative Acknowledgements:

An ALERT BURST followed by a single NAK (Negative AcKnowledgement character. This message gives a negative response to FREE BUFFER ENQUIRIES.

Notes:

The codes (all in HEX) for the special characters mentioned above are now given:

EOT-04 ENQ-05 SOH-01 ACK-06 NAK-15

The COUNT character for PACKETS may be = 512 − N if a "long packet" is being sent. The CRC polynomial used for data packets is:

$$X^{16} = X^{15} + X^2 + 1.$$

As receiving node, the AC will verify all incoming transmissions by checking for: 1] at least one mark and exactly one space preceding each character 2] a valid EOT, ENQ, SOH, ACK, or NAK following the ALERT BURST 3] proper CRC for data packets 4] correct number of characters, depending on the transmission: can be from 1 to 260 5] at least nine spaces following the very last character.

Network Control

All nodes in an ARCNET system are distinguished by a unique 8 bit ID (IDentification) value which is determined by switches, typically a DIP switch, that is associated with AC chip. An ID of '0' may not be assigned to any node however, since that ID indicates a BROADCAST to all nodes. Control of the Local Area Network (or "LAN") is based on an INVITATION TO TRANSMIT being passed between nodes. When an AC receives an INVITATION containing its own ID, it has "gotten the token," and it then takes control of the LAN. To pass a message, the host processor simply loads the message data and the destination ID into its AC's RAM buffer, and then writes an "Enable Transmit" command to the AC. An AC will know it has a message to send if it sees that TA (Transmitter Available) in its Status Register is LOW. When that AC has the token, it then transmits a FREE BUFFER ENQUIRY to the destination ID to see if it is able to take the message. If that destination is able to receive, it will transmit back an ACK. The controlling node then transmits the PACKET, complete with a 16 bit CRC. When an AC receives the token, but its TA is high (it has no message to send), it will send INVITATION TO TRANSMIT and thus pass on the token.

When an AC is sent FREE BUFFER ENQUIRY, it will poll its RI (Receiver Inhibited) Status bit. If the RI bit is set, that AC will transmit a NAK, and the controlling node will then pass the token. If an AC with a packet to send gets a NAK, then it will re-transmit a FREE BUFFER ENQUIRY the next time it receives the token. These attempts will continue until the AC "Times out," a process described in detail in a following section.

Upon Timeout, the AC sets its TA bit, and passes the token. After it has sent a PACKET, an AC waits a specified response time: if within that time, it receives ACK, it sets both the TMA (Transmit Message Acknowledged) and the TA Status bits and passes the token. If it does not receive ACK in time, it only sets TA and then passes the token.

All nodes recognize a PACKET when they see the SOH character, and all the ACs will write the SID into their Receive Buffer. If an AC perceives the first DID as its own, or the PACKET is a Broadcast (described below), the chip will write the second DID and the rest of the message into its Receive buffer. Otherwise, the AC will ignore the rest of the PACKET. After the PACKET has been fully received, it must pass three conditions to be considered a valid message: a) the CRC comparison b) correct length of characters, and c) valid DID in byte 0 of the Receive Buffer. Valid DIDs are either '0' (indicating Broadcast), or the AC's own ID. If a message is valid by these conditions, then the AC will set its RI Status bit. However, if the DID is that of the AC, it will transmit an ACK first before setting RI. If any of the conditions fail, the AC ignores the message and will write it over with any future PACKETS.

Reconfiguration and Broadcast

There are two activities that involve all nodes on the ARCNET system. The first is a Reconfiguration of the network, which is performed any time a node is removed from, or added to the system. Specifically, an AC will instigate a Reconfiguration when it is first powered on, or when it has not received an INVITATION TO TRANSMIT within 840 milliseconds. It does this by transmitting a RECONFIGURATION BURST: 8 marks and 1 space repeated 765 times. This burst has the effect of terminating all activity on the network. As it is longer than all the other types of transmissions, it will interfere with the next INVITATION TO TRANSMIT, in effect "destroying" the token, and thus no other node will take control of the line. The RECONFIGURATION BURST also provides enough line activity so that the AC that just sent the token will also release control of the network.

When any AC sees the line idle for 78 us, it will know that the network is being Reconfigured, and it will set an internal NID (Next ID) register to its own ID. The NID is normally the DID sent with an INVITATION TO TRANSMIT. Besides resetting its NID, an AC also starts a timeout of 146 us times the quantity 255 minus its own ID. If this timeout expires with no other line activity, the AC will start transmitting INVITATIONS TO TRANSMIT, with DID set pointing to itself. Only the AC with the largest ID value will actually timeout, however.

After sending an INVITATION TO TRANSMIT, the AC will look for any line activity, indicating the the DID is a valid node. If the sending AC detects no activity after 74 μs, it increments its NID, and sends another INVITATION. Eventually, the AC with the ID that should be next, will see its ID in the INVITATION, and it will take control of the line. The previous AC will then have its NID set correctly. The process repeats, with the end result that each AC will have a NID stored which will represent a truly active node to which it will always pass the token. No time will be wasted in trying to send the token to nonexistent nodes. If for some reason a node is removed from an active network, then when the previous node tries to pass the token, it will timeout. The previous AC will go through a cycle of incrementing its NID and transmitting INVITATIONS TO TRANSMIT, until it finds the next valid node. The total time to perform a Reconfiguration will vary depending on the system configuration, but is typically between 34 to 61 ms.

The other operation that pertains to all the nodes in the network is a Broadcast Packet. Broadcast Packets are a special subset of PACKETS. A PACKET is considered to be Broadcast if the DID is a value of '0.' No regular node may be assigned to the Broadcast ID. Nodes are set up to receive Broadcasts by issuing an "ENABLE RECEIVE to Page nn" command to the corresponding AC, with the most significant bit of the command set to '1.' All the AC commands are described in a following section.

Timecheck Functions

The various timeouts that have been mentioned previously are now summarized. A standard baseband system using RG-62 coax cable (the ARCNET standard) can take up to 31 μs for a one-way propagation, which corresponds to a distance of about 4 miles. Also, the maximum turnaround time that any AC takes to respond to an incoming message is 12 μs. Therefore, a maximum Response Time for any transmission should be 31 + 31 + 12 = 74 μs. To allow margin, the AC uses 74.7 μs as its basic Response timeout, the interval within which a controlling node expects to perceive any line activity after it makes a transmission.

An IDLE Timeout is the interval that transpires at the onset of a Reconfiguaration. After the RECONFIGURATION BURST, all the nodes will commence the Reconfiguration process after they have seen no line activity for the Idle timeout. In a standard network, the

Idle timeout is 78.2 μs. The TRANSFER Timeout is the ID-dependent interval that is also associated with Reconfiguration. This timeout is given by 146 μs $\times$ (255 $-$ ID), and it will transpire only for the node with the highest ID on the network. The last timeout is the interval that instigates a Reconfiguration. In a standard network, if any node has not received an INVITATION TO TRANSMIT within a Reconfiguration timeout of 840 ms, it will issue a RECONFIGURATION BURST and thus start a network Reconfiguration.

The above timeout values apply to a standard, or "basic" network with no 2 nodes further apart than about 4 miles. The network may operate over longer distances by appropriate setting of the ET1 and ET2 inputs. The table below shows the effect of ET1 and ET2 on two of the more pertinent timeouts. It is important that ET1 and ET2 be set to the SAME VALUE for all nodes on the network.

ET2	ET1	RESPONSE TIMEOUT	RECONFIGURATION TIMEOUT
1	1	74.7 µs	0.84 second
1	0	283.4 µs	1.68 seconds
0	1	561.8 µs	1.68 seconds
0	0	1118.6 µs	1.68 seconds

Host Interface

The "Typical ARCNET Node" diagram on page 1 of this manual illustrates the hardware aspects of a host processor interface to the 90C26 AC. The block on the left contains the signals of a generic processor with 8 bits of multiplexed address-data. The upper address bits are used to decode whether a pending host access is to/from the RAM buffer, or the AC. The RAM buffer in the drawing, a 2Kx8 Static RAM, can hold four pages at 512 bytes per page. Entire PACKETS are stored in a page in the RAM buffer. A 512 byte page corresponds to a Long Packet. Short Packets require 256 byte pages and a RAM buffer which is only a 1Kx8 SRAM. Leave A8 (from both the AC and the host) unconnected for a Short Packet network. The #1 latches capture the host's upper address signals for RAM buffer accesses. The #2 8-bit latch always contains the lower 8 address bits for RAM buffer accesses, by both the AC and the host. For host write cycles, latch #3 buffers both lower RAM address and Data under control of the AC. Finally, latch #4 holds Data that the host has asked for in a read cycle. A recommended device for these latches is 74LS373.

In dealing with an ARCNET node, the host need only be concerned with 2 I/O locations, described below, and the 1K or 2K bytes in the Message RAM buffer. The AC chip was designed to allow speedy data transfers between the host and the Local Area Network. The organization of the 4 pages in the RAM allow for one message to read out while concurrently a follow-on message is being written to the RAM. By transparently making alternative accesses, the AC gets a dual port implementation out of a standard component RAM. Since the host makes it access request at arbitrary times, the AC synchronizes the host by asserting WAIT at every access attempt by the host. In addition to passing messages, the host also used the Address-Data bus and the latches to access the AC directly. A low level on $\overline{\text{IOREQ}}$ at the falling edge of AS identified I/O cycles, which are used to read the AC's Status, or to write commands to the AC. The I/O accesses pass through 2 I/O ports within the AC. The ports and their direction are decoded by the states of AD0 (at the host) and R/$\overline{\text{W}}$ (at the AC) at the falling edge of AS as follows:

AD0	WRITE OPERATION (R/$\overline{\text{W}}$ low)	READ OPERATION (R/$\overline{\text{W}}$ high)
0	Write Interrupt Mask	Read Status Register
1	AC Command	[reserved function]

Commands

The six valid host commands to the AC are now described, followed by a summary of the Status Register bits:

DATA	FUNCTION
0000 0001	DISABLE TRANSMITTER: Causes an AC to cancel any pending transmit command. This command will cause the TA bit to be set the next time the AC receives the token.
0000 0010	DISABLE RECEIVER: Causes a pending receive command to be canceled. This command will cause the RI bit to be set the next time the AC receives the token. If a PACKET has already started arriving, then this command will have no effect.
000n n011	ENABLE TRANSMIT from Page nn: Tells the AC to prepare for a transmit operation out of RAM buffer page nn when it next receives the token. The TA and TMA bits are reset by an AC receiving this command. The TA bit is set to logic '1' at completion of the transmission; the TMA bit may already be set at that time if the destination node has sent back an acknowledgement. If TA is not true, this command should not be issued.
b00n n100	ENABLE RECEIVE to Page nn: Allows the AC to receive messages into RAM buffer page nn. The RI status bit is set to '0' by this command. If 'b' in the command is '0', then only messages addressed to the AC's ID will be received. If 'B' is '1', then Broadcast messages will also be accepted. RI will be set by a successful message reception.
0000 s101	DEFINE BUFFER SIZE: Tells an AC the size of its RAM buffer. If bit 's' is '0', the buffer is 1K bytes, and only short packets may be sent and received. If 's' is a '1', the buffer is 2K bytes, and both short and long packets may be handled.
000r p110	CLEAR FLAGS: Resets the POR and/or the RECON status bits depending on the variable bits. If 'r' is '1', the RECON flag is cleared, and if 'p' is '1', then POR is cleared.

Status Register

BIT	NAME	DESCRIPTION
0	TA	Transmitter Available: When = 1, this bit indicates that the node is available to carry out a transmit sequence, and any previous ENABLE TRANSMIT process has been completed.
1	TMA	Transmit Message Acknowledged: This bit, when set to 1, indicates that a message sent from a previous ENABLE TRANSMIT command was acknowledged by the receiving node.
2	RECON	Reconfiguration Flag: When = 1, this bit indicates that a system Reconfiguration took place due to the expiration of an Idle timeout. RECON is reset by the CLEAR FLAGS command.
3	TEST	Test Flag: Intended for test and diagnostic purposes. This bit will be a '0' under normal operating conditions.
4	POR	Power on Reset: When = 1, this bit indicates that the AC has experienced a Power Reset due to an active signal on the $\overline{\text{POR}}$ input. POR is reset by CLEAR FLAGS.
5	ETS1	Extended Timeout Status 1: The state of this bit reflects the logic level on the ETS1 pin. ETS1 will be '1' under normal operating conditions.
6	ETS2	Extended Timeout Status 2: Reflects the level on the ETS2 pin. Normally = '1'.
7	RI	Receiver Inhibited: When = 1, this bit indicates that the AC is not taking any messages from other nodes. If an ENABLE RECEIVE command has been issued to the AC, and no Power Reset has occurred, then RI = 1 means that a message has been received into the RAM buffer over the network.

NOTE: If an AC experiences a Power Reset, then the Status Register assumes the following states, where 'x' indicates "same as previous state."

BIT:	7	6	5	4	3	2	1	0
State:	1	x	x	1	0	0	0	1

Host Interrupts

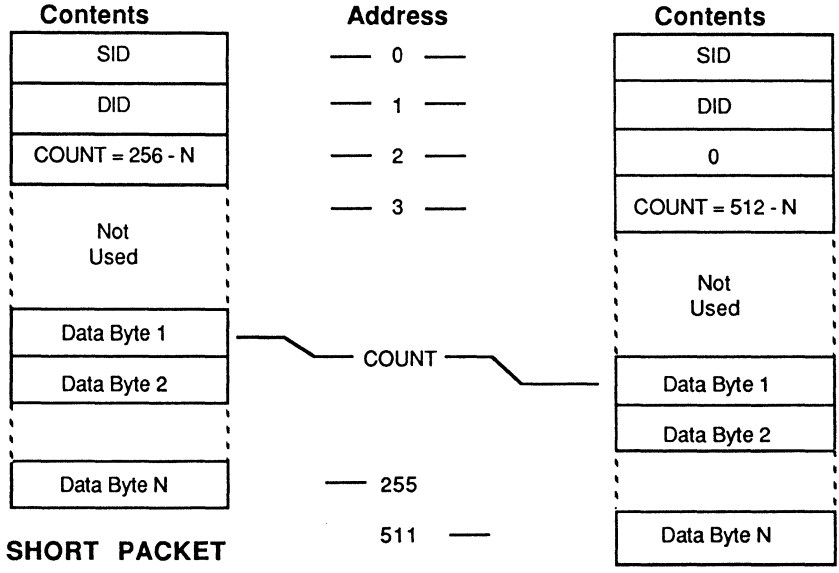
The 90C26 AC is able to generate an interrupt signal on the INTR pin in response to most of the conditions that cause Status bits to be set. A POR condition causes an unconditional (non-maskable) interrupt. There is a Mask Interrupt Register where any combination of four other Status conditions may be set to cause interrupts. Status bits TMA, ETS1, and ETS2 have no corresponding mask bits and can never cause interrupts. The 4 maskable Status bits are grouped in the following table:

Setting any of these bits to a '1' will cause the INTR signal to become active whenever the corresponding Status Condition becomes true. The other 4 bits in the Mask Register should be considered as "don't care." Once the INTR signal is active (high), it can be deasserted by either A) getting the corresponding Status Register bit to be reset, or B) resetting the Mask Register bit directly.

STATUS CONDITION	RECEIVE INHIBIT	TEST	RECON TIMEOUT	TRANSMITTER AVAILABLE
MASK BIT	7	3	2	0

RAM Buffer Map

The host needs to know the proper locations of the components in a message PACKET. The following map diagrams show these locations for both Short Packets (pages up to 256 bytes in length) and Long Packets (for pages up to 512 bytes long).



NOTES: • Addresses shown are relative to a Page, not absolute • SID = Source ID (not written in Transmit Packets) • DID = Destination ID (set = 0 for Broadcasts) • N = Message Length • "Not Used" bytes imply message is less than maximum length. These bytes would be written for Max. Messages: SHORT = 253 bytes, LONG = 508.

Electrical Specifications

Absolute Maximum Stress Ratings

Voltage on VDD pin with respect to Vss ..... −0.3 to +7.0 V
Voltage on any signal pin ..... −0.3 to VDD + 0.3
Power Dissipation ..... 100 mW
Operating Temperature Range ..... 0 °C to 70 °C
Storage Temperature Range ..... −65 °C to +150 °C

The values listed above are absolute maximum which if exceeded could cause permanent damage to the device. Voltages are with respect to circuit ground.

DC Characteristics

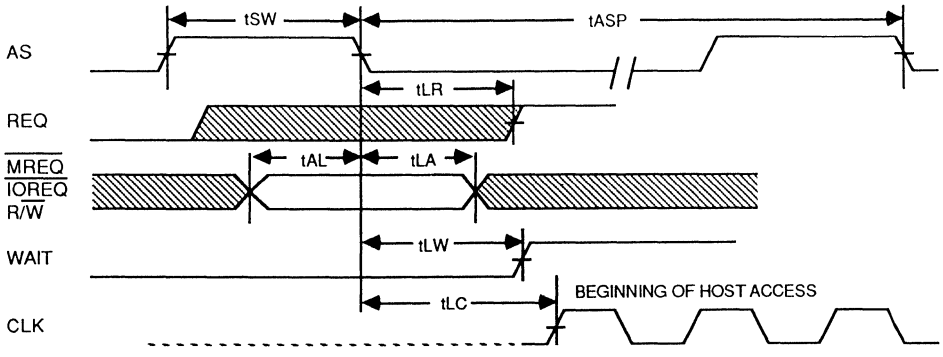
Symbol	Parameter	Min	Max	Units
V_{IL}	Input Voltage, Low	V_{SS} − 0.3	0.8	V
V_{IH}	Input Voltage, High	2.2	V_{DD}	V
$V_{IH}CLK$	Clock Input Voltage High (CA and CLK pins)	V_{DD} − 0.5	V_{DD} + 0.3	V
V_{OH}	Output Voltage, High (I_{OH} = −50 μA)	2.4		V
V_{OL}	Output Voltage, Low (I_{OL} = 2 mA)		0.5	V
C_{IN}	Input Capacitance		20	pf
C_{BS}	Data Bus Capacitance		50	pf
C_X	Remaining Pins,Capacitance		30	pf
I_I	Input Leakage Current (all input except $\overline{ECHO}$)		±10	μA
I_{DD}	Power Supply Current (V_{DD} = 5.25 V)		19	mA

Note: All parameters are assured over the 0°C to 70°C temperature range and with 4.75 ≤ V_{DD} ≤ 5.25 Volts. All voltages are with respect to circuit ground (Vss). The $\overline{ECHO}$ input has an internal pullup circuit.

Timing Diagrams

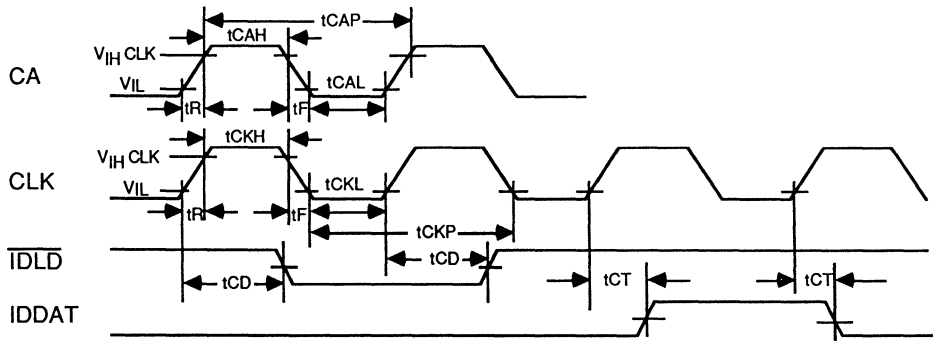
The following pages contain detailed diagrams illustrating the timing relations between the various AC device signals. Below each diagram is a table with the relevant time intervals specified. All timings are taken from the 10% or 90% points with respect to the specified VOL and VOH levels of the waveforms. A quick guide to the different timing diagrams is given below.

Processor Access Timing



NAME	DESCRIPTION	MIN	MAX	UNITS	NOTES
tSW	AS Strobe Width	50		ns	
tASP	Period of AS	3.5 tCAP		ns	
tLR	REQ delay after AS falling edge	0	100	ns	tCKP and tCAP are defined in diagram below.
tAL	Control Valid before AS falling edge	50		ns	
tLA	Control Hold after AS falling edge	50		ns	
tLW	WAIT delay after AS	0	200	ns	
tLC	Processor Cycle delay	tCKP	2 tCKP + 100	ns	

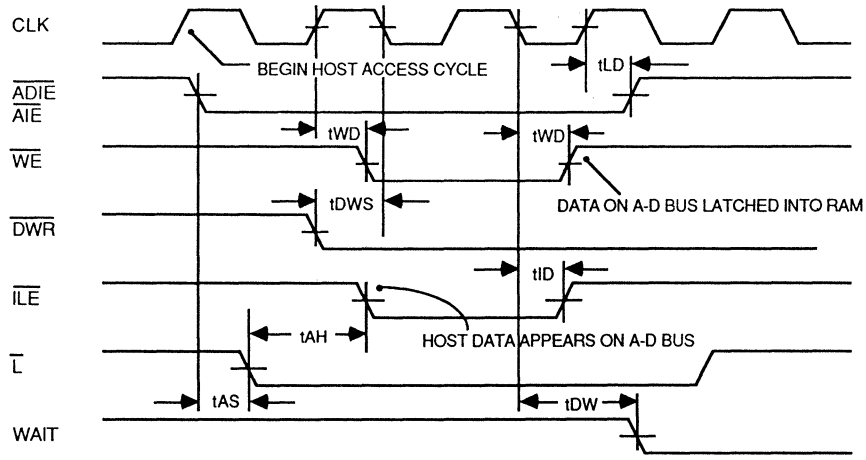
Clock Pulses & ID Input Timing



NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
tCAH	CA Pulse High Duration	60			ns
tCAL	CA Pulse Low Duration	60	100	300	ns
tR	Rise Time for CLK & CA			20	ns
tF	Fall Time For CLK & CA			20	ns
tCAP	CA Period	190			ns
tCKH	CLK Pulse High Duration	65			ns
tCKL	CLK Pulse Low Duration	65			ns
tCKP	CLK Period	190	200	600	ns

NAME	MIN	MAX
tCD, IDLD Delay from CLK Rising Edge	0	120 ns
tCT, IDDAT Delay from CLK Rising Edge	0	50 ns

Write RAM By Host Timing

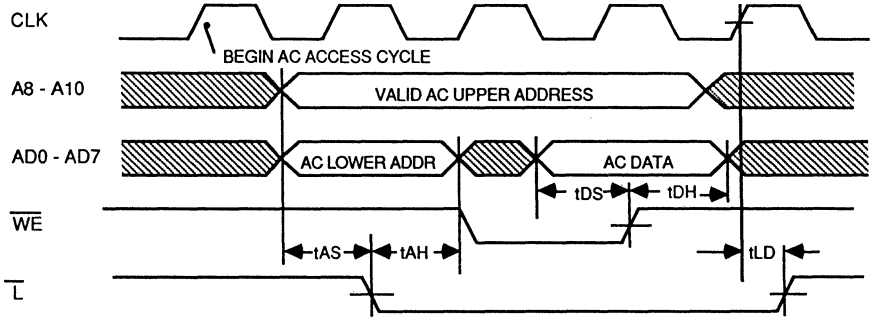


NAME	DESCRIPTION	MIN	MAX	UNITS
tWD	WE Delay from Clock edges	0	100	ns
tDWS	DWR Setup before Clock	50		ns
tID	ILE Hold after Clock	10	100	ns
tDW	WAIT Hold after Clock	20	100	ns

NOTES

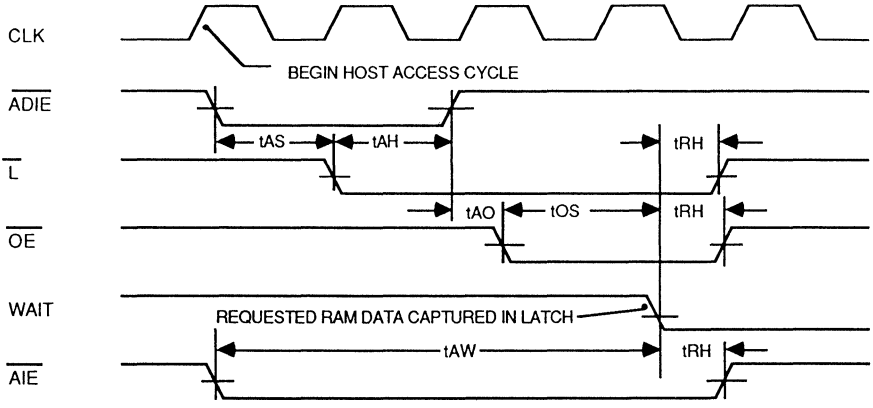
tLD, tAH, and tAS are specified below.

Write RAM By AC Timing



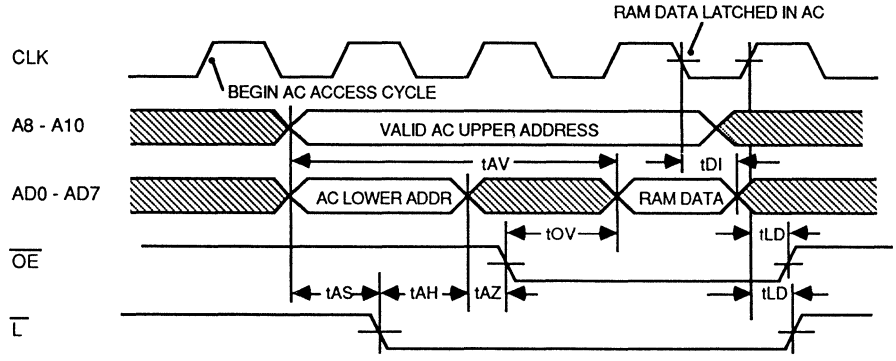
NAME	DESCRIPTION	MIN	MAX	UNITS
tAS	Address Setup to L	50		ns
tAH	Address Hold from L	50		ns
tDS	Data Setup to WE	60		ns
tDH	Data Hold from WE	50		ns
tLD	L turnoff delay from clock	0	100	ns

Read Ram By Host Timing



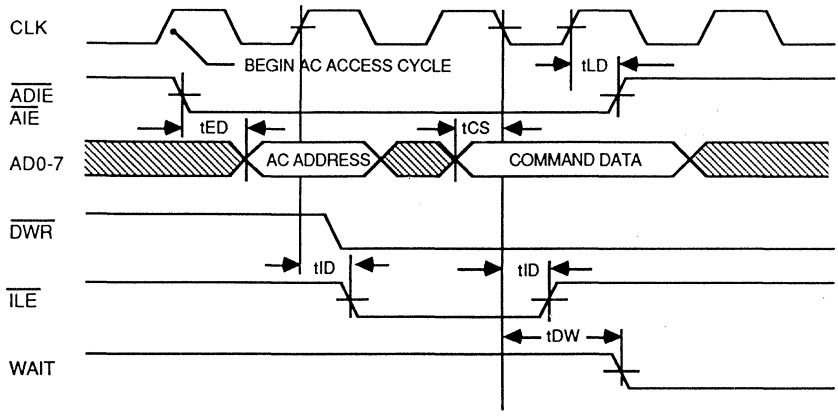
NAME	DESCRIPTION	MIN	MAX	UNITS
tRH	Read Data Hold after WAIT	20		ns
tAO	Delay from ADIE to OE Active	40		ns
tOS	OE Setup to WAIT Turnoff	140		ns
tAW	AIE Active before WAIT	300		ns
tAS	Address Setup to L	50		ns
tAH	Address Hold from L	50		ns

Read RAM By AC Timing



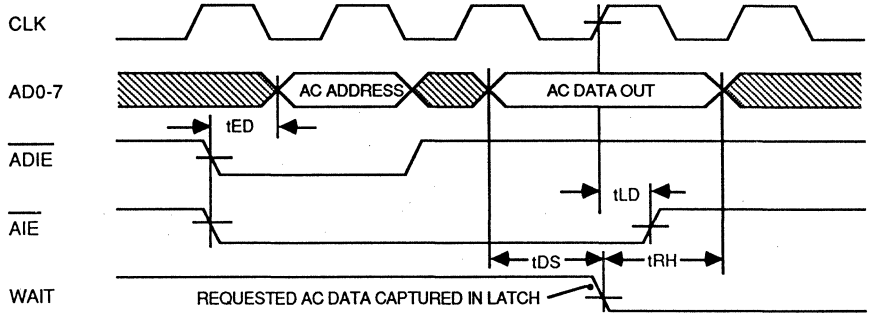
NAME	DESCRIPTION	MIN	MAX	UNITS	NOTES
tAV	Address Valid before RAM Data		300	ns	tAH and tAS are specified above.
tDI	RAM Data Hold for AC	80		ns	
tOV	RAM Data Delay after OE	0	140	ns	
tAZ	A-D Bus Hi Impedance before OE	0		ns	
tLD	L turnoff delay from clock	0	100	ns	

Write AC By Host Timing



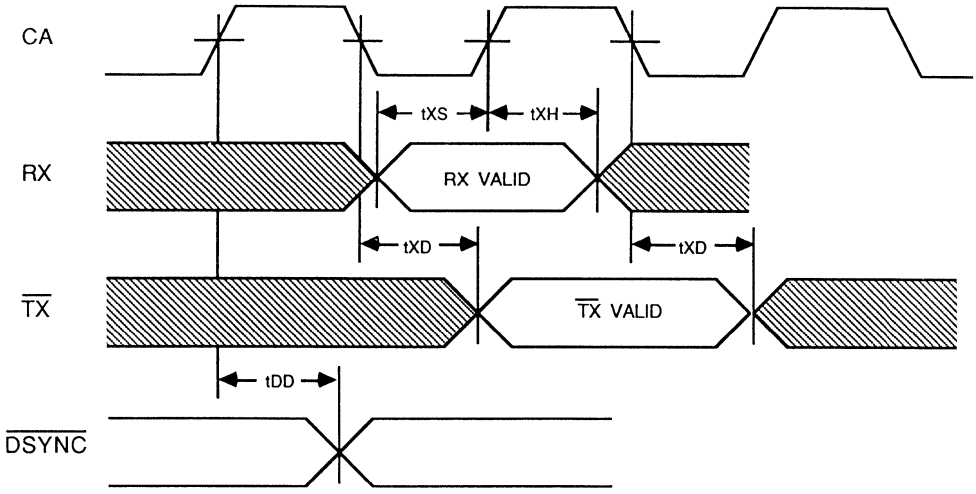
NAME	DESCRIPTION	MIN	MAX	UNITS
tED	Enable Setup before Address		50	ns
tCS	Command Setup before Clock	125		ns
tiD	ILE Delay after Clock	10	100	ns
tDW	WAIT hold after Clock	20	100	ns
tLD	Enable turnoff after Clock	0	100	ns

Read AC By Host Timing



NAME	DESCRIPTION	MIN	MAX	UNITS
tED	Enable Setup before Address		50	ns
tDS	AC Data Setup before WAIT	60		ns
tRH	AC Data Hold after WAIT	20		ns
tLD	AIE turnoff after Clock	0	100	ns

Receive & Transmit Timing



NAME	DESCRIPTION	MIN	MAX	UNITS
tXS	RX Setup to Clock Rising Edge	80		ns
tXH	RX Hold from Clock Rising Edge	30		ns
tXD	TX Valid delay from Clock Falling Edge	10	150	ns
tDD	Delay of DSYNC from Clock Rising Edge	10	150	ns

90C32 ARCNET TRANSCEIVER

Introduction to ARCNET Chip Set, 90C26 and 90C32

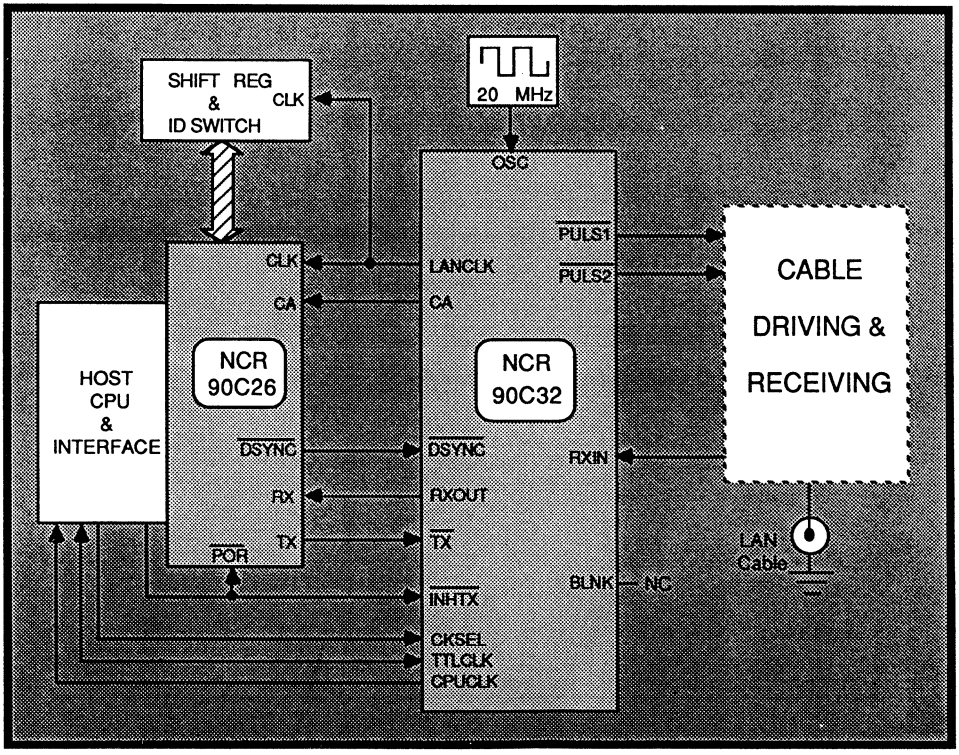
Arcnet is a registered trade mark of Datapoint Corporation. The NCR 90C26 AC (ARCNET Controller) and the NCR 90C32 AT (ARCNET Transceiver) together comprise all logical function of an ARCNET RIM (Resource Interface Module), in low power CMOS implementations. ARCNET is a popular token passing Local Area Network (LAN) scheme developed by Datapoint Corporation.

ARCNET is an ideal LAN for use in PC Networks, small office automation and factory automation.

Features

- Works with NCR 90C26 to transfer data at 2.5 Megabits per second
- Outputs the special clock signals required by the 90C26
- Generates output in NRZ format from incoming LAN data
- Provides two general purpose 4 MHz clock outputs for CPU interface
- Generates dual-pulse output signals for data transmission out to LAN

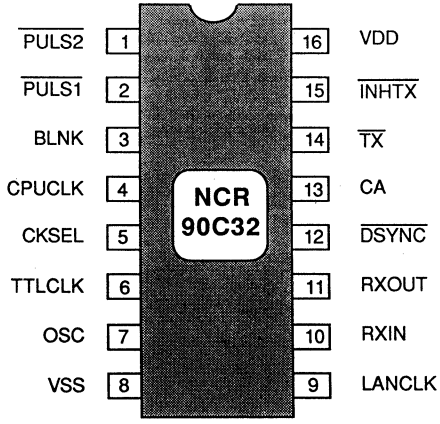
Typical 90C32 Hookup



General Description

The NCR 90C32 AT (ARCNET Transceiver) is a flexible data handling device that interfaces an NCR 90C26 ARCNET Controller chip to special cable driving and receiving circuitry. Directed by a host microprocessor, these components comprise a node in an ARCNET Local Area Network. The main function of the 90C32 is to convert data between a NRZ format on the hose side and dual-pulse on the cable side. In addition, the 90C32 provides special 5 MHz clocks required by the Controller chip and two other flexible 4 MHz clock signals that may be useful for the host and its interface. The only additional component required by the AT is a 20 MHz clock oscillator.

Pin Assignment



NCR 90C32 at Pin Descriptions

Symbol	Pin No.	I/O	Name and Function
OSC	7	I	OSCILLATOR: 20 MHz input signal.
CKSEL	5	I	CLOCK SELECT: Chooses the clock options for the extra TTLCLK and CPUCLK pins. When CKSEL is high, the 2 extra clocks both output a constant 4 MHz derived from the OSC signal. When CKSEL is low, then CPUCLK outputs the inverse of the input on TTLCLK.
LANCLK	9	O	NETWORK CLOCK: A free-running 5 MHz clock used for timing of 90C26 bus cycles and other functions.
CA	13	O	AUXILIARY CLOCK: A gated 5 MHz clock used to control data transfers of the 90C26. CA is periodically halted by $\overline{\text{DSYNC}}$ and restarted by RX.
TTLCLK	6	I/O	TTL CLOCK: May be either input or output. See description for CKSEL above.
CPUCLK	4	O	CPU CLOCK: An additional clock output. See description for CKSEL.
$\overline{\text{TX}}$	14	I	TRANSMIT DATA: Serial data input to the AT for conversion and transmission over the LAN.
$\overline{\text{PULS1}}$, $\overline{\text{PULS2}}$	2, 1	O	PULSE1 & PULSE2: These outputs are nonoverlapping negative going pulses that are asserted to the cable circuitry for every active pulse detected at $\overline{\text{TX}}$. PULSE1 is the first, and PULSE2 is the second pulse in this dual-pulse process.
RXIN	10	I	RECEIVE DATA IN: Serial input data from the LAN.
$\overline{\text{DYSNC}}$	12	I	DELAYED SYNC: This signal is asserted by the 90C26 to cause the AT to halt the CA clock.
RXOUT	11	O	RECEIVED DATA OUT: An NRZ form of the serial data input at the RXIN pin.
$\overline{\text{INHTX}}$	15	I	TRANSMIT INHIBIT: A Power-Reset input. $\overline{\text{INHTX}}$ active forces $\overline{\text{PULS1}}$ and $\overline{\text{PULS2}}$ high and BLNK low. These actions block the $\overline{\text{TX}}$ input from causing transmissions on the LAN.
BLNK	3	O	BLANK: A delayed and shortened version of $\overline{\text{TX}}$. This signal should be left unconnected when used in a typical circuit such as that on page 1.
VDD	16		+5 VOLT SUPPLY.
VSS	8		GROUND.

Description of Operation

The primary job of the NCR 90C32 AT is to format data and synchronize transfers over the ARCNET node to which it is attached. The LANCLK output to the 90C26 Controller device assures that all transactions between the 2 chips are synchronous. The analog cable circuit converts all dipulses (logic ones) received on the LAN to single positive pulses on the RXIN pin into the AT. This data, after some delay, is converted into NRZ format and sent to the 90C26 device. After a full byte has been transferred, the 90C26 will assert $\overline{\text{DSYNC}}$, which halts the CA clock. When the AT detects the first bit of the next byte, it will restart CA. The CA signal is used by the 90C26 to sample the levels on RXOUT at the times indicated on page 7. For outward transmissions, the 90C26 puts active-low pulses 200 ns in duration on $\overline{\text{TX}}$ for every logic one it sends. Each of these active-low pulses is converted to two 100 ns-wide nonoverlapping active-low pulses on the $\overline{\text{PULS1}}$ & $\overline{\text{PULS2}}$ outputs. These signals will drive the cable circuitry to create the dipulse that signifies logic one on the LAN cable itself.

Electrical Specifications

Absolute Maximum Stress Ratings

Voltage on VDD pin with respect to Vss	−0.3 to +7.0V
Voltage on any signal pin	−0.3 to V_{DD} + 0.3
Power Dissipation	53 mW
Operating Temperature Range	0 °C to 70 °C
Storage Temperature Range	−65 °C to +150 °C

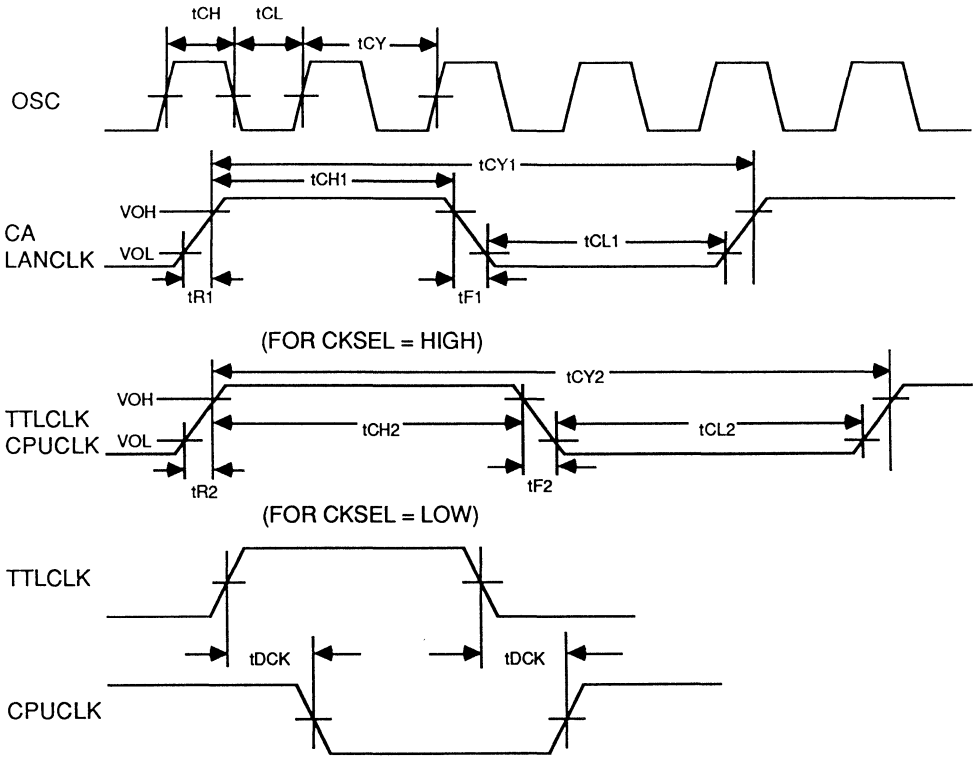
The values listed above are absolute maximums which, if exceeded, could cause permanent damage to the device. Voltages are with respect to circuit ground.

DC Characteristics

Symbol	Parameter	Min	Max	Units
V_{IL}	Input Voltage, Low (all inputs)	-0.3	0.8	V
V_{IH}	Input Voltage , High (all inputs)	2.7	V_{DD}+0.3	V
I_{OHX}	Transfer Output High Current (@ VOH min = 4.0V) for: PULS1, PULS2, RXOUT, and TTLCLK outputs		- 400	µA
I_{OLX}	Transfer Output Low Current (@ VOL max = 0.4V) for: PULS1, PULS2, RXOUT, and TTLCLK outputs		+ 4.0	mA
I_{OHK}	Clock Output High Current (@ VOH min = VDD - 0.5) for: CA, LANCLK, and CPUCLK		- 100	µA
I_{OLK}	Clock Output Low Current (@ VOL max = 0.4V) for: CA and LANCLK		+ 400	µA
I_{OLU}	Output Low Current (@ VOL max = 0.4V) for: CPUCLK		+ 100	µA
C_{IN}	Input Capacitance		30	pF
I_{IT}	Input Leakage Current (TTLCLK pin as Input)		±50	µA
I_{I}	Input Leakage Current (all other inputs)		±10	µA
I_{DD}	Power Supply Current (V_{DD} = 5.25 V)		10	mA

Notes: All parameters are assured over the 0°C to 70°C temperature range and with 4.75 ≤V_{DD}≤ 5.25 Volts. All voltages are with respect to circuit ground (Vss).

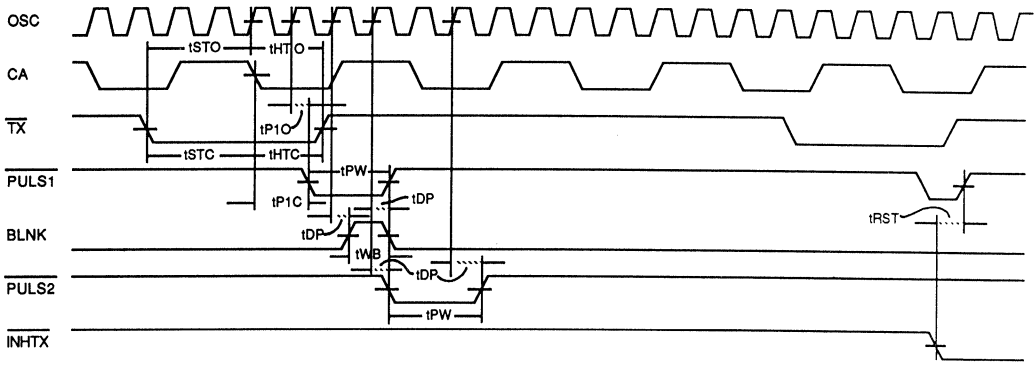
Clock Timing Parameters



NAME	DESCRIPTION	MIN	TYP	MAX	UNITS
tCH	High Duration of OSC	20			ns
tCL	Low Duration of OSC	20			ns
tCY	Period of OSC		50		ns
tCH1	High Duration of 90C26 Clocks	75			ns
tCL1	Low Duration of 90C26 Clocks	75			ns
tCY1	Period of 90C26 Clocks		200		ns
tR1	Rise Time of 90C26 Clocks			20	ns
tF1	Fall Time of 90C26 Clocks			20	ns
tCH2	High Duration of Extra Clocks	110			ns
tCL2	Low Duration of Extra Clocks	110			ns
tCY2	Period of Extra Clocks		250		ns
tR2	Rise Time of Extra Clocks			30	ns
tF2	Fall Time of Extra Clocks			30	ns
tDCK	CPUCLK Delay from TTLCLK Edges			45	ns

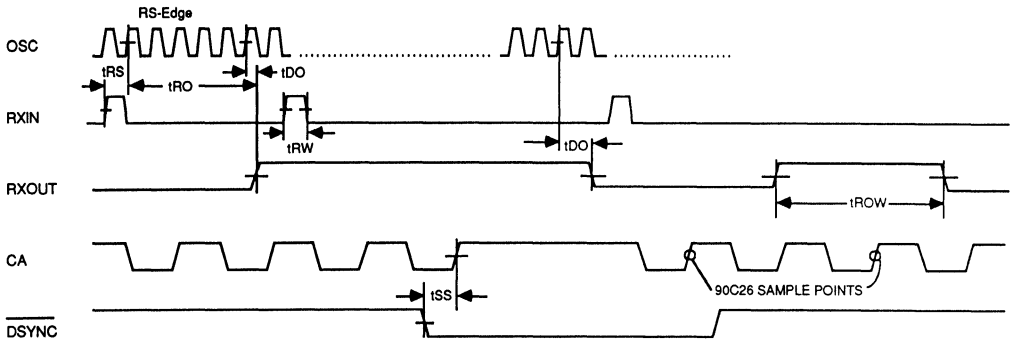
NOTE: All timings in this specification are taken from the 10% & 90% points with respect to the specified VOL & VOH of the waveforms.

Transmit Timing



NAME	DESCRIPTION	MIN	TYP	MAX	UNITS	NOTES
tSTC	$\overline{TX}$ Setup to CA Falling Edge		10		ns	
tSTO	$\overline{TX}$ Setup to OSC Rising Edge	10			ns	
tHTC	$\overline{TX}$ Hold after CA Falling Edge		10		ns	
tHTO	$\overline{TX}$ Hold after OSC Rising Edge	10			ns	
tP1C	$\overline{PULS1}$ Delay after CA Falling Edge	60			ns	
tP1O	PULS1 Delay after OSC Rising Edge			60	ns	
tDP	PULSE & BLNK Edge Delays after OSC Rising Edge			60	ns	
tPW	Duration of the two PULSE Signals		2 tCY		ns	tCY was defined
tWB	Pulse Duration for BLNK		tCY		ns	on previous page.
tRST	PULS1 Turnoff from INHTX Falling Edge			40	ns	

Receive Timing



NAME	DESCRIPTION	MIN	TYP	MAX	UNITS	NOTES
tRS	RXIN Setup to RS-Edge	10			ns	
tRW	RXIN Active Duration	10			ns	
tDO	RXOUT Delay from OSC Rising Edge			70	ns	
tRO	RXOUT Active after RS-Edge		5 tCY + tDO		ns	tCY was defined
tSS	CA Turnoff after DSYNC	10	20		ns	on page 5
tROW	RXOUT Isolated '1' Duration		400		ns	

20C12 HART™ MODEM

Introduction

HART is a trade mark of Rosemount Corporation.

The NCR 20C12 (Highway Addressable Remote Transducer) HART low power FSK modem is a single chip CMOS model that operates at the Bell 202 standards.

The NCR 20C12 HART modem is ideally suited for factory automation, especially in areas that require ultra low power such as intrinsically safe environments.

Features

- Operates at the Bell 202 Standard Forward Bit Rate of 1200 bits/sec
- Uses the Bell 202 Shift Frequencies of 1200 Hz and 2200 Hz
- Transmit Modulation of 1200 Baud
- Receive Modulation of 1200 Baud
- Single Chip Frequency -Shift-Keying (FSK)
- Optimal for Intrinsically Safe Applications
- Singel 5-V Power Supply
- CMOS and TTL Compatible
- Reliable CMOS Technology
- 16 pin DIP or 28-pin PLCC packages available

General Description

The NCR 20C12 (Highway Addressable Remote Transducer) HART low power FSK Modem is a single-chip, CMOS modem for use in process control instruments or other low power equipment. The modem circuitry is digital. It provides only the modulating and demodulating functions and is intended to be used with external circuits which may amplify, filter, and shape the media signals.

The modem operates at the Bell-202 standard upper (forward) bit rate of 1200 bits/sec and uses the Bell-202 shaft frequencies of nominally 1200 Hz and 2200 Hz. With proper conditioning of them media, it can communicate with other Bell-202 modems.

To conserve power, the demodulator is stopped while the modulator operates and vice versa. Therefore, operation is half-duplex. The modem is packaged in a standard 16-pin Dual In-line Package (DIP) or in a 28-pin Plastic Leaded Chip Carrier (PLCC). The HART modem operates from a single 5-volt supply and requires an externally generated clock fo 460.8 kHz.

The modem provides an active low carrier detect output and a 19.2 kHz clock output. No external adjustments or special biasing are required.

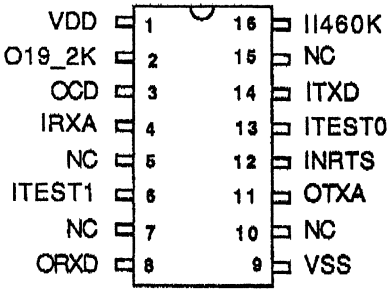
Pin Description

SIGNAL	INPUT or OUTPUT	PIN #		DESCRIPTION
		DIP	PLCC	
VDD	-	1	1	+ 5 V power supply
O19_2k	O	2	2	User clock. Nominally 19.2 kHz (16 times bit rate).
CCD	O	3	3	Carrier Detect is low (0) when carrier is present.
IRXA	I	4	4	Demodulator Input accepts 1200 (2200) Hz square wave FSK modulated carrier.
NC	-	5	5-12	No internal connection.
ITEST1	I	6	13	Test Input One must be connected to VSS during normal operation.
NC	-	7	-	No Internal Connection.
ORXD	O	8	14	Demodulator Output provides a logic 1 (0) in response to a 1200 (2200) Hz FSK square wave signal at IRXA. Demodulation takes place only when INRTS is a high (1). With INRTS low (0) the ORXD output is not defined.
VSS	-	9	15	Power supply ground.
NC	-	10	-	No Internal connection.
OTXA	O	11	16	Modulator Output provides a square wave FSK output 1200 (2200) Hz in response to a logic 1 (0) applied to ITXD. OTXA is active when INRTS is low (0). It goes to a high-impedance state when INRTS is high (1).
INRTS	I	12	17	Request To Send selects operation of the modulator when low (0) and the demodulator when high (1), causing OTXA to go to a high-impedance state when high (1).
ITEST0	I	13	26	Test Input Zero must be connected to VSS during normal operation.
ITXD	I	14	27	Modulator Input accepts input data (logic 1 or 0) for modulating the carrier output at OTXA.
NC	-	15	18-25	No Internal Connection.
I1460K	I	16	28	Input Clock clocks the modem circuitry; the frequency is nominally 460.8 kHz.

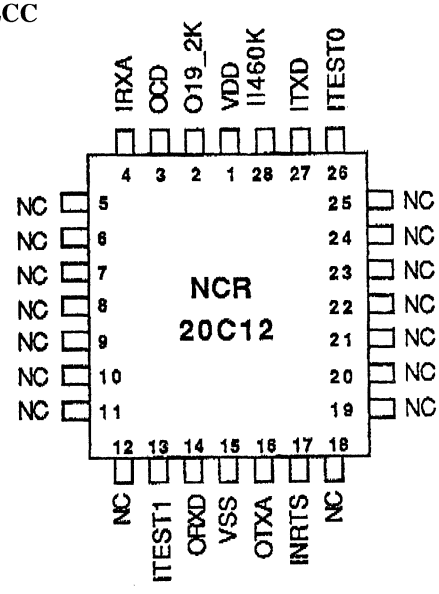
Note: Signal names starting with 'I' are inputs and signal names starting with 'O' are outputs.

Pin Out

16 Pin Dip



28 Pin PLCC



Electrical Characteristics

The conditions for the following electrical characteristics apply to the full operating temperature range with a supply voltage of 5.0 volts.

DC Characteristics

Absolute Maximum Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
TSTG	Storage Temperature	-65	150	°C
TA	Operating Free-air Temperature	-40	+85	°C
VDD	Supply Voltage (non-operating)		7	V

Cautions

1. CMOS Devices are damaged by high energy electrostatic discharge. Devices must be stored in conductive foam or with all pins shunted. Precautions should be taken to avoid application of voltages higher than the maximum rating.
2. Remove power before insertion or removal of this device.
3. Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operation conditions specified in this document is not implied.

Operating Conditions

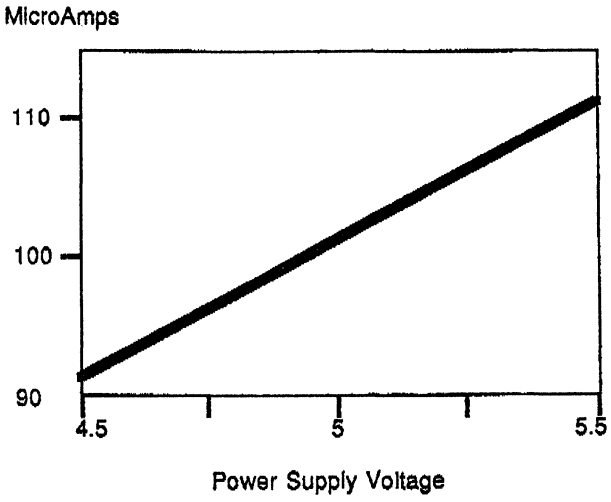
SYMBOL	PARAMETER	MIN	MAX	UNIT	CONDITIONS
VDD	Supply Voltage	4.5	5.5	V	
VIN	Input Voltage	VSS -0.3	VDD +0.3	V	
VIH	Logical 1 Input Voltage	2.0	VDD +0.3	V	
VIL	Logical 0 Input Voltage	VSS -0.3	0.8	V	
VOL	Output Low Voltage		0.4	V	Output Sink Current=4.0mA
VOH	Output High Voltage	2.4	VDD -0.3	V	Output Source Current=4.0mA
IDD*	Supply Current		437	µA	During Demodulation
			101	µA	During Modulation
IIN	Input Leakage Current		+/-1	µA	VIN=VDD or VSS
ITO	High-Impedance Output Leakage Current		+/- 10 µA typical		Applies only to OTXA
CIN	Input Capacitance		10pF typical		

* **IDD NOTES:**

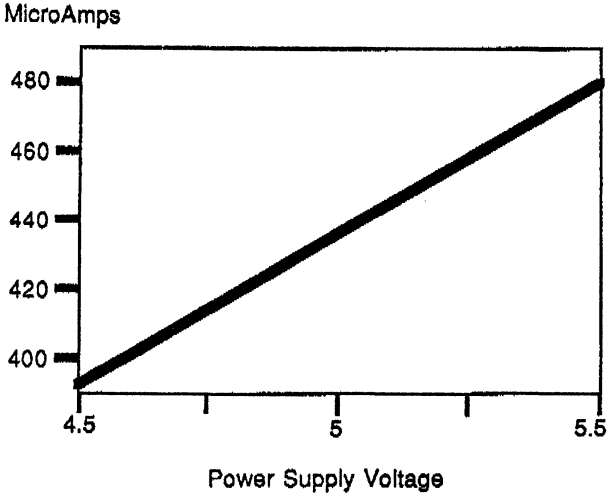
- 50pF load on all outputs
- All input rise and fall time requirements satisfied
- Clock (II460K) freq=460.8kHz +/-0.1%
- See the following Current vs Voltage Charts for more information.

Current vs Voltage Charts

Modulation



Demodulation



AC Characteristics

Clock Frequency (at the II460K pin):	460.8 kHz +/- 0.1%
Clock Pulse Widths:	high level = 150 nanoseconds min. low level = 150 nanoseconds min.
Clock Rise and Fall Time:	10 nanoseconds max.
IRXA Input Rise and Fall Time:	200 nanoseconds max.
Other Inputs Rise and Fall Time:	50 nanoseconds max.
Output Rise and Fall Time*:	10 nanoseconds max.

*Condition: 50 pF capacitance on all outputs.

Modem Characteristics

Carrier Detect

Carrier Detect Frequency Range: This is the range of frequencies applied at IRXA over which OCD must go low (logic 0). Conditions: 1) Clock (II460K) frequency of 460.8 kHz +/- 0.1%. 2) Input (IRXA) asymmetry of 5% max.	1000 Hz to 2575 Hz.
Time From Carrier Input To Carrier Detect: This is the time from the start of a valid carrier signal at IRXA until OCD goes to logic low (0).	Minimum = 9.0 millisec. Maximum = 15.0 millisec.
Time From Carrier Loss To Carrier Undetect: This is the time from the loss of a valid carrier signal at IRXA until OCD goes to a logic high (1).	1 millisecond maximum
User Clock (O19_2K) Frequency: This clock frequency is proportional to the 460.8kHz clock at II460K.	19.2 kHz nominal.

Modulator

Modulator Output Frequency (at the OTXA pin):
Nominal high frequency (logic 0 applied to ITXD) = 2194.3 Hz
Nominal low frequency (logic 1 applied to ITXD) =1196.9 Hz
The modulator output frequencies are proportional to the input clock frequency (applied to the II460K pin).

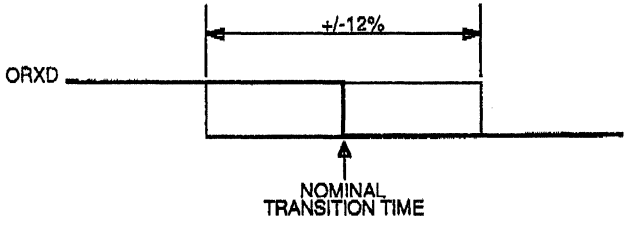
Modulator phase continuity error: +/- 10 degrees maximum.

Demodulator

Maximum Demodulator Jitter: +/- 12% of one bit time

Conditions:
1) Input frequencies at 1200 Hz +/- 10 Hz and 2200 Hz +/- 20 Hz.
2) Clock (II460K) frequency of 460.8 kHz +/- 0.1 %.
3) Input (IRXA) asymmetry = 0.

Representation of Jitter:



Functional Description

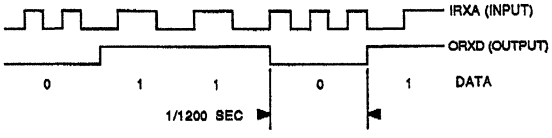
The modem uses shift frequencies of nominally 1200 Hz (representing a digital one) and 2200 Hz (digital zero). The bit rate is nominally 1200 bit/second. There are four major functional blocks: clocks, demodulator, modulator and carrier detect.

Clocks

The clock section accepts a digital input frequency of 460.8 kHz from an external source. This is used in various counter arrangements to generate several internal clocks. One internal clock at a nominal frequency of 19.2 kHz is also brought to an output pin for use in reconstructing the data in circuits external to the modem. All circuit operations can be performed with a master clock frequency of 460.8 kHz. This is considerably lower than the clock frequencies used in existing single-chip modems and contributes to a low current consumption. Various sections of the model are shut down (not clocked) when not being used. This further reduces power consumption.

Demodulator

The demodulator accepts an FSK signal at its IRXA input and reproduces the original modulating signal at its ORXD output. Both the input and output are digital signals as illustrated below.



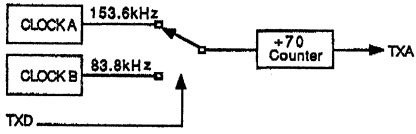
Demodulator Signals

Modulator

The modulator accepts digital data in NRZ form at its ITXD input and generates the FSK modulated signal at its OTXA output. An additional requirement for the modulator is that it provide phase continuous modulation. That is, when switching from one to the other of the shift frequencies, the phase angle of the modulated signal is preserved.

A simplified modulator block diagram is shown below. It works by counting down one or the other of two clocks, A or B, depending on whether the input signal is high or low. Clock A (153.6 kHz) is derived by dividing the 460.8 kHz oscillator frequency by 3. Clock B (83.78) kHz is derived by dividing the 460.8 kHz oscillator frequency by 5.5. The counter in the diagram is a divide-by-70. This means that the two shift frequencies are 1196.9 Hz and 2194.3 Hz.

Since the modulator works by counting down clocks, its output transitions are quantized as to when they can occur. The maximum timing error due to quantization is 12 microseconds, which is the period of the slower of the two clocks, A and B. At the higher of the two shift frequencies (2200 Hz), the phase error is 9.5 degrees, and at the lower of the two shift frequencies the phase error is 5.2 degrees.



Modulator Block Diagram

Carrier Detect

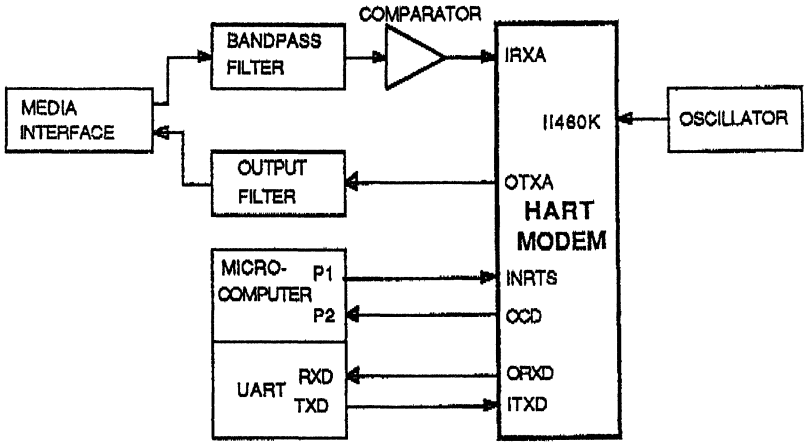
The active low carrier detect works by looking for incoming signal frequencies in the approximate range of 1200 Hz to 2200 Hz. This is done by measuring interval times between input transitions. If several successive interval time measurements fall within specific limits, carrier detect is indicated, and the carrier detect output is asserted. If, after the carrier has been detected, the measured interval time falls outside of these limits for some length of time, the carrier detect output is unasserted.

Typical Application

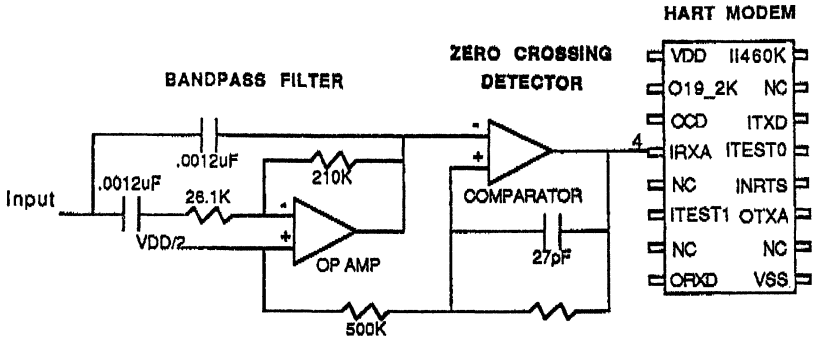
The modem will usually be applied as shown in the block diagram below. The received signal is filtered to reduce noise. The filter should include the band of frequencies from about 900 Hz to 2.5 kHz. The output of the filter is applied to the input of a comparator to provide the required signalling levels to the modem. Additional amplification may be required ahead of the filter to provide sufficient signal to the comparator. Be sure that the comparator switches at the zero crossings of the incoming waveform and is glitch free.

Depending on the application, the output filter may not be necessary. Omitting it will cause square waves to be applied to the signalling media.

A schematic showing a typical applications follows the block diagram.



Block Diagram of a Typical Application

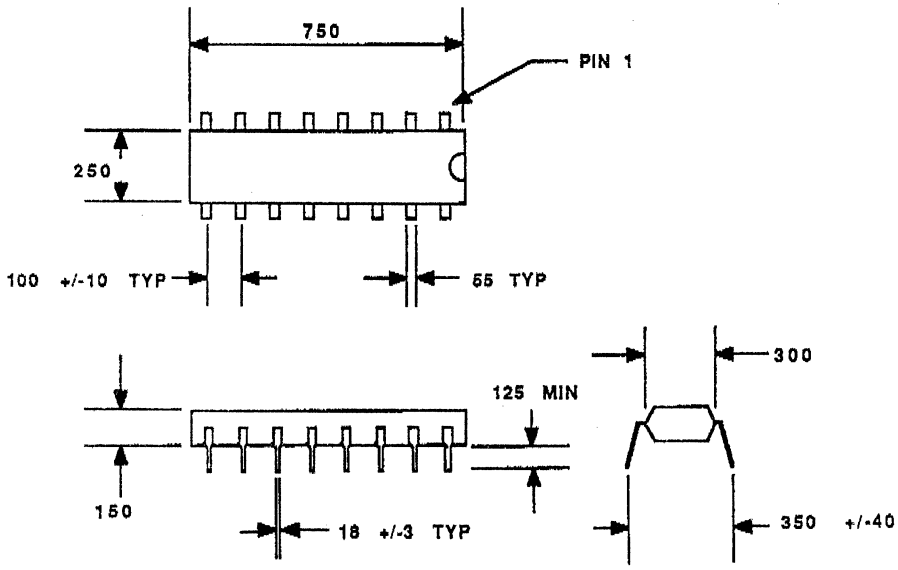


Schematic of a Typical Application

Mechanical Drawings

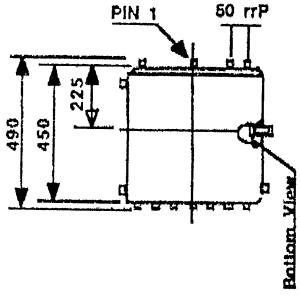
16-Pin Dual In-line Package (DIP)

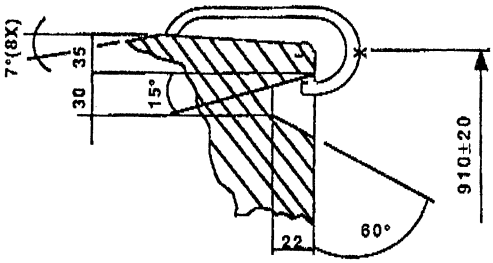
Note: All units are 1/1000 inches



28-Pin Plastic Leaded Chip Carrier (PLCC) Package

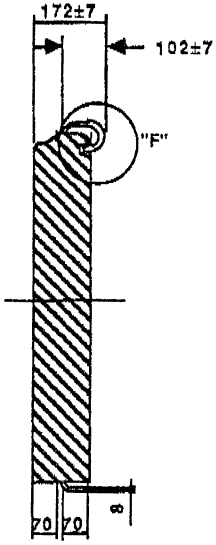
Note: All units are 1/1000 inches

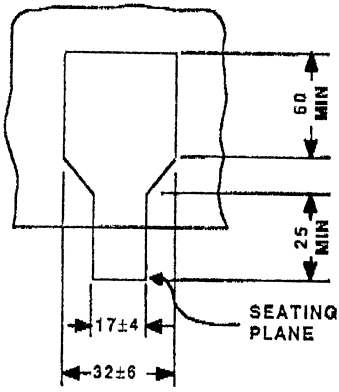




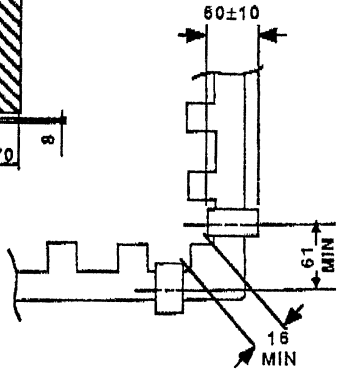
DETAIL OF "F"

X: SURFACE MOUNT POINT





SIDE VIEW



BOTTOM VIEW

COMMUNICATIONS PRODUCTS

GEOMETRIC ARITHMETIC PARALLEL PROCESSOR

Introduction

Announced in 1984, the Geometric Arithmetic Parallel Processor, or GAPP™, chip is the first commercially available Single-Instruction/Multiple Data class parallel processor integrated circuit. The GAPP chip is a general purpose processor that can be used in a wide variety of applications such as: Pattern Recognition, Automated Inspection, Machine Vision, Image Processing, and Morphological Processing.

Each GAPP chip contains 72 processing elements arranged in a 6 by 12 matrix. All 72 processing elements concurrently execute instructions on their respective data elements, thus a single GAPP chip processes 72 data items simultaneously. Each processing element in the chip can communicate with its immediate neighbor processing elements, a very desirable feature for performing neighborhood computational algorithms typical in image processing.

GAPP chips can additionally be cascaded into arbitrarily sized processing element arrays in a fashion that makes chip boundaries completely transparent to the programmer. For instance, two GAPP chips can be connected to form a 12 by 12 array or a 24 by 6 array or processing elements. Regardless of the configuration, with two GAPP chips you have the power of 144 individual processing elements operating in parallel. As devices are cascaded to form larger arrays of processing elements, the performance of a GAPP-based system provides unmatched processing power.

Performance of GAPP Systems

Calculation	Execution Time (per Image Plane with 10MHz Clock)	Pixels per Second Processed		
		6x12 Array (1 GAPP)	48 x 48 Array (32 GAPPs)	192 x 192 Array (512 GAPPs)
8-Bit Add	2.5μS	28.8M	921M	14.7B
8-Bit Multiply	27.1μS	2.6M	124M	1.3B
8-Bit 3x3 Gauss Filter*	71.7μS	557K	30M	503M
8-Bit 3x3 Convolution*	100μS (Typical)	400K	22M	361M
8-Bit 5x5 Convolution*	273μS (Typical)	58K	7M	129M
Binary 21x21 Correlation*	1.1mS	—	712K**	27M
Edge Detection (Sobel Gradient)	58.1μS	688K	36M	621M

*Efficiency improved if edge buffers used.
**2M if edge buffers are used.

GAPP Development Systems

In order to allow engineers, researchers, and programmers to rapidly develop application software for GAPP™ systems, NCR Microelectronics offers three development systems for the GAPP.

The entry level GAPP support product is the GAPP PC Development System (NCR45GDS1). This product consists of a board which plugs into an IBM, or compatible, personal computer and software to develop and debug GAPP programs. The 12 by 12 array of GAPP processing elements provided by the GAPP PC Development System allows customers to easily evaluate GAPP technology without investing a great deal of time or money.

The mid-level GAPP support product is the GAPP System Simulator Package (NCR45GS4). This product consists of software that allows users to develop GAPP programs and then to execute the programs on a simulator. The simulator supports array sizes of up to 108 by 96 processing elements. The GAPP System Simulator Package runs on PC-AT class computers under DOS or on a VAX computers under VMS.

At the high-end of GAPP application software development products, NCR offers the SIMD Processor Development Station (NCR45SPDS). This product includes a processing electronics enclosure, incorporating an array of GAPP processing chips and a 10 megahertz controller, that connects to a PC-AT class computer via a high-speed host interface bus. A multiwindow, menu-driven user interface provides the user with complete control over all features of the NCR45SPDS hardware and provides an optimal environment for writing and debugging GAPP software.

All three GAPP development products include the NCR-GAL™ (GAPP Algorithm Language) compiler and a library of source code routines written for the GAPP in NCR-GAL. NCR-GAL is a subset of the "C" programming language with built in GAPP mnemonics. Since all GAPP development products use NCR-GAL for source code, an applications written on any one of the GAPP development products may be ported to the other two.

45CG72 GEOMETRIC ARITHMETIC PARALLEL PROCESSOR (GAPP)™

Applications

- Pattern Recognition
  - — Correlation
  - — Sobel Transform
  - — Spoke Filter
  - — Template Matching
  - — Automated Inspection
  - — Machine Vision
- Parallel Data Processing
  - — Convolution
  - — Search and Sort
- Image Processing
  - — Image Enhancement
  - — Edge Detection
  - — Two Dimensional Convolution
  - — Compression
  - — Spatial Filtering
  - — Differential Imaging
- Associative Processor
  - — Content Addressable Memory
  - — Limit Search
  - — Hamming Distance

General Description

The NCR45CG72 is a two-dimensional parallel array processor chip. It is a mesh-connected six by twelve arrangement of 1 bit processor elements. Each processor element can communicate with four neighbors: N,E,S, and W. Each processor element is composed of a bit serial ALU, 128 × 1 bit RAM and 4 single bit latches: three latches hold inputs to the ALU and the fourth latch allows I/O through the cell without interrupting the ALU, i.e. I/O operations are overlapped with computation.

The cascadability of the GAPP allows system designers to implement arrays of processors of arbitrary size in multiples of 6 × 12 elements.

Features

- CMOS parallel array with 72 processors per chip
- 6 × 12 array of bit serial processor elements
- Single instruction multiple data stream architecture — all processor elements operate in parallel
- GAPP devices are fully cascadable
- System throughput increases linearly with number of processor elements in the system
- Broadcast global input and output
- Separate I/O bux = overlapped I/O and computation
- 128 Bits of static RAM per processor
- VLSI double layer metal CMOS technology
- 500 milliwatts power at 10 Mhz

Absolute Maximum Ratings

Supply Voltage, V_{DD} ..... + 7V
Voltage on any pin with respect to ground ..... −0.3 to V_{DD} + 0.3V
Storage temperature ..... −65 °C to 150 °C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operation conditions stipulated elsewhere in this specifications is not implied.

CAUTION

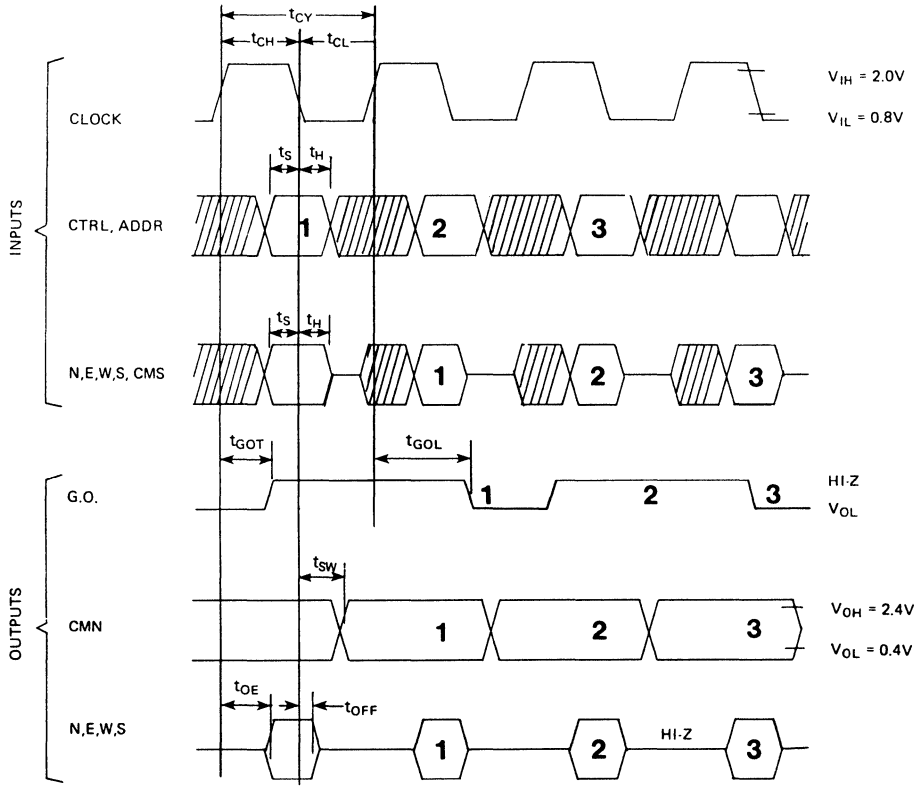
1. CMOS Devices are damaged by high energy electrostatic discharge. Devices must be stored in conductive foam or with all pins shunted. Precautions should be taken to avoid application of voltages higher than the maximum rating.

2. Remove power before insertion or removal of this device.

Operating Characteristics

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	4.5	5.0	5.5	V
Supply Current (10 pF loads) 45CG72-1	I_{DD}	–		100	mA
Input Low Voltage	V_{IL}	0.0		0.8	V
Input High Voltage	V_{IH}	2.0		V_{DD}	V
Output Low Voltage (I_{OL} = 2 mA)	V_{OL}	–		0.4	V
Output High Voltage (I_{OH} = 1 mA)	V_{OH}	2.4		–	V
Temperature	T_A	0		70	°C
Input Capacitance	C_{IN}	–		8	pF
Output Capacitance	C_O	–		8	pF
Leakage Current on any Input or I/O Pin	I_{IN}	–		10	μA

Timing Diagram



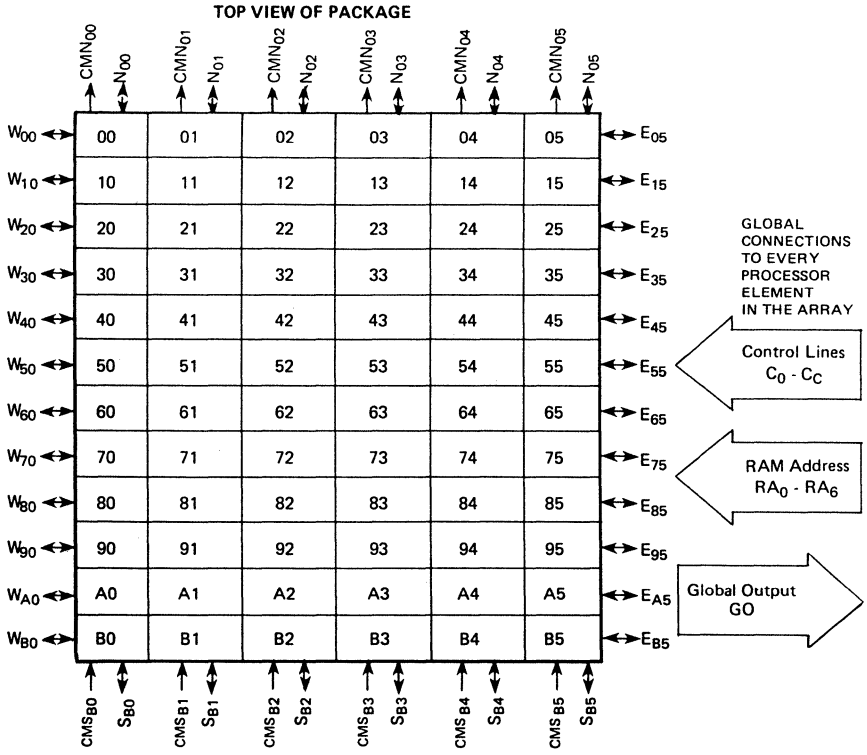
NOTE: 1,2,3 refer to the staging sequence of instruction, data in and data out.

AC Characteristics

PARAMETER	SYMBOL	NCR45CG72-1		
		MIN	MAX	UNITS
CYCLE TIME	t_{CY}	100		ns
CLOCK LOW	t_{CL}	50	5000	ns
CLOCK HIGH	t_{CH}	50	5000(1)	ns
SETUP TIME	t_S	10		ns
INPUT HOLD TIME	t_H	10		ns
OUTPUTS ENABLED (2)	t_{OE}	10	25	ns
OUTPUT HOLD TIME	t_{OFF}	10	30	ns
GLOBAL OUTPUT LOW (2)	t_{GOL}	10	160	ns
GLOBAL OUTPUT TRISTATE	t_{GOT}	10	35	ns
CMN OUTPUT (2)	t_{SW}	10	55	ns

NOTE: (1) d.c. by design; tested at 5 μsec.
(2) tested with 22pF load. Add 0.06 nS/pF for additional load.

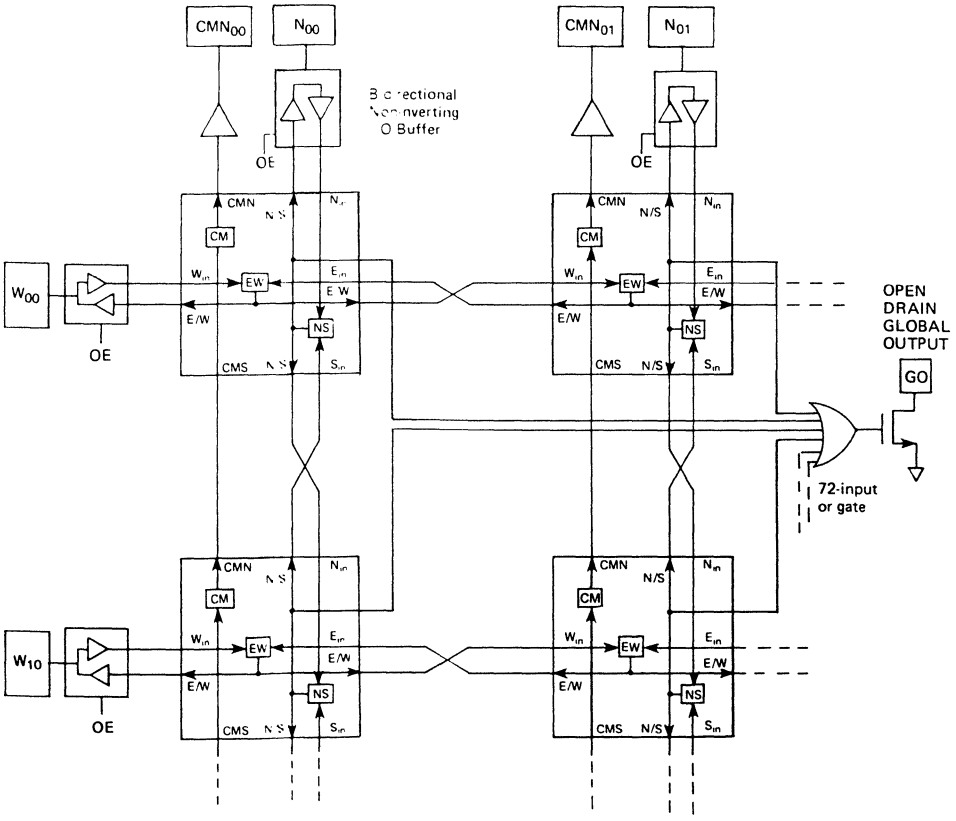
Processor Element and Data Bus Identification



NOTE: This numbering scheme may be extended in systems which contain more than one GAPP device.

PIN LABELS	
$W_{00} - W_{B0}$	WEST DATA BUS
$E_{05} - E_{B5}$	EAST DATA BUS
$N_{00} - N_{05}$	NORTH DATA BUS
$S_{B0} - S_{B5}$	SOUTH DATA BUS
$CMS_{B0} - CMS_{B5}$	INPUT BUS
$CMN_{00} - CMN_{05}$	OUTPUT BUS
$RA_0 - RA_6$	RAM ADDRESS BUS
$C_0 - C_C$	CONTROL LINES – INSTRUCTION BUS – GLOBAL DATA INPUT BUS
GO	GLOBAL OUTPUT LINE

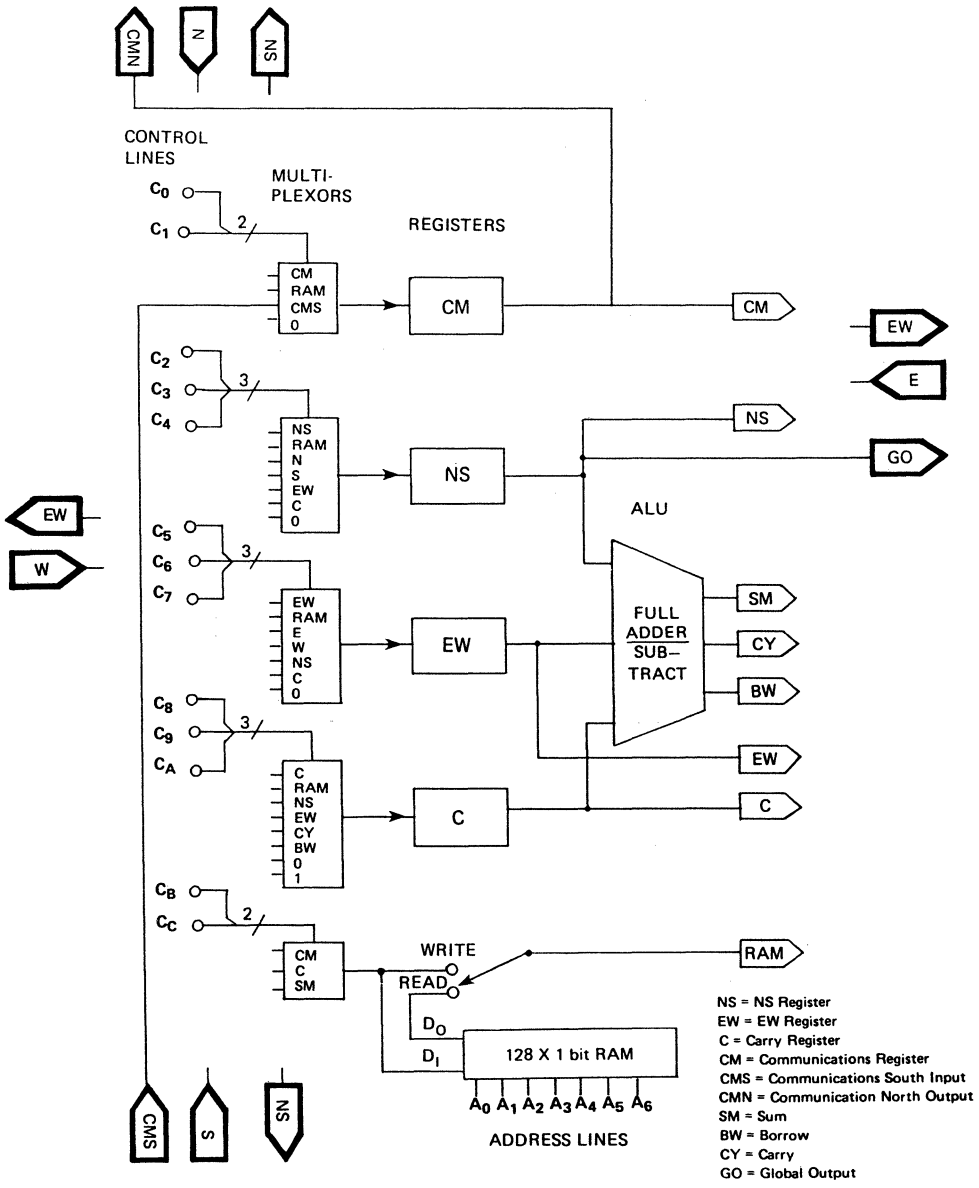
Block Diagram of Connections Between Four Processor Elements



OE = Output Enable is an internal connection
East Outputs enabled whenever EW:=W
West Outputs enabled whenever EW:=E
North Outputs enabled whenever NS:=S
South Outputs enabled whenever NS:=N
GO is pulled low whenever any NS register contains 1

COMMUNICATIONS PRODUCTS

Schematic Diagram of One Processor Element



Instruction Set

Register Operation	Mnemonic	Control Lines C_C	C_B	C_A	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	Description
CM	CM: = CM	X	X	X	X	X	X	X	X	X	X	X	0	0	MICRO-NOP
	CM: = RAM	X	X	X	X	X	X	X	X	X	X	X	0	1	LOAD CM FROM RAM
	CM: = CMS	X	X	X	X	X	X	X	X	X	X	X	1	0	MOVE FROM CMS INTO CM
	CM: = 0	X	X	X	X	X	X	X	X	X	X	X	1	1	LOAD Ø INTO CM
NS	NS: = NS	X	X	X	X	X	X	X	X	0	0	0	X	X	MICRO-NOP
	NS: = RAM	X	X	X	X	X	X	X	X	0	0	1	X	X	LOAD NS FROM RAM
	NS: = N	X	X	X	X	X	X	X	X	0	1	0	X	X	MOVE FROM N INTO NS
	NS: = S	X	X	X	X	X	X	X	X	0	1	1	X	X	MOVE FROM S INTO NS
	NS: = EW	X	X	X	X	X	X	X	X	1	0	0	X	X	MOVE FROM EW INTO NS
	NS: = C	X	X	X	X	X	X	X	X	1	0	1	X	X	MOVE FROM C INTO NS
	NS: = 0	X	X	X	X	X	X	X	X	1	1	0	X	X	LOAD Ø INTO NS
EW	EW: = EW	X	X	X	X	X	0	0	0	X	X	X	X	X	MICRO-NOP
	EW: = RAM	X	X	X	X	X	0	0	1	X	X	X	X	X	LOAD EW FROM RAM
	EW: = E	X	X	X	X	X	0	1	0	X	X	X	X	X	MOVE FROM E INTO EW
	EW: = W	X	X	X	X	X	0	1	1	X	X	X	X	X	MOVE FROM W INTO EW
	EW: = NS	X	X	X	X	X	1	0	0	X	X	X	X	X	MOVE FROM NS INTO EW
	EW: = C	X	X	X	X	X	1	0	1	X	X	X	X	X	MOVE FROM C INTO EW
	EW: = 0	X	X	X	X	X	1	1	0	X	X	X	X	X	LOAD Ø INTO EW
C	C: = C	X	X	0	0	0	X	X	X	X	X	X	X	X	MICRO-NOP
	C: = RAM	X	X	0	0	1	X	X	X	X	X	X	X	X	LOAD C FROM RAM
	C: = NS	X	X	0	1	0	X	X	X	X	X	X	X	X	MOVE FROM NS INTO C
	C: = EW	X	X	0	1	1	X	X	X	X	X	X	X	X	MOVE FROM EW INTO C
	C: = CY	X	X	1	0	0	X	X	X	X	X	X	X	X	LOAD C FROM CARRY
	C: = BW	X	X	1	0	1	X	X	X	X	X	X	X	X	LOAD C FROM BORROW
	C: = 0	X	X	1	1	0	X	X	X	X	X	X	X	X	LOAD Ø INTO C
	C: = 1	X	X	1	1	1	X	X	X	X	X	X	X	X	LOAD 1 INTO C
RAM	READ	0	0	X	X	X	X	X	X	X	X	X	X	X	READ FROM RAM
	RAM: = CM	0	1	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM CM
	RAM: = C	1	0	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM C
	RAM: = SM	1	1	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM SUM

Arithmetic Operations

Adder/Subtracter Operations

INPUT			OUTPUT		
NS	EW	C	SM	CY	BW
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	0
1	1	0	0	1	0
0	0	1	1	0	1
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	1	1	1

Logic Operations

LOGICAL OPERATION	DESCRIPTION	CONDITIONS
INV	$SM = \overline{NS}$	EW = 0, C = 1
	$SM = \overline{EW}$	NS = 0, C = 1
	$SM = \overline{C}$	NS = 0, EW = 1
AND	$CY = NS \bullet EW$	C = 0
	$CY = EW \bullet C$	NS = 0
	$CY = NS \bullet C$	EW = 0
	$BW = \overline{NS} \bullet EW$	C = 0
OR	CY = NS + EW	C = 1
	$BW = \overline{NS} + EW$	C = 1
	BW = EW + C	NS = 0
XOR	$SM = NS \oplus C$	EW = 0
	$SM = NS \oplus EW$	C = 0
	$SM = EW \oplus C$	NS = 0
XNOR	$SM = \overline{NS \oplus EW}$	C = 1

COMMUNICATIONS PRODUCTS

NCR45CG72 Pin Numbers vs. Signal Labels (Plastic Chip Carrier)

PIN NO.	LABEL	PIN NO.	LABEL	PIN NO.	LABEL
1	NC	29	CMS_{B5}	57	CMN_{05}
2	NC	30	C_8	58	N_{05}
3	NC	31	C_9	59	CMN_{04}
4	W_{60}	32	C_A	60	N_{04}
5	W_{70}	33	C_B	61	CMN_{03}
6	W_{80}	34	C_C	62	N_{03}
7	W_{90}	35	E_{B5}	63	CLK
8	W_{A0}	36	E_{A5}	64	VSS
9	W_{B0}	37	E_{95}	65	RA_5
10	C_2	38	E_{85}	66	RA_3
11	C_3	39	E_{75}	67	RA_1
12	C_4	40	E_{65}	68	CMN_{02}
13	S_{B0}	41	VDD	69	N_{02}
14	CMS_{B0}	42	NC	70	CMN_{01}
15	S_{B1}	43	NC	71	N_{01}
16	CMS_{B1}	44	NC	72	CMN_{00}
17	S_{B2}	45	VDD	73	N_{00}
18	CMS_{B2}	46	E_{55}	74	C_1
19	RA_0	47	E_{45}	75	C_0
20	RA_2	48	E_{35}	76	W_{00}
21	RA_4	49	E_{25}	77	W_{10}
22	VSS	50	E_{15}	78	W_{20}
23	RA_6	51	E_{05}	79	W_{30}
24	S_{B3}	52	C_7	80	W_{40}
25	CMS_{B3}	53	$VDD_{(sub)}$	81	W_{50}
26	S_{B4}	54	GO	82	VDD
27	CMS_{B4}	55	C_6	83	NC
28	S_{B5}	56	C_5	84	NC

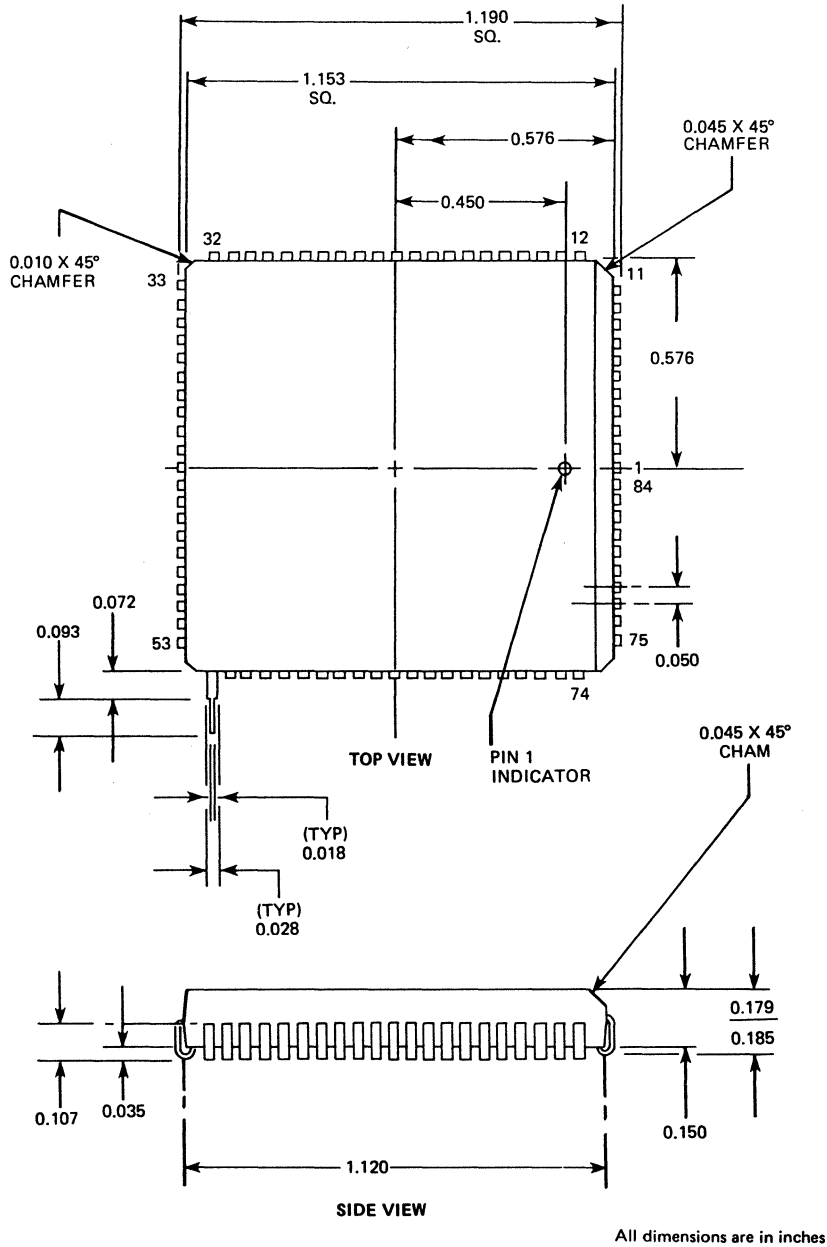
NC = No Connection

Table of Signal Names vs. Pin Numbers

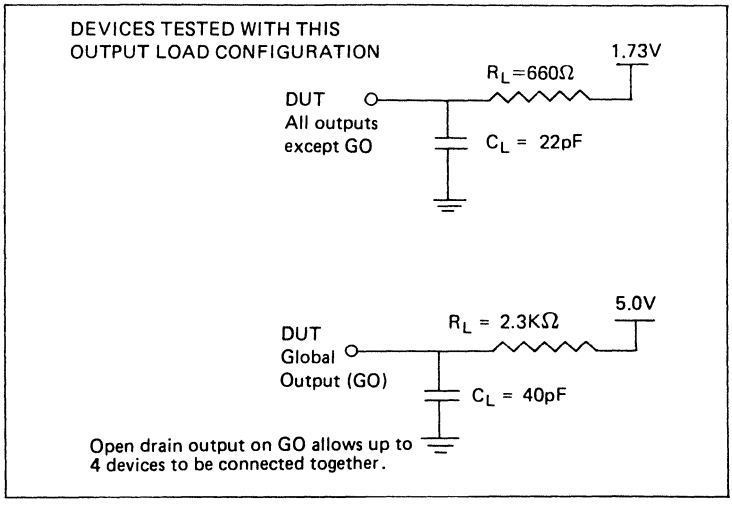
SIGNAL NAMES	PLA PIN	SIGNAL NAMES	PLA PIN	SIGNAL NAMES	PLA PIN
CMS_{B0}	14	N_{04}	60	C_7	52
CMS_{B1}	16	N_{05}	58	C_8	30
CMS_{B2}	18	S_{B0}	13	C_9	31
CMS_{B3}	25	S_{B1}	15	C_A	32
CMS_{B4}	27	S_{B2}	17	C_B	33
CMS_{B5}	29	S_{B3}	24	C_C	34
CMN_{00}	72	S_{B4}	26	RA_0	19
CMN_{01}	70	S_{B5}	28	RA_1	67
CMN_{02}	68	W_{00}	76	RA_2	20
CMN_{03}	61	W_{10}	77	RA_3	66
CMN_{04}	59	W_{20}	78	RA_4	21
CMN_{05}	57	W_{30}	79	RA_5	65
E_{05}	51	W_{40}	80	RA_6	23
E_{15}	50	W_{50}	81	GO	54
E_{25}	49	W_{60}	4	VSS	22
E_{35}	48	W_{70}	5	VSS	64
E_{45}	47	W_{80}	6	VDD	82
E_{55}	46	W_{90}	7	VDD	41
E_{65}	40	W_{A0}	8	VDD	45
E_{75}	39	W_{B0}	9	VDD *	53
E_{85}	38	CLK	63	N.C.	1
E_{95}	37	C_0	75	N.C.	2
E_{A5}	36	C_1	74	N.C.	3
E_{B5}	35	C_2	10	N.C.	42
N_{00}	73	C_3	11	N.C.	43
N_{01}	71	C_4	12	N.C.	44
N_{02}	69	C_5	56	N.C.	83
N_{03}	62	C_6	55	N.C.	84
				N.C.	–

* substrate connection

Plastic Chip Carrier Package



All dimensions are in inches


DEVICES TESTED WITH THIS
OUTPUT LOAD CONFIGURATION
1.73V
R_L=660Ω
DUT
All outputs
except GO
C_L = 22pF
5.0V
R_L = 2.3KΩ
DUT
Global
Output (GO)
C_L = 40pF
Open drain output on GO allows up to
4 devices to be connected together.

GAPP APPLICATION NOTE NUMBER 3

Detection of Edges and Gradients in Binary and Gray Scale Images with the GAPP Processor

Introduction

The Geometric Arithmetic Parallel Processor chip (GAPP™) is a parallel processing integrated circuit. On each chip are 72 processing elements in a 12×6 array which operate concurrently to solve parallel processing problems (Figure 1). Each of the 72 processing elements executes the same instruction simultaneously while operating on its own data. If we contrast this with the traditional Von Neumann architecture that is commonly used in computers today (in which a single processing element processes a single data item at a time), by comparison, a single GAPP chip processes 72 times as many data items at a time. For this reason the architecture of the GAPP chip is referred to as Single Instruction Multiple Data, or SIMD as opposed to the Von Neumann architecture which is referred to as Single Instruction Single Data, or SISD.

Another feature of the GAPP chip is that each processing element can communicate with its four neighboring processing elements to the North, South, East, and West, a very desirable characteristic in many parallel processing applications (Figure 2). In effect, each processing element is only aware of processing elements in its local neighborhood. This property allows designers the ability to directly cascade chips to form larger arrays of processing elements without complicated interface logic. For example, if we connect 2 GAPP chips we have the power of 144 processing elements available to us, with all of these processing elements executing operations in tandem.

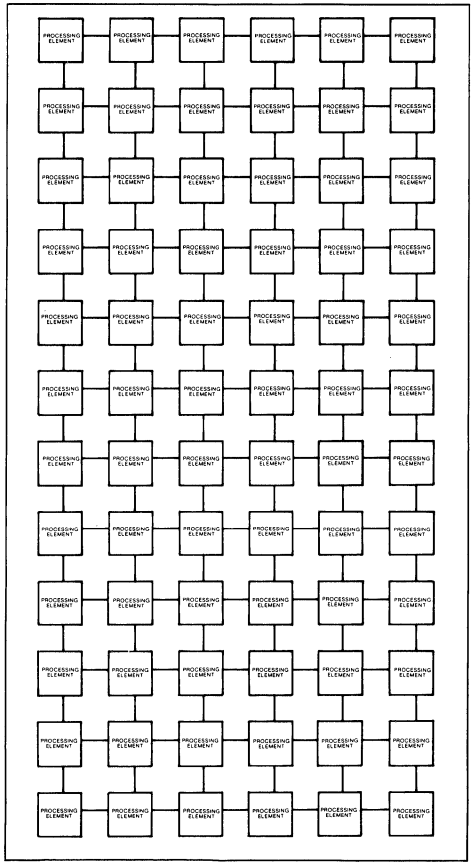


Figure 1. Array of GAPP Processing Elements on a single chip.

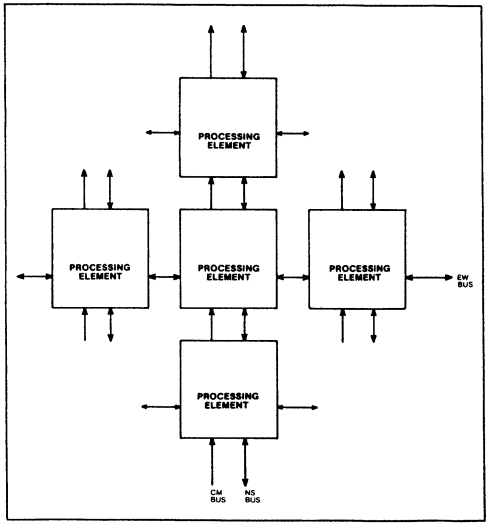


Figure 2. Communications Between Processing Elements.

In Figure 3 we see a schematic diagram of the individual processing element. This processing element is comprised of 4 registers, an ALU, and internal RAM. All data paths within and between processing elements are one bit wide. The CM register controls the flow of data on a unidirectional bus referred to as the CM bus. The CM bus allows data to be input and output from the processing element without interrupting the operation of the ALU. In effect the CM register acts as a shift register. The NS register is primarily concerned with managing the flow of data along a bidirectional bus called the NS bus. The NS bus is used to allow an individual processing element to communicate with processing elements to the North and South. The EW register handles the flow of data along the bidirectional EW bus. This bus allows the individual processing element to communicate with processing elements to the East and West. With this discussion, we now have a picture of how the individual processing element communicates with its neighbors. The final register that we must look at, the C register, is not used to communicate with neighboring processing elements, its function is strictly local to the individual processing element. But the C register is critical to the ALU function as we will see below in our discussion of the processing element ALU.

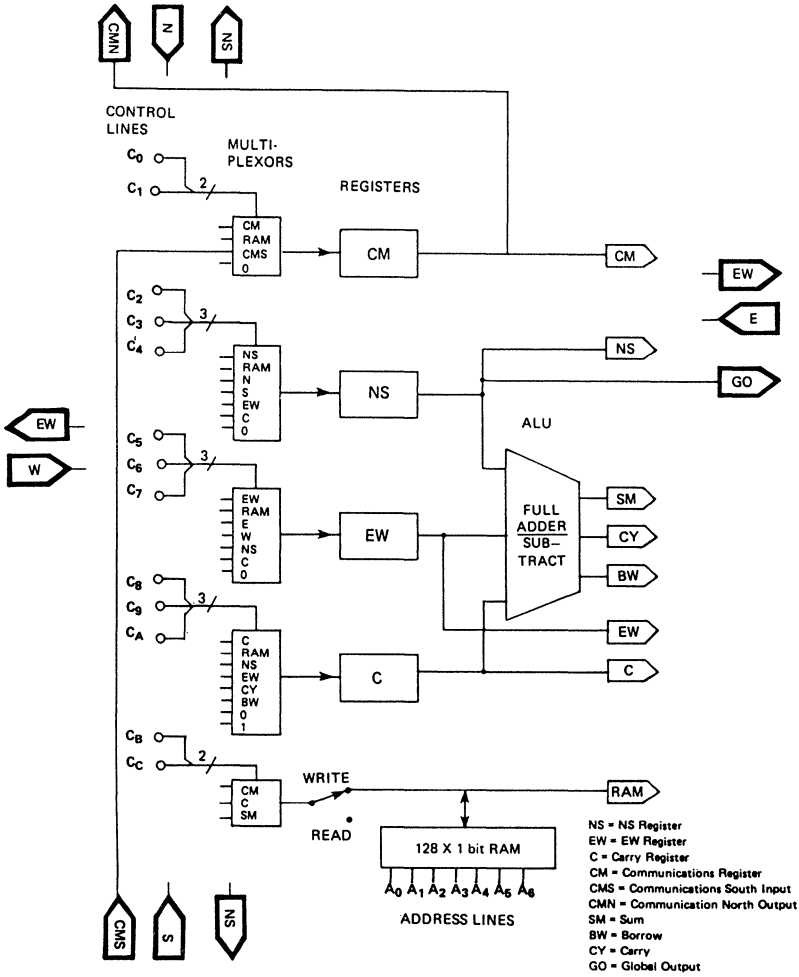


Figure 3. Schematic Diagram of the Processing Element.

Let us now turn our attention to the function of the ALU. The ALU within the processing element performs the full adder/subtractor function on its inputs: the NS register, the EW register, and the C register. The outputs of the ALU are sum (SM), carry (CY), and borrow (BW). The carry and borrow output can only be read by the C register. Thus, we see that the primary function of the C register is to act as a Carry register to store carrys and borrows that result from ALU operations. The fact that the ALU provides SM, CY, and BW outputs allows the ALU to perform all arithmetic and logical operations on its inputs in a bit serial fashion as shown in Figure 4.

If we take another look at the registers within each processing element we see that they receive their inputs from multiplexors that select from a variety of input sources. The C register can read its own output (a no-op condition), it can take its input from the processing element's own RAM, or from the NS, or EW registers. In addition, it can take its input from the CY and BW outputs from the ALU, or the register can be set to a vaue of 0 or 1. Likewise, other registers can also take inputs from a variety of sources as specified by their own multiplexors. For instance, we see that the EW register can receive its input from the East (or West) input to the processing element. In essence, the EW processing element is taking its data input from an EW register in the neighboring processing element to the east (or west). Similarly, the NS register can receive input from the North (or South) input to the processing element. Now that we have discussed the communications between processing elements and the processing within processing elements, let us now move on to a discussion of the processing element RAM.

Each processing element has its own 128 × 1 bit static RAM which is accessible to all of the registers within the processing element. While neighboring processing elements cannot access the individual processing element RAM directly, they can receive access indirectly by using the NS, or EW registers to transfer data to neighboring processing elements. The RAM is essential for working with multiple bit data. For instance, in working with 8-bit data the word would be stored in GAPP RAM and accessed one bit at a time as needed. To add two 8-bit numbers, A and B, we would store them in GAPP RAM using 16 of the 128 bits of available RAM, and add the bits one at a time. Thus, we could take the LSB of A, and put it in the NS register, the LSB of B and put it in the EW register. The result of the add is available on the SM output of the ALU, while the carry of the add is available on the CY output of the ALU. The SM output would then be stored in a location within GAPP RAM as the LSB of R, our result (Figure 5). The CY output would be stored as the carry in the C register while the next most significant bits of A and B are retrieved from GAPP Ram and loaded into the NS, and EW registers. This operation continues until each of the bits of A and B are processed. Thus, we see that the GAPP Ram is a vital component of the processing element in that it allows the processing element to perform bit serial manipulations on data of any bit length.

Arithmetic Operations

Adder/Subtracter Operations

INPUT			OUTPUT		
NS	EW	C	SM	CY	BW
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	0
1	1	0	0	1	0
0	0	1	1	0	1
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	1	1	1

Logic Operations

LOGICAL OPERATION	DESCRIPTION	CONDITIONS
INV	$SM = \overline{NS}$	EW = 0, C = 1
	$SM = \overline{EW}$	NS = 0, C = 1
	$SM = \overline{C}$	NS = 0, EW = 1
AND	$CY = NS \bullet EW$	C = 0
	$CY = EW \bullet C$	NS = 0
	$CY = NS \bullet C$	EW = 0
	$BW = \overline{NS} \bullet EW$	C = 0
OR	$CY = NS + EW$	C = 1
	$BW = \overline{NS} + EW$	C = 1
	$BW = EW + C$	NS = 0
XOR	$SM = NS \oplus C$	EW = 0
	$SM = NS \oplus EW$	C = 0
	$SM = EW \oplus C$	NS = 0
XNOR	$SM = \overline{NS \oplus EW}$	C = 1

Figure 4. GAPP Arithmetic and Logic Operations.

The GAPP chip uses an instruction set consisting of a sequence of mnemonics that controls the actions of the registers and RAM as shown in Table I. What we can see in the Table is that the actions of the registers and RAM are controllable independent of one another. Thus, we can conceivably have 5 operations that can be executed with a single 13 bit microcode word. In addition to the 13-bit wide microcode shown in Table I, we also need 7-bits to select an address within GAPP RAM to be read from or written to.

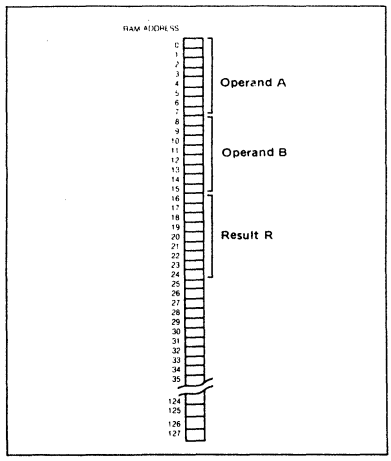
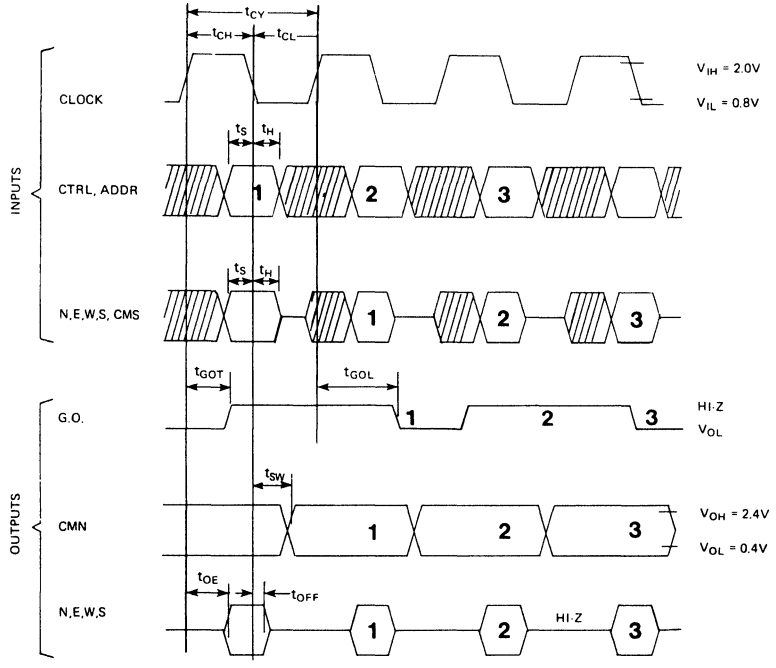


Figure 5. RAM within each GAPP processing element. Here we see how GAPP RAM can be used to store two operations (A and B) and the result R.

Register Operation	Mnemonic	Control Lines C_C	C_B	C_A	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	Description
CM	CM: = CM	X	X	X	X	X	X	X	X	X	X	X	0	0	MICRO-NOP
	CM: = RAM	X	X	X	X	X	X	X	X	X	X	X	0	1	LOAD CM FROM RAM
	CM: = CMS	X	X	X	X	X	X	X	X	X	X	X	1	0	MOVE FROM CMS INTO CM
	CM: = 0	X	X	X	X	X	X	X	X	X	X	X	1	1	LOAD 0 INTO CM
NS	NS: = NS	X	X	X	X	X	X	X	X	0	0	0	X	X	MICRO-NOP
	NS: = RAM	X	X	X	X	X	X	X	X	0	0	1	X	X	LOAD NS FROM RAM
	NS: = N	X	X	X	X	X	X	X	X	0	1	0	X	X	MOVE FROM N INTO NS
	NS: = S	X	X	X	X	X	X	X	X	0	1	1	X	X	MOVE FROM S INTO NS
	NS: = EW	X	X	X	X	X	X	X	X	1	0	0	X	X	MOVE FROM EW INTO NS
	NS: = C	X	X	X	X	X	X	X	X	1	0	1	X	X	MOVE FROM C INTO NS
	NS: = 0	X	X	X	X	X	X	X	X	1	1	0	X	X	LOAD 0 INTO NS
EW	EW: = EW	X	X	X	X	X	0	0	0	X	X	X	X	X	MICRO-NOP
	EW: = RAM	X	X	X	X	X	0	0	1	X	X	X	X	X	LOAD EW FROM RAM
	EW: = E	X	X	X	X	X	0	1	0	X	X	X	X	X	MOVE FROM E INTO EW
	EW: = W	X	X	X	X	X	0	1	1	X	X	X	X	X	MOVE FROM W INTO EW
	EW: = NS	X	X	X	X	X	1	0	0	X	X	X	X	X	MOVE FROM NS INTO EW
	EW: = C	X	X	X	X	X	1	0	1	X	X	X	X	X	MOVE FROM C INTO EW
	EW: = 0	X	X	X	X	X	1	1	0	X	X	X	X	X	LOAD 0 INTO EW
C	C: = C	X	X	0	0	0	X	X	X	X	X	X	X	X	MICRO-NOP
	C: = RAM	X	X	0	0	1	X	X	X	X	X	X	X	X	LOAD C FROM RAM
	C: = NS	X	X	0	1	0	X	X	X	X	X	X	X	X	MOVE FROM NS INTO C
	C: = EW	X	X	0	1	1	X	X	X	X	X	X	X	X	MOVE FROM EW INTO C
	C: = CY	X	X	1	0	0	X	X	X	X	X	X	X	X	LOAD C FROM CARRY
	C: = BW	X	X	1	0	1	X	X	X	X	X	X	X	X	LOAD C FROM BORROW
	C: = 0	X	X	1	1	0	X	X	X	X	X	X	X	X	LOAD 0 INTO C
	C: = 1	X	X	1	1	1	X	X	X	X	X	X	X	X	LOAD 1 INTO C
RAM	READ	0	0	X	X	X	X	X	X	X	X	X	X	X	READ FROM RAM
	RAM: = CM	0	1	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM CM
	RAM: = C	1	0	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM C
	RAM: = SM	1	1	X	X	X	X	X	X	X	X	X	X	X	LOAD RAM FROM SUM

Table I. GAPP Instruction Set.

The timing requirements for the GAPP chip are shown in Table II. The NCR45CG72-1 uses a 10 MHz clock while the NCR45CG72-2 uses a 5 MHz clock. In the discussion below, clock timings for the 45CG72-2 are indicated in parentheses when they differ from those for the 45CG72-1.



NOTE: 1,2,3 refer to the staging sequence of instruction, data in and data out.

AC Characteristics

PARAMETER	SYMBOL	NCR45CG72-2		NCR45CG72-1		
		MIN	MAX	MIN	MAX	UNITS
CYCLE TIME	t_{CY}	200		100		ns
CLOCK LOW	t_{CL}	100	5000	50	5000	ns
CLOCK HIGH	t_{CH}	100	5000(1)	50	5000(1)	ns
SETUP TIME	t_S	20		10		ns
INPUT HOLD TIME	t_H	10		10		ns
OUTPUTS ENABLED	t_{OE}	10	50	10	35	ns
OUTPUT HOLD TIME	t_{OFF}	10	50	10	30	ns
GLOBAL OUTPUT LOW	t_{GOL}	20	100	10	70	ns
GLOBAL OUTPUT TRISTATE	t_{GOT}	10	50	10	35	ns
CMN OUTPUT	t_{SW}	20	120	10	85	ns

NOTE: (1) d.c. by design; tested at 5 μsec.

Table II. GAPP Timing Diagram.

Control (CTRL) and address (ADDR) data are latched on the falling edge of the system clock and require a 10ns (20ns) set up time and a 10ns hold time (denoted by '1' in Table II). Inputs to processing elements from the N, E, W, S, and CMS lines become valid on the falling edge of the *next* clock cycle and require the same set up and hold times as the CTRL and ADDR signals (denoted by '1' in Table II).

The GAPP chip has an output referred to as Global Output (GO). This is the output from a 72 input NOR gate whose inputs are from each of the NS registers in the chip. This output can be used to flag conditions within the chip by loading the data into the NS registers. GO tristates within 35ns (50ns) relative to the rising clock edge allowing 65ns (150ns) for an external pull up resistor to bring the GO point high. If a longer pullup time is required in a system design, executing a NOP on the NS registers will hold the current values for additional cycles. This time for a valid low output is 70ns (100ns) relative to the rising clock edge (denoted by '1' in Table II).

The CMN output becomes valid 85ns (120ns) after the falling edge of the clock. The output (denoted by '1') in Table II is valid for CMS input (denoted by "1"). This means that the CMN output is available to the neighboring processing elements to the North at least 5ns (60ns) prior to when it is needed for input set up.

Similarly, N, E, W, and S outputs become valid at least 5ns before they are needed as inputs to their neighbors. Consequently, the output enable time is 35ns (50ns) as measured from the rising clock edge. The output hold time is 30ns (50ns). The output indicated at '1' in Table II is valid for N, E, W, and S inputs denoted as '1'.

Edge Detection with GAPP Device

The parallel architecture of the GAPP chip, in which 72 individual processing elements execute instructions concurrently, makes it ideal for many "neighborhood" operations frequently encountered in image processing applications. Edge detection (or edge extraction) is one example of an algorithm which uses such neighborhood operations.

Ideally, is a gray-level picture, an edge is the boundary between two regions of constant gray-level. In practice, however, there are often shadings that degrade the sharpness of the edge.

Edge extraction is a critical step in tasks such as feature extraction, object recognition, and region segmentation. There are numerous approaches to the task of edge detection in the literature.[1-6] There is, however, no best approach as the "best" edge detection method is determined according to the requirements of the particular application.

In the example discussed in this application note, we will use a technique known as the Sobel operator. The Sobel operator has the property whereby the determination of whether a point in the image belongs to an edge is made independent of whether neighboring points in the image also belong to an edge. To make this determination in the Sobel operation we start with 3×3 mask coefficients in the X and & directions:

X-mask -

-1	0	1
-2	0	2
-1	0	1

Y-mask =

1	2	1
0	0	0
-1	-2	-1

operating on a 3×3 neighborhood of pixels within the image:

3x3 Neighborhood =

A	*B*	*C*
D	*E*	*F*
G	*H*	*I*

Using the X-mask and Y-mask gives us the following equations for image gradients in the X and Y directions (where *A* through *I* represent gray-level intensities for the respective pixels in the 3×3 neighborhood):

$$X_S = \text{X-gradient} = (C + 2F + I) - (A + 2D + G) \quad [1]$$

$$Y_S = \text{Y-gradient} = (A + 2B + C) - (G + 2H + I) \quad [2]$$

The gradient equations above represent the Sobel gradients about point *E*.

A system implemented using GAPP chips can determine the Sobel gradient for every point in an image simultaneously. By performing a threshold operation on the derived gradients over each point in the image (simultaneously using the GAPP chip) to eliminate points whose gradient is below a predetermined gray-level value, the edge of an object can be extracted. Once the edge is extracted, subsequent processing such as spoke filtering (to recognize shape, etc.) can be performed.[7]

As an illustrative example of GAPP programming, we will develop a program for the GAPP to perform edge detection using the Sobel algorithm. In our example we will use a 12 × 12 pixel image (Figure 6) of uniform intensity against a dark background (also of uniform intensity). We implement (in hardware) a cylindrical image wrap-around in both X and Y directions. As a result, data output on one edge of the array becomes input to the other edge of the array on the same line or column (Figure 7).

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	50	50	0	0	0	0	0	0	0
0	0	50	50	50	0	0	0	0	0	0	0
0	50	50	50	50	50	0	0	0	0	0	0
0	50	50	50	50	50	50	50	0	0	0	0
0	50	50	50	50	50	50	50	50	0	0	0
0	0	50	50	50	50	50	50	50	50	0	0
0	0	0	50	50	50	50	50	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 6. Input Image. Uniform Intensity Object Against.

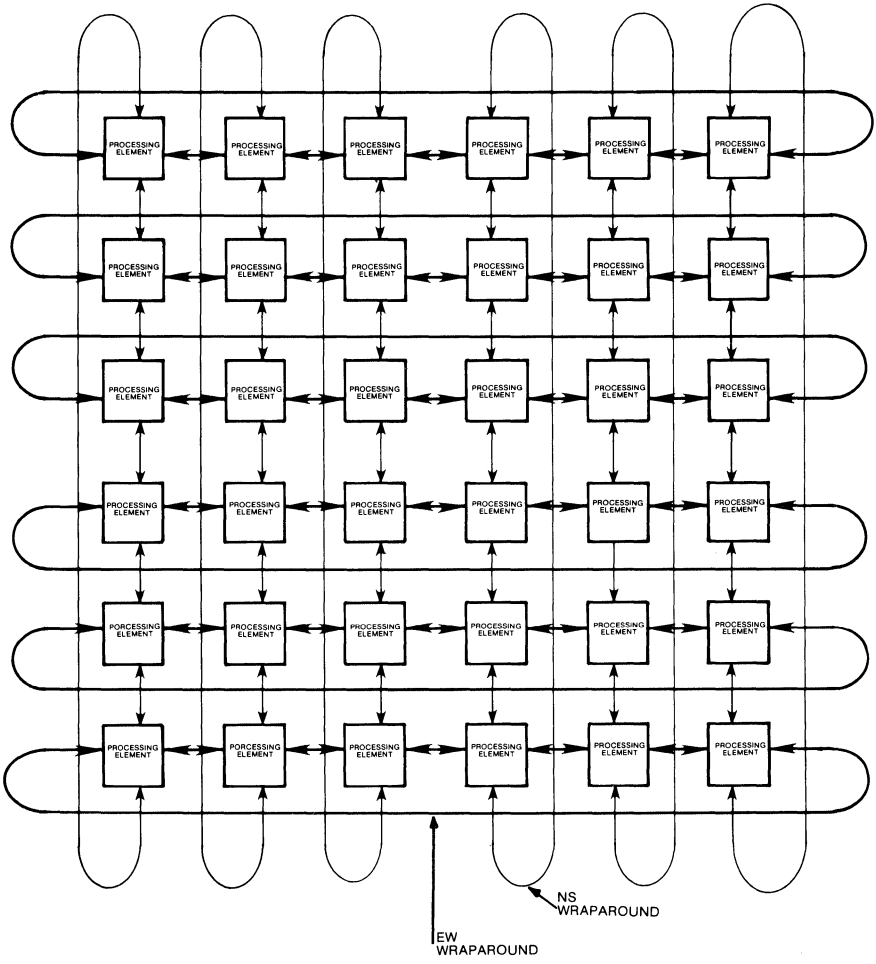


Figure 7. Cylindrical wraparound in NS and EW directions.

Because the GAPP chip contains 72 processing elements, arrayed in a 12 × 6 configuration, we can use just two GAPP chips to implement a 12 × 12 array of processing elements which will then allow us to process the entire 12 × 12 pixel image in a single pass. To process larger images, the system designer has two options: (a) implement a system using more GAPP chips, or (b) use a frame buffer to load portions of the image 12 × 12 pixels at a time into the GAPP subsystem.

If we concern ourselves, for the moment, with the 3 × 3 neighborhood within the 12 × 12 image over which the Sobel gradient is being computed, we can illustrate how the GAPP can be used to determine the Sobel Y-gradient about point *E*.

We start (in Figure 8) with each GAPP processing element in the 3 × 3 neighborhood containing the gray scale value of each pixel in the 3 × 3 neighborhood [a]. Next, we add the gray scale value of each pixel (from [a]) to its immediate neighbor to the left. This gives us the result [b] in the 3 × 3 neighborhood. A '—' represents a processor element whoe contents are irrelevant to the computation of the Sobel Y-gradient at point *E* (but relevant to the computation of the Sobel Y-gradient in adjacent 3 × 3 neighborhood within the image). Then we add the gray scale value of each pixel (from [b]) to its immediate neighbor to the right ([a]). Finally, by taking the subtractive difference of the gray scale values of the pixels above and below our center point, *E* (from [c]), we derive the Sobel Y-gradient at point *E*. By using a similar process we can also process the Sobel X-gradient at point *E*.

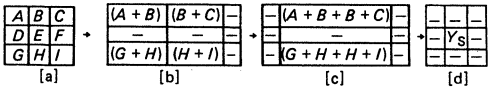


Figure 8. Sobel X Transform.

Computing the Sobel Gradient

Appendix A shows a GAPP program to perform edge detection using the Sobel gradient operator. This program was developed and executed on the GAPP PC Development System (NCR45GDS1) which runs on IBM and IBM compatible personal computers. The code was written in a language called the GAPP Algorithm Language (GAL™) which is a subset of the C programming language. "Image" variables are used to assign names to a set of locations within GAPP RAM at each processing element. For example, the statement on line 100 of the program **"image A:0:7",** assigns RAM locations 0-7 to the variable the input data (represented in 8-bit, two's complement format). **"T"** represents the gradient threshold level at which we accept or reject points that belong to an edge, RAM locations 8-15 are assigned to image variable **"R,"** the output from the Sobel operation. We begin by tracing through the steps required to generate the Sobel Y-gradient as illustrated in Figure 8.

Computing the First Partial Sum

Lines 300-640 in our program compute the Sobel Y-gradient according to the equation [2]. The first "while" loop (lines 350-410) takes the input gray scale data (Figure 8 [a]) and produces the first partial sum (Figure 8 [b]). Above, in line 320 we initialize the C (Carry) registers of every processing element within the GAPP array by setting them to 0. On the first pass through the loop, when i=0, we get the following (again, just focusing on a 3 × 3 neighborhood about a point *E*:

INSTRUCTION: A:i EW := RAM; (i = 0)

EW =

A_0	B_0	C_0
D_0	E_0	F_0
G_0	H_0	I_0

Where A_0 through I_0 are the least significant bits of the 8-bit gray level intensities. These LSB's are now contained in the EW registers within the GAPP array:

INSTRUCTION: A:i NS :=RAM EW :=E; (i=0)

NS =

A_0	B_0	C_0
D_0	E_0	F_0
G_0	H_0	I_0

EW =

B_0	C_0	—
E_0	F_0	—
H_0	I_0	—

Where the NS registers within each processing element contain the LSB's of gray levels in the 3 × 3 neighborhood as above. In addition, the local connectivity among processing elements allows us to move data between processing elements as was done with the EW registers that now contain the input data after being shifted to the left in the processing element array. Again, a "—" refers to a data value that is irrelevant to the Sobel computation at point *E* within our selected 3 × 3 neighborhood. Also note that within each processing element we have different values in the EW and NS registers. For example, the NS register of the processing element in the upper left hand corner (the 1.1 location in matrix notation) contains the LSB of the pixel *A* from Figure 8 [a], while the EW register contains the LSB of pixel *B*.

INSTRUCTION: B:0 RAM := SM C := CY; (i = 0)

RAM(B:0) =

$(A_0 + B_0)$	$(B_0 + C_0)$	–
$(D_0 + E_0)$	$(E_0 + F_0)$	–
$(G_0 + H_0)$	$(H_0 + I_0)$	–

C =

Carry$(A_0 + B_0)$	Carry$(B_0 + C_0)$	–
Carry$(D_0 + E_0)$	Carry$(E_0 + F_0)$	–
Carry$(G_0 + H_0)$	Carry$(H_0 + I_0)$	–

The result of addition of the contents of EW and NS registers is stored as the LSB of image *B*, a temporary variable to store the first partial sum of the Sobel Y-gradient calculation. The C register within each processing element holds the carry bit from the addition of LSB's above (which will be added on the next pass through the loop). Thus, for i = 1 we have:

INSTRUCTION: B:1 RAM := SM C := CY; (i = 1)

RAM(B:1) =

$(A_1 + B_1 + \text{Carry}(A_0 + B_0))$	$(B_1 + C_1 + \text{Carry}(B_0 + C_0))$	–
$(D_1 + E_1 + \text{Carry}(D_0 + E_0))$	$(E_1 + F_1 + \text{Carry}(E_0 + F_0))$	–
$(G_1 + H_1 + \text{Carry}(G_0 + H_0))$	$(H_1 + I_1 + \text{Carry}(H_0 + I_0))$	–

C =

Carry$(A_1 + B_1 + \text{Carry}(A_0 + B_0))$	Carry$(B_1 + C_1 + \text{Carry}(B_0 + C_0))$	–
Carry$(D_1 + E_1 + \text{Carry}(D_0 + E_0))$	Carry $(E_1 + F_1 + \text{Carry}(E_0 + F_0))$	–
Carry$(G_1 + H_1 + \text{Carry}(G_0 + H_0))$	Carry$(H_1 + I_1 + \text{Carry}(H_0 + I_0))$	–

This process continues in bit serial fashion until we have processed the entire 8-bit number. Note that in the above discussion, only elements (1,1) (1,2), (3,1), and (3,2) (matrix notation) in the 3 × 3 neighborhood contribute to the Sobel Y-gradient at point *E*. We complete the first partial sum by performing a sign extension to generate a 9-bit first partial sum. This is done in line 420 where we store the final carry bit from the 8-bit addition.

Computing the Second Partial Sum

To compute the second partial sum, we start with our first partial sum from above, and derive the second partial sum as in Figure 8 [c]. This is done in lines 430-520 as shown below:

INSTRUCTION: B:0 EW := RAM; (i = 1)

EW =

$(A + B)_0$	$(B + C)_0$	–
–	–	–
$(G + H)_0$	$(J + I)_0$	–

EW registers contain LSB's of each partial sum in the 3 × 3 neighborhood:

INSTRUCTION: B:0 NS := RAM EW := W; (i = 1)

NS =

$(A + B)_0$	$(B + C)_0$	–
–	–	–
$(G + H)_0$	$(H + I)_0$	–

EW =

–	$(A + B)_0$	$(B + C)_0$
–	–	–
–	$(G + H)_0$	$(H + I)_0$

INSTRUCTION: C:0 RAM := SM C := CY; (i = 0)

$$\text{RAM(C:0)} = \begin{array}{|c|c|c|}\hline - & ((A+B)_0 + (B+C)_0) & - \\ \hline - & - & - \\ \hline - & ((G+H)_0 + (H+I)_0) & - \\ \hline \end{array}$$

$$\text{C} = \begin{array}{|c|c|c|}\hline - & \text{Carry}((A+B)_0 + (B+C)_0) & - \\ \hline - & - & - \\ \hline - & \text{Carry}((G+H)_0 + (H+I)_0) & - \\ \hline \end{array}$$

As before, this process continues until we bit-serially process all the bits in the first partial sum. In line 520, we extend the sign bit.

Completing the Sobel Y-Gradient Calculation

Finally, we are to the stage of Figure 8 [c] and can now complete the Sobel Y-gradient procedure (lines 530-630):

INSTRUCTION: C:0 NS := RAM; (i = 0)

$$\text{NS} = \begin{array}{|c|c|c|}\hline - & (A+2B+C)_0 & - \\ \hline - & - & - \\ \hline - & (G+2H+I)_0 & - \\ \hline \end{array}$$

INSTRUCTION: NS := S;

$$\text{NS} = \begin{array}{|c|c|c|}\hline - & - & - \\ \hline - & (G+2H+I)_0 & - \\ \hline - & - & - \\ \hline \end{array}$$

INSTRUCTION: C:0 NS := RAM EW := NS; (i = 0)

$$\text{NS} = \begin{array}{|c|c|c|}\hline - & (A+2B+C)_0 & - \\ \hline - & - & - \\ \hline - & (G+2H+I)_0 & - \\ \hline \end{array}$$

$$\text{EW} = \begin{array}{|c|c|c|}\hline - & - & - \\ \hline - & (G+2H+I)_0 & - \\ \hline - & - & - \\ \hline \end{array}$$

INSTRUCTION: NS := N; (i = 0)

$$\text{NS} = \begin{array}{|c|c|c|}\hline - & - & - \\ \hline - & (A+2B+C)_0 & - \\ \hline - & - & - \\ \hline \end{array}$$

INSTRUCTION: Y:0 RAM := SM C := BW; (i = 0)

$$\text{RAM(Y:0)} = \begin{array}{|c|c|c|}\hline - & - & - \\ \hline - & [(A+2B+C)_0 - (G+2H+I)_0] & - \\ \hline - & - & - \\ \hline \end{array}$$

$$\text{C} = \begin{array}{|c|c|c|}\hline - & - & - \\ \hline - & \text{Borrow}[(A+2B+C)_0 - (G+2H+I)_0] & - \\ \hline - & - & - \\ \hline \end{array}$$

After bit serially processing the second partial sums and extension of the sign bit (line 640), we have our Sobel Y-gradient stored as image variable **Y**.

Computing the Sobel X-Gradient

Computing the Sobel X-gradient follows the same procedure from a conceptual standpoint as the computation of the Sobel Y-gradient. Thus, we show the data flow for calculation of the Sobel X-gradient without further explanation:

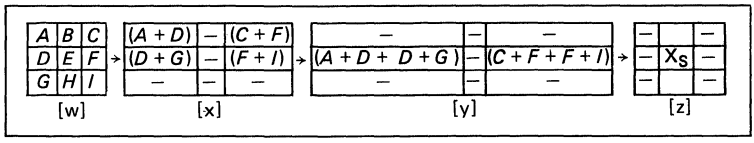


Figure 9. Sobel X Transform.

Computing the Gradient Magnitude

Having computed the X and Y gradients, we must next compute the magnitude of the gradient at point *E* (which we refer to a Grad(E)):

$$\text{Magnitude [Grad(E)]} = (X_S^2 + Y_S^2)^{1/2}$$

which we approximate as:

$$\text{Magnitude [Grad(e)]} = \text{Max}(|X_S|,|Y_S|) + {}^1/_2[\text{Min}(|X_S|,|Y_S|)]$$

This function is computed in lines 990-1930. These operations do not involve exchange of data between processing elements. Thus, in this portion of the program, each processing element independently operates on the pixel data within its own RAM.

The flow of this section of code is as follows:

a) Derive absolute value of X_S and Y_S by performing bitwise invert of X and Y gradients if the sign bit (MSB) is 1.

b) If sign bit is 1, add 1 to the result of operation a).

c) Derive Min ($|X_S|$, $|Y_S|$)

d) Derive Max ($|X_S|$, $|Y_S|$)

e) Derive Magnitude [Grad(E)]

Statement 1030, "n = size (X)" returns a value equal to the number of bits in image X (in this case 11). In statement 1040, we take the MSB of X_S (stored in RAM location X:10) and load it into the C register. We then enter a while loop (lines 1060-111-) in which we:

1) Take each bit of X_S (starting with the LSB) from GAPP RAM and load it into the EW register (line 1080).

2) As we process each bit from X_S we perform an exclusive-or of that bit with the sign bit (line 1090). To perform this operation we set the NS register to 0, store the sign bit in the C register, and the bits of X_S in EW (bit serially). The SM output of each processing element's ALU represents the logical operation C ⊕ EW. If the sign bit is 0, no bitwise invert is performed since the result of the exclusive-or operation is whatever is in the EW register.

3) Replace the previous value of the i^{th} bit of X_S with the result from 2) as we bit serially processX_S.

After we complete the while loop of lines 1060-1110, we enter another while loop (lines 1140-1190) in which we add the sign bit to the result of the operation performed in lines 1060-1110 (which we will refer to as X'_S):

1) Serially take each bit of X'_S: and load it into the EW register (line 1160).

2) Add the contents of the C register to the contents of the EW register (line 1170). The SM output contains the result of the addition operaton. On the first pass (when i = 0) the C register still contains the sign bit and the NS register still contains 0. Thus, we are adding the sign bit to the LSB of X'_S on the first pass. The carry generated by this operation becomes the new value stored in the C register.

3) Replace the previous value of the i^{th} bit of X'_S with the result of 2).

In lines 1210-1400 we derive the aboslute value of Y_S. Figure 10 show $|X_S|$ and $|Y_S|$.

The first step towards determining the Min ($|X_S|$, $|Y_S|$) is to set a flag. This is done in the while loop of lines 1440-1490. Here we bit serially subtract the absolute value of Y_S from the absolute value of X_S. The borrow is stored in the C register. At the end of the while loop, we store the final borrow in RAM as a flag bit. If $|X_S|$ is greater than or equal to $|Y_S|$, then the final borrow (stored as image variable xF) will be 0, otherwise it will be 1 (Figure 11).

0	0	0	0	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	150	0	150	50	0	0	0	0	0	0
0	150	200	0	200	150	0	0	0	0	0	0
50	200	150	0	150	200	50	0	0	0	0	0
150	200	50	0	50	150	100	50	50	0	0	0
200	200	0	0	0	50	50	100	150	50	0	0
150	200	50	0	0	0	0	50	150	150	50	0
50	150	150	50	0	0	0	0	100	200	100	0
0	50	150	150	50	0	0	50	150	150	50	0
0	0	50	150	100	0	0	100	150	50	0	0
0	0	0	50	50	0	0	50	50	0	0	0

(a)

0	0	0	0	0	0	0	0	0	0	0	0
0	0	50	100	50	0	0	0	0	0	0	0
0	50	150	200	150	50	0	0	0	0	0	0
0	50	100	100	100	50	0	0	0	0	0	0
50	100	50	0	50	100	50	0	0	0	0	0
50	100	50	0	50	150	200	150	50	0	0	0
0	0	0	0	0	50	150	200	150	50	0	0
50	100	50	0	0	0	0	50	150	150	50	0
50	150	150	50	0	0	0	0	0	0	0	0
0	50	150	150	50	0	0	50	150	150	50	0
0	0	50	150	200	200	200	200	150	50	0	0
0	0	0	50	150	200	200	150	50	0	0	0

(b)

Figure 10. (a) Absolute value of Sobel gradient in X direction: $|X_S|$.
(b) Absolute value of Sobel gradient in Y direction: $|Y_S|$.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0

Figure 11. Image Variable xF. A '1' indicates pixels in the image where $|X_S| < |Y_S|$.

In lines 1520-1650, we determine Min ($|X_S|$,$|Y_S|$). First, the stored image variable, xF, is loaded into the NS register. We then enter a while loop which performs the logical operation $|X_S|$*xF + $|Y_S|$*~xF, which returns the lesser of $|X_S|$ and $|Y_S|$. In line 1560, we put the absolute value of X_S (bit serially), in the EW register, at the same time we set the C register to 0. In line 1570, we put in C the result of the logical operation: $|X_S|$*xF. in line 1580, this result is stored as the temporary image variable xT while the C register is again set to 0.

In line 1590 we put the absolute value of Y_S (bit serially), in the EW register, but this time the result put into the C register (line 1600) is the result of the logical operation $|Y_S|$*~xF. Had we not set the C register to 0 in line 1580, we could have done so in line 1590 and the result of line 1600 would have been the same. We simply took advantage of the fact that each register can be independently set to a value on a single line of code to set the C register at the first available instance.

In line 1610 we set the C, EW, and NS registers in one instruction. The result from line 1600 is transferred from the C register, and put into the NS register on line 1610. The temporary image variable, xT, is taken from GAPP RAm and loaded into the EW register. Finally, the C register is set to 1. On line 1620 the result loaded into the C register is the logical operation: $|X_S|$*xF+$|Y_S|$*~xF. On this same line the NS register is reloaded with the flag bit xF. On line 1630, the current contents of the C register are stored as image variable xD in GAPP RAM.

We then return to the beginning of the loop and process the next bits of $|X_S|$, and $|Y_S|$. This entire process repeats until all the bits of $|X_S|$ and $|Y_S|$ have been processed.

If we return to line 1620 we can see that there is an advantage in loading the flag bit, xF, into the NS register at that time. If we look at the flow of the while loop, we can see that at the start of each iteration it is assumed that the flag bit is stored in the NS register. Thus, if we did not set the NS register before we started each iteration we could not do so on the first instruction in the loop. This is because we can access only one RAM address per instruction. Since the current first instruction accesses a bit from $|X_S|$, then in order to load the flag bit into the NS register we would have to add an instruction to load the NS register from RAM with the flag bit before we could obtain the result in line 1570. By setting the NS register at the end of the previous iteration in line 1620, we save one instruction per iteration that would have been needed had we decided to wait until the beginning of the next iteration before loading the NS register.

The above operations are repeated in lines 1670-1800 to produce the Max($|X_S|$,$|Y_S|$). This is done by simply interchanging $|X_S|$, and $|Y_S|$ from lines 1520-1640.

At this point, we have Min ($|X_S|$, $|Y_S|$) stored as image variable xD, while Max ($|X_S|$, $|Y_S|$) is stored as image variable xC. To complete the magnitude calculation, we must now sum xC + xD/2. This is done in lines 1830-1920 (Figure 12).

0	0	0	0	0	0	0	0	0	0	0	0
0	0	75	100	75	0	0	0	0	0	0	0
0	75	225	200	225	75	0	0	0	0	0	0
0	175	250	100	250	175	0	0	0	0	0	0
75	250	175	0	175	250	75	0	0	0	0	0
175	250	75	0	75	225	250	175	75	0	0	0
200	200	0	0	0	75	175	250	225	75	0	0
175	250	75	0	0	0	0	75	225	225	75	0
75	225	225	75	0	0	0	0	100	200	100	0
0	75	225	225	75	0	0	75	225	225	75	0
0	0	75	225	250	200	200	250	225	75	0	0
0	0	0	75	175	200	200	175	75	0	0	0

Figure 12. Image variable $\not{Z}$: the unnormalized Gradient Magnitude.
$\not{Z}$ = Max ($|X_S|$, $|Y_S|$) + ½ Min ($|X_S|$, $|Y_S|$).

The procedure for adding xD/2 to xC is to truncate the LSB of xD and roundup to nearest integer. In line 1830 we load the LSB of xD into the C register. Starting on line 1840 we enter a for loop that performs the operation xC + xD/2. On the first iteration, when i = 0 we add the LSB of xD (stored in the C register), to the second most significant bit of xD (stored in the EW register), and to the LSB of xC (stored in the NS register). What we are doing on this first iteration is treating the LSB of xD (which we are using for roundup) as the initial carry (instead of starting with an initial carry of 0), which in effect it is. We continue on subsequent iterations by adding the i^{th} bit of xC to the $(i + 1)^{th}$ bit of xD.

Completing the Edge Detection Procedure

The final steps toward deriving the edges of our input image are performed in lines 1970-2280 in which we follow the procedure below:

a) Compare the Sobel gradient result of Figure 12 against the threshold gray level

b) Indicate all gradients in the image with a gray level greater than 60 by raising a flag bit

c) Use the resulting flag bit field to identify points in the input image of Figure 6 that belong to an edge

In lines 1970-2100 we threshold the magnitude of the Sobel gradient derived above to eliminate points in the image that fall below a set threshold in gray level intensity. This is basically done by taking each bit of the Sobel gradient and performing a bitwise subtract with each bit of the predetermined threshold level (in this case we set the threshold to 60 at the beginning of the program). If the Sobel gradient is greater than the threshold level, then the final borrow from the subtraction of: [(Threshold Level) — (Sobel Gradient)] will be '1', otherwise it will be '0'. The final carry as we leave the while loop is stored in the C register, and subsequently stored as image variable "flag" in GAPP RAM. Since the entire image is processed in parallel in our example, each of the 12 × 12 processing elements in the array will record a 0 or a 1 depending upon whether the Sobel gradient at a particular point in the image exceeds the threshold gray level.

As we examine the while loop in lines 1970-2090 we observe that the EW register is used to store successive bits representing the value of the Sobel gradient for each processing element, while the NS register is used to store successive bits representing the value of the threshold gray level. The if-else test of lines 2010-2030 generates the successive bits in the threshold and loads them into the C register. After completing the test, the result is subsequently loaded into the NS register on line 2060. We also note that an integral part of the if-else test is an unconditional loading of the NS register with the previous contents of the C register. At the beginning of each iteration in the while loop, we find that the C register contains the borrow from the previous iteration's bit subtraction. Thus, we preserve the prior borrow by storing it in the NS register while we reset the C register with a new bit from the threshold. When we complete the if-else test we then (on line 2060) not only load the NS register with the result of the if-else text from the C register, but we at the same time load the C register with the current borrow from the NS register (in effect swapping data between the NS and C registers).

0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0
1	1	1	0	1	1	1	0	0	0	0	0
1	1	1	0	1	1	1	1	1	0	0	0
1	1	0	0	0	1	1	1	1	1	0	0
1	1	1	0	0	0	0	1	1	1	1	0
1	1	1	1	0	0	0	0	1	1	1	0
0	1	1	1	1	0	0	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	0	0	0

Figure 13. Image variable "flag": A '1' indicates a pixel whose gray level exceeds the threshold level.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	50	50	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	50	0	50	50	0	0	0	0	0	0
0	50	0	0	0	50	50	50	0	0	0	0
0	50	50	0	0	0	0	50	50	0	0	0
0	0	50	50	0	0	0	0	50	50	0	0
0	0	0	50	50	0	0	50	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 14. Thresholded Edge.

Now we have an image variable, flag (Figure 13), that represents in a pattern of 1's and 0's which pixels in the image have an image gradient that lies on an edge (or at least what we have determined with our threshold to be an edge). We now "and" each bit of our input image variable, image A, with our flag bit field. The result is a representation of the edges that lie in the original image. This is stored as image variable "R", our result field. Lines 2150-2280 perform the "and" function.

Effect of Threshold

In lines 2150-2280 we start by loading the flag bit into the NS register (line 2180), and the LSB of the input image (image A) into the EW register (line 2190). We then enter our while loop that successively takes each bit of image A and performs a logical "and" operation on each of these bits with the flag bit. On the first iteration when i = 0 we take the result of the "and" of the LSB of image A with the flag bit and load into the C register (line 2240). At the same time we load the next bit of image A into the EW register for the next iteration. Still on the first iteration we take the current contents of the C register, and store it as the LSB of our result, image variable R (line 2250). Since the logical operation NS * EW appears on the CY output of the ALU when the contents of C register is 0, we also set the C register to 0 on line 2250 for the next iteration. After completion of this while loop, our final result is derived (Figure 14).

If we look at the gradients in the image (Figure 12), it becomes immediately apparent that all non-zero gradients are greater than our set threshold of 60. Thus, this set threshold has no impact at all on the final result (Figure 14). If we vary the threshold levels as in Figures 15 we can see how the threshold level affects the final edge that we observe at the end of the procedure. In 15(a) we see the case of a threshold of 0. Here, the result is the same as in Figure 14. In 15(b), we set the threshold to 75 and can see that the edge is now narrower than in the previous cases. In 15(c) and 15(d) we set the threshold to 100 and 150 respectively. Here we see no difference in the final result between the two cases. In 15(e), however, when the threshold was set 175 we see a discontinuity in the edge. As we vary the threshold from 200 to 225, in 15(f) and 15(g) respectively, we see the edge continue to disintegrate. Finally, when we set the threshold to 250 as in 15(h), the edge disappears completely.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	50	50	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	50	0	50	50	0	0	0	0	0	0
0	50	0	0	0	50	50	50	0	0	0	0
0	50	50	0	0	0	0	50	50	0	0	0
0	0	50	50	0	0	0	0	50	50	0	0
0	0	0	50	50	0	0	50	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(a). Threshold Level of 0.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	50	50	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	0	0	0	50	0	0	0	0	0	0
0	50	0	0	0	0	50	50	0	0	0	0
0	50	0	0	0	0	0	0	50	0	0	0
0	0	50	0	0	0	0	0	50	50	0	0
0	0	0	50	0	0	0	0	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(b). Threshold Level of 75.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	0	0	0	50	0	0	0	0	0	0
0	50	0	0	0	0	50	50	0	0	0	0
0	50	0	0	0	0	0	0	50	0	0	0
0	0	50	0	0	0	0	0	0	50	0	0
0	0	0	50	0	0	0	0	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(c). Threshold Level of 100.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	50	0	0	0	50	0	0	0	0	0	0
0	50	0	0	0	0	50	50	0	0	0	0
0	50	0	0	0	0	0	0	50	0	0	0
0	0	50	0	0	0	0	0	0	50	0	0
0	0	0	50	0	0	0	0	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(d). Threshold Level of 150.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	50	0	0	0	50	0	0	0	0	0	0
0	50	0	0	0	0	0	50	0	0	0	0
0	50	0	0	0	0	0	0	50	0	0	0
0	0	50	0	0	0	0	0	0	50	0	0
0	0	0	50	0	0	0	0	50	0	0	0
0	0	0	0	50	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(e). Threshold Level of 175.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	50	0	0	0	50	0	0	0	0	0	0
0	0	0	0	0	0	0	50	0	0	0	0
0	50	0	0	0	0	0	0	50	0	0	0
0	0	50	0	0	0	0	0	0	0	0	0
0	0	0	50	0	0	0	0	50	0	0	0
0	0	0	0	50	0	0	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(f). Threshold Level of 200.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	50	0	50	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	50	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	50	0	0	0	0
0	50	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	50	0	0	50	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(g). Threshold Level of 225.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Figure 15(h). Threshold Level of 250.

Hardware—Software Relationship

The Sobel gradient is a program that manipulates data over a 3 × 3 processing window. In our treatment we have focused on the movement of relevant data in a 3 × 3 neighborhood around a defined center point which we named point *E*. In fact, the GAPP array processed the entire 12 × 12 image concurrently such that the Sobel gradient was computed at every 3 × 3 neighborhood in the image. Thus, after we executed the 581 cycles required to complete the algorithm, we were finsihed with the entire image. From a software standpoint, the fact that the GAPP chip allows an entire image, or portion of an image, to be processed in parallel allows software designers to design hardware-independent software. In this case, hardware-independence refers to the fact that we can change the hardware configuration in a number of ways and the software that we have discussed above remains the same. Thus, we could run the same Sobel gradient algorithm on a 24 × 24 or 24 × 12 array of processing elements and the results would be the same if the images being processed were also 24 × 24 or 24 × 12 respectively. This relieves the software designer from having to rewrite programs every time the hardware configuration changes.

Appendix A. GAPP Program Code for Sobel Edge Detection

```
100  image A:0:7 ;          /* A is input */
110  image R:8: 15;       /* 8 bit resolution gray scale output */
120  int T = 60 ;           /* T is trial constant for performing threshold */
130  int i;
140  int j;
150  int n;
160
170  /* EDGE (A,R,T)   Two's complement arithmentic */
180  main ()
190  {
200  image B:9;      /* scratch */
210  image C:10;     /* scratch */
220  image X: 11;    /* X-sobel and absolute value */
230  image Y: 11;    /* Y-sobel and absolute value */
240  image Z: 12:    /* magnitude approximation  */
250  image flag: 1;  /* threshold result flag        */
260
270  image xF: 1;
280  image xC: 11;
290  image xD: 11, xT: 1;
300      {
310          /* Compute the Y-Sobel */
320          C:= 0 ;       /* Compute first partial sum */
330          n = size(A) ;
340          i  = 0 ;
350          while (i < n)
360          {
370               A:i EW := RAM;
380               A:i NS := RAM  EW := E;
390               B:i RAM := SM   C := CY;
400               i = i + 1;
410          }
420          B:n RAM := SM  C := 0;   /* sign extension */
430          /* Compute second partial sum */
440          i = 0;
450          while (i < n + 1)
460          {
470               B:i EW := RAM;
480               B:i NS  := RAM   EW := W;
490               C:i RAM := SM     C := CY;
500               i = i + 1;
510          }
520          C:n+1  RAM := SM   C := 0;                /* sign extension */
530          /* Compute difference */
540          i = 0;
550          while (i < n + 2)
560          {
570               C:i NS := RAM;
580               NS := S;
590               C:i NS := RAM   EW := NS;
600               NS := N;
610               Y: i RAM := SM    C := BW;
620               i = i + 1;
630          }
640          Y:n+2   RAM := SM   C := 0;   /* sign extension */
650          /* Compute the X-Sobel */
```

```
660         /* Compute the first partial sum */
670         i = 0;
680         while (i < n)
690         {
700               A:i NS := RAM;
710               A:i EW := RAM  NS := S;
720               B:i RAM : = SM    C := CY;
730               i = i + 1;
740         }
750         B:n  RAM := SM  C := 0;   /* Carry extension */
760         /* Compute second partial sum */
770         i = 0;
780         while (i < n + 1)
790         {
800               B:i NS := RAM;
810               B:i EW := RAM  NS := N;
820               C:i RAM := SM    C :=CY;
830               i = i + 1;
840         }
850         C:n+1   RAM := SM   C := 0;       /* Carry extension */
860         /* Compute difference */
870         i = 0;
880         while (i < n + 2)
890         {
900               C:i  EW := RAM;
910               EW := W;
920               C:i EW := RAM    NS := EW;
930               EW := E;
940               X:i RAM := SM    C := BW;
950               i = i + 1;
960         }
970         X:n+2   RAM := SM;       /* sign extension */
980   }
990   /* The following functions are MAGNITUDE */
1000  /* MAGNITUDE (A,B,R) */
1010        /* Absolute value of X */
1020  {                       /* Perform bitwise invert on X if MSB = 1 */
1030        n = size (X);
1040        X:n-1   C:= RAM   NS := 0;       /* MSB */
1050        i = 0;
1060        while (i < n)
1070        {
1080              X:i EW := RAM;
1090              X:i RAM := SM;         /* SM = C xor EW */
1100              i = i + 1;
1110        }
1120        /* Invert if MSB = 1 */
1130        i = 0;
1140        while (i < n)
1150        {
1160              X:i EW := RAM;
1170              X:i RAM :=SM   C:=CY;          /* ignore final carry */
1180              i = i + 1;
1190        }
1200  }
1210  /* Absolute value of Y */
1220  {                       /* Perform bitwise invert on Y if MSB = 1 */
1230        n = size (X);
1240        Y:n-1   C:= RAM    NS :=  0;          /* MSB */
```

```
1250        i = 0;
1260            while (i < n)
1270            {
1280            Y:i EW := RAM;
1290            Y:i RAM := SM;          /* SM = C xor EW */
1300            i = i + 1;
1310        }
1320        /* Add "1" if MSB = 1 */
1330        i = 0
1340        while (i < n)
1350        {
1360            Y:i EW := RAM;
1370            Y:i RAM := SM   C :=CY;         /* ignore final carry */
1380            i = i + 1;
1390        }
1400    }
1410
1420    /* MAGNITUDE(A,B,R) */
1430    C:=0;
1440    for ( i=0; i < size(X) -1; i++)
1450    {
1460    X:i   NS: = RAM;
1470    Y:i   EW: = RAM;
1480          C:=BW;
1490    }
1500    xF: RAM:=C;
1510    /* Use MUX operation D = X*xF + Y*~xF = min(X,Y) */
1520    {
1530          xF: NS:=RAM;
1540          for ( i=0; i<size(X)-1; i++ )
1550          {
1560              X:i  EW:=RAM   C:=0;
1570              C:=CY;
1580              xT:    RAM:=C   C:=0;
1590              Y:i   EW:=RAM;
1600              C:=BW;
1610              xT:        NS:=C        EW:=RAM        C:=1;
1620              xF:        C:=CY        NS:=RAM;
1630              xD:i       RAM := C;
1640          }
1650    }
1660    /* Use MUX operation C = Y*xF + X*~xF =max (X,Y)   */
1670    {
1680          xF: NS:=RAM;
1690          for ( i=0; i<size(X)-1; i++ )
1700          {
1710              Y:i  EW:=RAM  C:= 0;
1720                  C:=CY;
1730                  xT:    RAM:=C    C := 0;
1740                  X:i  EW:=RAM;
1750                  X:=BW;
1760                  xT:    NS:=C   EW:=RAM   C:=1;
1770                  xF:    C:=CY   NS:=RAM;
1780                  xC:i   RAM:=C;
1790          }
1800    }
1810    /* Add C + C/2 */
1820    {
1830          xD:0  C:=RAM;       /* round by adding lsb */
1840          for (i=0; i<size(X)-2; i++)
```

```
1850                {
1860                    xC:i        NS:=RAM;
1870                    xD:i + 1   EW:=RAM;
1880                    Z:i   RAM:=SM   C:=CY;
1890                }
1900                xC:i      NS:=RAM EW:=0;
1910                Z:i       RAM:=SM   C:=CY;
1920                Z:i+1     RAM:=C;
1930            }
1940
1950            i = 0;
1960            C:= 0;
1970            while (i < size(Z))
1980            {
1990                /* detect "i"th bit value in T, load it in C
2000                                        and save previous borrow in NS     */
2010                if(T & (1 << i)) C := 1   NS :=C; /* bitwise 'and' the
2020                                                value T with left shifted "1" */
2030                else C := 0   NS := C;
2040                /* load image in EW, put threshold value into NS and
2050                                        previous borrow back in C      */
2060                Z:i   EW := RAM   NS := C   C := NS;
2070                C := BW;
2080                i = i + 1;
2090            }
2100            flag:   RAM := C;
2110
2120            /* Flag contains the binary edge image. R is produced below
2130                which restores the gray-scale component to the edge image.   */
2140
2150            /* flag anded with image Z and output to R */
2160            {
2170                j = size(A);
2180                flag: NS := RAM;
2190                A:0    EW := RAM;
2200                C := 0;
2210                i = 0;
2220                while (i < j)
2230                {
2240                    A:i+1 C := CY   EW := RAM;
2250                    R:i RAM := C   C := 0;
2260                    i = i + 1;
2270                }
2280            }
2290    }
```

References

1. L.S. Davis, "A Survey of Edge Detection Techniques," Computer Graphics and Image Processing, vol. 4, pgs. 248-270, 1975.

2. Y. Hara, N. Akiyama, and K. Karasaki, "Automatic Inspection System for Printed Circuit Boards," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, No. 6, pgs. 623-630, November 1983.

3. A. Rosenfeld, and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Transactions on Computers, vol. C-20, No. 5, pgs. 562-569, May 1971.

4. S. Tanimota, "Programming Techniques for Hierarchical Parallel Image Processors," *Multicomputers and Image Processing, Algorithms and Progarms,* K. Preston, and L. Uhr, Eds, Academic Press, New York 1982, pgs. 421-429.

5. T. F. Schatzki, A. Grossman, and R. Young, "Recognition of Agricultural Objects by Shape," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, No. 6, pgs. 645-653, November 1983.

6. E. M. Riseman, and M. A. Arib, "Computational Techniques in Visual Segmentation of Static Scenes," Computer Graphics and Image Processing, vol. 6, pg. 221-276, 1977.

7. E. Cloud, and W. Holsztynski, "Higher Efficiency For Parallel Processors," IEEE Southcon Proceedings, January 1984.

GAPP APPLICATION NOTE NUMBER 4

GAPP Chip Performs MxN Correlation on Binary Images

Introduction

The parallel architecture of the GAPP™ processor chip offers unmatched performance in many fields including: Pattern and Speech Recognition, Machine Vision and Automatic Inspection, Parallel Data Processing, Image Processing, and Associative Processing. In this application note we will focus on an example of a "sliding window" function in one of the most commonly used operations in image processing: the mxn binary correlation.

Here we will perform a 3x3 binary correlation and operate on a 3x3 neighborhood of pixels within the image:

3x3 Neighborhood =

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

using the following array of correlation coefficients:

Correlation Mask =

1	1	0
1	1	0
1	1	0

Applying this correlation mask on the 3x3 neighborhood above gives the following equation for the correlation of the image with the mask about point *e:*

$$Corr(e) = a + b + c' + f' + g + h + i' \quad [1]$$

where a correlation score of 9 indicates perfect correlation between the mask and the image while a correlation score of 0 indicates perfect negative correlation.

A system implemented using GAPP devices can perform the 3x3 correlation on each 3x3 neighborhood of pixels in an image simultaneously. The ability to process images on an image by image basis, instead of on a pixel by pixel basis as is the case with conventional processing approaches makes the GAPP chip orders of magnitude faster in performing this operation than systems implemented using conventional processing approaches. In addition, the ability of the GAPP processor to perform general processing allows the system designer high performance without having to trade-off flexibility as occurs with dedicated and special purpose processors. Consequently, the system designer has the best of *both* worlds: high performance *and* flexibility!

GAPP Correlation Algorithm

Appendix A shows an algorithm written for the GAPP processor that performs the 3x3 correlation. For the purposes of this discussion, we use 12x12 pixel image as our input (Figure 1). Because the GAPP processor contains 72 processing elements, arrayed in a 12x6 configuration, we can use just two GAPP chips to implement a 12x12 array of processing elements which will then allow us to process the entire 12x12 pixel image in a single pass. To process larger images, the system designer has two basic options: (a) implement a system using more GAPP chips, or, (b) use a frame buffer to load portions of the image 12x12 pixels at a time into the GAPP subsystem.

The program shown in Appendix A was developed and executed on the GAPP PC Development System (NCR-45GDS1) which runs on IBM and IBM compatible personal computers. The code was written in a language called the GAPP Algorithm Language (GAL™) which is a subset of the C programming language. "Image" variables are used to assign names to a set of locations within GAPP RAM in each pricessing element. For example, the statement on lines 100-130 of the program, "image A:0:0:, B:1:1, T:2:2, C:3:11," assigns RAM location 0 to variable A, RAM location 1 to variable B, RAM location 2 to variable T, and RAM locations 3 through 11 to variable C. Variable A is the input binary image, variable B is used as a scratchpad for sliding data around in the 3x3 window, variable T is used as a scratch location to permit reads and writes from GAPP RAM on the same cycle, and variable C is the output of the correlation function.

Although we will be performing the correlation operation on a 12x12 pixel image, in the following discussion we will focus on a 3x3 neighborhood about point *e* as shown earlier. Since the GAPP processor will perform this operation on each 3x3 neighborhood simultaneously, we can focus on a single 3x3 neighborhood with the knowledge that each 3x3 neighborhood in the 12x12 array will be performing the same operation. Thus, we see that regardless of the overall processor element array size, from an algorithmic or software standpoint the relevant array is the 3x3 neighborhood.

Initial Data Movements

The first task that we perform is to clear GAPP RAM locations 1 through 11. This is done in lines 210 through 230. On line 210 we set the **C** register within each GAPP processing element to 0. We then enter a 'for' loop in which we sequentially copy the contents of the **C** register into RAM locations 1 to 11.

In lines 250 and 260 we copy the input image to our scratchpad location, image variable B. In line 250 we load the input from GAPP RAM location A, to the **C** register within each processing element. In line 260 we copy the contents of the **C** registers into GAPP RAM location 1.

On line 280 we perform our initial shift operations on the input data. On line 320 we load image variable B into the **NS** registers within the processing elements. Looking at our 3x3 neighborhood around point *e:*

INSTRUCTION: B: **NS** := RAM;

NS =

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

On line 330, we load the contents of the **NS** registers into the **EW** registers.

The 'for' loop in lines 340 and 350 slides the data one column to the right:

INSTRUCTION: **EW** := W;

EW =

-	*a*	*b*
-	*d*	*e*
-	*g*	*h*

where a '—' indicates a data value that is not relevant to the computation of the correlation function in the 3x3 neighborhood about point *e* (although relevant to the correlation function in adjacent 3x3 neighborhoods).

On line 360 we load the contents of the **EW** registers into the **NS** registers. On lines 370 and 380 we enter a 'for' loop that shifts the data one row upward:

INSTRUCTION: **NS** := S;

NS =

—	*d*	*e*
—	*g*	*h*
—	—	—

On line 390 we copy the contents of the **NS** registers to the **C** registers. On the next instruction (line 400) we transfer the contents of the **C** registers to the GAPP RAM location assigned to image variable B:

INSTRUCTION: B: RAM := **C**;

RAM(B : 0) =

—	*d*	*e*
—	*g*	*h*
—	—	—

0	1	1	1	0	1	1	1	0	0	1	0
1	1	1	0	1	1	1	1	0	0	0	1
0	0	1	1	0	1	0	1	0	1	1	1
1	0	0	1	1	0	1	1	1	1	0	1
1	1	1	1	0	0	0	0	1	0	0	0
1	1	0	0	1	0	1	0	0	1	1	1
0	1	0	0	1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	1	0	0	0	1
1	0	0	1	1	1	0	1	0	1	1	0
1	1	1	1	1	0	0	1	1	1	0	1
1	1	1	1	0	1	1	1	1	1	1	0
0	0	0	0	1	0	0	0	0	1	1	0

Figure 1. Input Image Data

Beginning the Loop

In the next phase of the program we enter a nested 'for' loop which processes each of the 9 coefficients in the 3x3 convolution mask. First we set integers k, and dl to 1 in lines 430 and 440. Integer k tells us which column we are on within a given row, integer dl tells us which direction we will be sliding the window in the x-direction. When k is equal to 3, we slide the window to the left, if dl is equal to −1 then we slide the window to the right.

We first enter a 'for' loop that cycles us through the rows in the 3x3 convolution grid (line 450). For the first iteration i = 0. We then enter an inner 'for' loop that cycles us through the columns along each row (line 470). For the first interation j is also equal to 0. Below we trace the execution of the first iteration.

First Iteration (i=0, j=0)

1. On lines 580, 590, and 600 we set dx=1, dy=0, and k=2 respectively.

2. Evaluating the expressions on lines 780 and 790 we get s=0, and r=1

3. Load image B into **EW** registers and set **C** registers to 0 (line 840):

EW =

–	*d*	*e*
–	*g*	*h*
–	–	–

4. Enter 'for' loop to bit serially process each bit of the resultant image, image C. First, retrieve the LSB of image C (line 980):

INSTRUCTION: C : 0 **NS** := RAM; (ix = 0)

NS =

0	0	0
0	0	0
0	0	0

add the current contents of the **EW** registers, and store sum in RAM (line 990):

INSTRUCTION: C : 0 RAM := SM **C** := CY
EW := 0;

RAM(C : 0) =

–	*d*	*e*
–	*g*	*h*
–	–	–

propogate carry (in **C** registers) and continue addition operation over all bits in image C.

5. Shift Operation. Set x=dx=1, and y=dy=0 (lines 1050 and 1060). On line 1090 load current image variable B into **NS** registers. Then, on line 1120, copy the contents of **NS** registers into **EW** registers. On line 1170:

INSTRUCTION: **EW** = E ;

EW =

d	*e*	*f*
g	*h*	*i*
–	–	–

6. Copy contents of **EW** registers to **NS** registers (line 1270). Then copy contents of **NS** registers to **C** registers (line 1460). Finally, store as new image B (line 1470):

RAM (B : 0) =

d	*e*	*f*
g	*h*	*i*
–	–	–

Second Iteration (i=0, j=1)

1. Set k=3 on line 600

2. Evaluating lines 780 and 790 gives us s=0, and r=1

3. Load current image B into **EW** registers (line 840):

EW =

d	*e*	*f*
g	*h*	*i*
–	–	–

4. Bit serially process each bit of the current accumulated value within image C. For simplicity, we will look at the value of image C for point *e* only (line 870):

INSTRUCTION: C : 0 **NS** := RAM; (ix = 0)
NS = LSB (Image C) = g

add the current contents of the **EW** registers, and store sum in RAM (line 880):

RAM(C : 0) = LSB (*g* + *h*)

C = Carry (*g* + *h*)

propogate carry and continue addition bit serially until all bits of image C have been processed.

5. Shift Operation. After executing line 1170 this time:

EW =

e	*f*	–
h	*i*	–
–	–	–

6. Save new image B (line 1470):

RAM(B : 0) =

e	*f*	–
h	*i*	–
–	–	–

Third Iteration (i=0, j=2)

1. Set dl = −dl = −1 (line 510), dx = 0 (line 520), dy = −1 (line 530), and k = 1 (line 540)

2. Evaluate lines 780 and 790: s = 0, r = 0

3. Invert image B; then add inverted image B' (i.e. the logical complement of B) to accumulated value in image C. First, load current image B into the **EW** registers, and set the **C** registers to 1, and the **NS** registers to 0 (line 940):

EW =

e	*f*	–
h	*i*	–
–	–	–

4. Put image B′ into **EW** registers (line 950):

EW =

e′	*f′*	–
h′	*i′*	–
–	–	–

5. Bit serially process each bit of the current accumulated value within image C. Again, looking point *e* only we have after executing the instruction on line 980:

NS = LSB (Image C) = LSB (g + h)
(ix = 0)

We then proceed to add the contents of the **EW** registers which, after executing instruction 990, yields:

RAM(C : 0) = LSB (g + h + i')

We propagate the carry as we process each bit of the accumulated result, image C until all the bits have been processed.

6. Shift Operation. This time it is slightly different. In this case, we load image B into the **NS** registers and then execute line 1420:

NS =

b	*c*	–
e	*f*	–
h	*i*	–

7. Save new image B (line 1470):

RAM(B : 0) =

b	*c*	–
e	*f*	–
h	*i*	–

Fourth Iteration (i = 1, j = 0)

1. Set dx = dl = −1, dy = 0, and k = 2 (lines 580, 590, and 600)

2. Evaluating lines 780 and 790 gives us s = 0, r = 0

3. Load current image B into **EW** registers and invert image B as before (line 950):

NS =

b′	*c′*	–
e′	*f′*	–
h′	*i′*	–

4. Bit serially process each bit of the accumulated result in image C (line 980):

NS = LSB(Image C) = LSB(g + h + i')
(ix = 0)

After instruction 990:

RAM(C:0) = LSB(g + h + i' + f')

5. Shift Operation. On this iteration we also have a somewhat different shift procedure. First, we set x = dx = −1, and y = dy = 0. In this case we load image B into the **NS** register as before (line 1090), but we then put image B into the **EW** registers (instruction 1120) and execute line 1240:

EW =

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

6. We save the above as the new image B (line 1470):

RAM(B : 0) =

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

Fifth Iteration (i=1, j=1)

1. Set dx = dl = −1, dy = 0, k = 3
2. Evalute lines 780 and 790: s = 0, r= 1
3. Load current image B into **EW** registers and add to accumulated correlation score (line 870):

   NS = LSB($g + h + i' + f'$) (ix = 0)

   After instruction 880:

   RAM(C : 0) = LSB($g + h + i' + f' + e$)

4. Shift Operation. Set x = dx = −1, and y = dy = 0. Load image B into the **NS** register as before (line 1090), and then put image B into the **EW** register (instruction 1120) and execute instruction 1240:

EW =

–	*a*	*b*
–	*d*	*e*
–	*g*	*h*

5. Save the new image B (instruction 1470):

RAM(B : 0) =

–	*a*	*b*
–	*d*	*e*
–	*g*	*h*

Sixth Iteration (i = 1, j = 2)

1. Set dl = −dl = 1, dx = 0, dy = −1, and k = 1 (lines 510 through 540)
2. Evaluate lines 780 and 790: s = 0, r = 1
3. Load current image B into **EW** registers and add to accumulated correlation score (line 870):

   NS = LSB($g + h + i' + f' + e$)

   After executing statement 880:

   RAM(C : 0) = LSB($g + h + i' + f' + e + d$)

4. Shift operation. Set x = dx = 0, and y = dy = −1. Load image B into the **NS** register (line 1090), and execute instruction 1420:

NS =

–	–	–
–	*a*	*b*
–	*d*	*e*

5. Save the new image B (instruction 1470):

RAM(B : 0) =

–	–	–
–	*a*	*b*
–	*d*	*e*

Seventh Iteration (i = 2, j = 0)

1. Set dx = d1 = 1, dy = 0, and k = 2 (lines 580 to 600)
2. Evaluate s = 0, and r = 1
3. Load current image B into **EW** registers and add to accumulated correlation score (line 870):

   NS = LSB($g + h + i' + f' + e + d$)

   After instruction 880:

   RAM(C : 0) = LSB($g + h + i' + f' + e + d + a$)

4. Shift operation. Set x = dx = 1, and y = dy = 0. Load image B into the **NS** register and then load image B into the **EW** register, and execute line 1170.

EW =

–	–	–
a	*b*	*c*
d	*e*	*f*

5. Save new image B (instruction 1470):

RAM(B : 0) =

–	–	–
a	*b*	*c*
d	*e*	*f*

Eighth Iteration (i = 2, j = 1)

1. Set dx = dl = 1, dy = 0, and k = 3

2. Evaluate s = 0, and r = 1

3. Load current image B into **EW** registers and add to accumulated correlation score (line 870):

   NS = LSB($g + h + i' + f' + e + d + a$)

   After executing instruction 880:

   RAM(C : 0) = LSB($g + h + i' + f' + e + d + a + b$)

4. Shift Operation. Set x = dx = 1, and y = dy = 0. Load image B into the **NS** register and then load image B into the **EW** register, and execute line 1170:

   EW =

–	–	–
b	*c*	–
e	*f*	–

5. Save new image B (instruction 1470):

   RAM(B : 0) =

–	–	–
b	*c*	–
e	*f*	–

Ninth Iteration

1. Set d1 = −d1 = −1, dx = 0, dy = −1, k = 1

2. Evaluate s = 0, and r = 0

3. Load current image B into **EW** and invert image B (lines 940 and 950). Then add to accumulated correlation score (line 980):

   NS = LSB($g + h + i' + f' + e + d + a + b$)

   After executing line 990:

   RAM(C : 0) = LSB($g + h + i' + f' + e + d + a + b + c'$)

[end of program]

Discussion

The results of the correlation algorithm given the input shown in Figure 1 are displayed in Figure 2. The program shown in Appendix A was able to derive this output from the 3x3 binary correlation in 230 cycles. At 10MHz per cycle the GAPP processor is able to perform this function in a mere 23us!

To expand on our discussion, let us raise four issues worth mentioning concerning the program:

1. Specifying a Different Set of Mask Coefficients

2. Varying the Size of the Convolution Window

3. Using the "enable" Variable

4. Modifying the Order in which Coefficients are used.

1. Changing the Mask Coefficients

The mask coefficients in this example were stated as a hexadecimal number. The value was determined by the mask that we wished to implement and the method of traversal that was used to slide the 3x3 window over the input image. In our approach, we started with the lower left hand corner coefficient, or position (3,1) using matrix notation, and moved to position (3,1) using matrix notation, and moved to position (3,2) to (3,3), to (2,3), to (2,2), to (2,1), to (1,1), to (1,2), and finally to (1,3). Given our method of traversal about the 3x3 correlation mask, we see that the specified input mask, cmask, is a representation of the 3x3 coefficient mask with mask coefficient (3,1) being the LSB of cmask, and coefficient (1,3) being the leading bit of cmask. With this knowledge, a different mask can be specified according to the needs of the application. For instance, the 3x3 correlation mask:

0	1	0
1	0	1
0	1	0

can be represented by the hex word: cmask = 0x0aa.

2. Varying Correlation Mask Size

In this example we used a 3x3 mask of correlation coefficients. This was determined on line 190 of the program where we set n (the number of columns), and m (the number of rows) to 3. Thus, the size of the correlation mask is set by this declaration. One must be mindful that if the coefficient array size is modified, the matrix of coefficients may be altered. For instance, if our specified value for cmask, 0x0f3 is applied to a 4x3 coefficient mask we get:

0	0	0
1	1	0
1	1	0
1	1	0

However, for a 3x4 matrix, we get:

0	0	0	0
1	1	1	1
1	1	1	1

which may not be the desired correlation window.

3. Using the "Enable" Variable

You will notice a variable named "enable" in the program. This variable can be used to mask don't cares in the mxn window. This is particularly useful if we wanted to use, for example, a diamond shaped correlation mask:

x	1	x
1	1	1
x	1	x

where x indicates a don't care. The enable flag could be used to mask out any data within the correlation window that we did not want to correlate with the image. For example, the above correlation mask could be generated if cmask = 0x1ff, and enable = 0x145. Using enable in this situation saves processing time that would otherwise be wasted evaluating meaningless mask coefficients.

4. Modifying the Ordering of Coefficient Evaluation

As mentioned in discussion module 1 above, we followed a zig-zag pattern of traversal through the 3x3 correlation mask starting at mask location (3,1). The program, however, is general enough to support any arbitrary order of coefficient evaluation by changing dx, d1, and dy. For instance, if we defined our initial starting point to be the (1,3) position instead of the (3,1) position, and changed our initial value of d1 from 1 to -1, and further changed dy $= -1$ to dy $= 1$ on line 510, then we would traverse the 3x3 correlation mask in the opposite direction. As a consequence, with cmask as given we would have the following correlation mask:

0	1	1
0	1	0
0	1	0

Using the same input image as in Figure 1, we re-ran our correlation algorithm with the changes noted above. The output is shown in Figure 3. Likewise, other mask traversal patterns can also be accomodated.

Conclusion

The GAPP processor offers a high performance alternative for processing sliding window functions such as the 3x3 correlation that was treated in this application note. In our treatment we were able to focus on a 3x3 neighborhood surrounding a defined center point, *e*. In addition, the 3x3 correlation was computed for every 3x3 neighborhood in the entire image in 230 cycles or 23us. A full 21x21 binary correlation can be done in 1.1ms using the GAPP processor. From the programmers' perspective, the ability to focus on a *relevant* processing element neighborhood to the exclusion of the overall processing element array decouples the programmer to a large extent from the specifics of the hardware design. If we were to process a 48x48 pixel image using 32 GAPP devices, the same algorithm as used with our 12x12 processing element array could be used with only modifications to the I/O drivers.

3	4	6	4	4	5	5	7	4	2	5	5
4	3	6	7	3	5	5	8	5	2	4	6
7	4	4	5	5	5	4	7	5	5	3	5
6	4	3	7	6	4	3	4	5	6	4	4
6	7	4	4	7	3	4	4	4	6	4	3
4	7	6	3	6	3	5	3	3	6	4	4
5	8	5	1	6	4	6	6	3	5	3	5
5	7	3	2	6	6	5	6	4	5	4	4
6	7	3	4	7	7	3	6	5	5	4	3
5	6	4	6	6	6	3	5	5	6	7	3
4	5	5	5	6	5	3	4	4	6	7	4
2	4	5	6	4	4	5	6	4	3	8	5

Figure 2. Correlation Score for Input Data

5	6	4	4	6	5	5	3	2	6	5	3
4	7	6	3	5	7	5	4	1	6	6	4
3	4	6	5	5	5	6	5	3	5	5	7
4	4	7	5	2	4	5	6	5	4	4	6
6	3	4	6	3	3	6	4	6	4	4	7
6	5	2	5	4	3	5	3	7	4	4	6
5	4	1	7	6	4	6	2	5	5	5	7
5	3	3	8	6	4	5	4	4	5	4	6
6	3	5	8	5	3	5	6	3	5	4	7
7	4	6	6	4	4	5	7	5	6	3	5
6	5	5	5	4	3	5	6	6	6	3	4
6	6	5	4	4	6	5	4	4	7	4	1

Figure 3. Correlation Score for Reverse Traversal in 3x3 Correlation Matrix for Input Data

Appendix A

```
100   image  A : 0 : 0,      /*  input image */
110          B : 1 : 1,      /*  temporary value */
120          T : 2 : 2,      /*  scratch */
130          C : 3 : 11;     /*  output value */
140   main ( )
150   {
160          int cmask = 0x0f3;
170          int enable = 00; /* code values */
180          int x,y,i,ix,dx,dy,dl,r,s,j,k;
190          int n = 3, m = 3; /* size of correlation mask */
200          /* clear ram */
210          c : = 0;
220          for (i = 0; i<size(C)+2; i++ )
230          : i + 1 ram:=C;
240          /* copy A into B */
250          A: c:=ram;
260          B: ram:=c;
270
280          /* Initial SHIFT operation */
290          x=n/2;
300          y=m/2;
310          {
320          B: NS:=RAM;
330          NS:=EW EW:=NS;
340          for (i=0; i<x; i++)
350                { EW:=W; }
360          NS:=EW EW:=NS;
370          for (i=0; i<y; i++)
380                { NS:=S; }
390          C:=NS;
400          B: RAM:=C;
410          }
420          /* Begin loop */
430          k = 1;
440          dl=1;
450          for (i=0; i<m; i++)
460          {
470                for (j=0; j<n; j++)
480                {
490                      if ( k = = n)
500                      {
510                           dl = –dl;
520                           dx = 0;
530                           dy = –1;
540                           k = 1;
550                      }
560                      else
570                      {
580                           dx = dl;
590                           dy = 0;
600                           k = k+1;
610                      }
620                      /* convert mask code to value */
630                      /* represent mask code as single hex value in zig-zag pattern
640                                represent enable code in same fashion for don't care */
650                      if ( j = = 0)   { cmask = 0x0f3; enable = 00; } /* code values */
```

```
660                 if ( j = = 1)   { cmask = 0x0f3; enable = 00; }  /* code values */
670                 if ( j = = 2)   { cmask = 0x0f3; enable = 00; }  /* code values */
680                 if ( j = = 3)   { cmask = 0x0f3; enable = 00; }  /* code values */
690                 if ( j = = 4)   { cmask = 0x0f3; enable = 00; }  /* code values */
700                 if ( j = = 5)   { cmask = 0x0f3; enable = 00; }  /* code values */
710                 if ( j = = 6)   { cmask = 0x0f3; enable = 00; }  /* code values */
720                 if ( j = = 7)   { cmask = 0x0f3; enable = 00; }  /* code values */
730                 if ( j = = 8)   { cmask = 0x0f3; enable = 00; }  /* code values */
740                 if ( j = = 9)   { cmask = 0x0f3; enable = 00; }  /* code values */
750                 if ( j = = 10)  { cmask = 0x0f3; enable = 00; }  /* code values */
760                 if ( j = = 11)  { cmask = 0x0f3; enable = 00; }  /* code values */
770
780                 s = 1 & (enable >> (j+m*i));
790                 r = 1 & (cmask >> (j+m*i));
800                 if ( s = = 0)
810                 {
820                     if (r = = 1)  /* add un-inverted B value */
830                     {
840                         B : 0 ew:=ram c:=0;
850                         for ( ix = 0; ix < size(C)-1; ix++ )
860                         {
870                             C : ix ns:=ram;
880                             C : ix ram:=sm c:=cy ew:=0;
890                         }
900                         C:ix ram:=c;
910                     }
920                     else /* add inverted B value */
930                     {
940                         B : 0 ew:=ram c:=1 ns:=0;
950                         T : 0 ew:=ram ram:=sm c:=0;
960                         for (ix=0; ix< size (C)-1; ix++ )
970                         {
980                             c:ix ns:=ram;
990                             c:ix ram:=sm c:=cy ew:=0;
1000                        }
1010                        c:ix ram:=c;
1020                    }
1030                }
1040                /* SHIFT operation */
1050                x=dx;
1060                y=dy;
1070                if ( j+3*i < m*n-1 )
1080                {
1090                    B: NS:=RAM;
1100                    if ( x ! = 0)
1110                    {
1120                        NS:=EW EW:=NS;
1130                        if (x > 0)
1140                        {
1150                            for ( ix=0; ix<x; ix++ )
1160                            {
1170                                EW:=E;
1180                            }
1190                        }
1200                        else
1210                        {
1220                            for ( ix=x; ix<0; ix++ )
```

```
1230                            {
1240                                EW:=W;
1250                            }
1260                        }
1270                        NS:=EW EW:=NS;
1280                    }
1290                    if ( y != 0)
1300                    {
1310                        if (y>0)
1320                        {
1330                            for (ix = y; ix>0; ix--)
1340                            {
1350                                NS:=S;
1360                            }
1370                        }
1380                        else
1390                        {
1400                            for ( ix=y; ix<0; ix++ )
1410                            {
1420                                NS:=N;
1430                            }
1440                        }
1450                    }
1460                    C:=NS;
1470                    B: RAM:=C;
1480                }
1490            }
1500        }
1510    }
```

GAPP APPLICATION NOTE NUMBER 5

Data Input/Output Techniques for the Geometric Arithmetic Parallel Processor (GAPP)

Introduction

One of the most important aspects of designing with the GAPP parallel array processor (NCR45CG72) is the method of getting data into and out of the array of GAPP processors. When designing a real time image processing system, the system is generally required to operate fast enough to accept 30 frames of data a second. This rate allows the array of GAPP processors to execute 333,000 instructions to process the image during the video time frame (the GAPP processor executes one instruction every 100 nanoseconds). However, processing a typical high resolution image (512 by 512 pixels with 8 bits of gray-scale), imposes the requirement of having to load over 60 million bits of data into the processor array every second. The system designer must be careful to use an I/O method which is both fast enough to satisfy the throughput requirements of the system and is efficient enough so that the I/O method does not interfere with the execution of the image processing algorithm. Systems designed to do other types of numerical processing may not be limited to a 33.3 millisecond frame rate, but they will have similar requirements for throughput and efficient I/O.

Data input to a system comprised of GAPP processors will not be in the format required by the GAPP processor array. Data from an analog-to-digital converter, or a system bus, is usually a stream of values (arranged word-serial, bit-parallel) with anywhere from 6 to 16 bits per value. The CMS inputs at the south edge of the GAPP array provides an easy way to load data into the array. The most efficient way is to present a row of bit values to these inputs and use the "cm:=cms" instruction to shift the bits into the south edge of the array. However, the GAPP array expects the data to be in a bit-serial, word-parallel format. Therefore, some hardware which is external to the array of GAPP processors must be used to store a row of data values and serially present each bit position to the CMS inputs of the array. The data must physically make a 90 degree change of direction to be reformatted from bit-parallel to bit-serial, therefore the reformatting process is called "corner-turning."

One method of corner-turning uses a corner-turn line buffer (also referred to as a "CTLB") constructed of GAPP devices as the external I/O hardware. The I/O process splits the corner-turn process between the corner-turn line buffer and the main array of GAPP processor elements (see next page). This method requires that several instruction cycles be "stolen" from the main array and be used for I/O rather than computation. For large arrays, and for data with a large number of bits per value, use of this method can potentially interfere with the main algorithm and may not allow the execution of complex algorithms at video frame rates.

A second method for higher performance I/O, called RAM-based corner-turning, is described in section 3. This method of corner-turning uses a corner-turn line buffer constructed of GAPP devices, as does the first method, but it also makes use of the GAPP RAM in the reformatting process. Almost all of the corner-turn reformatting is done in the corner-turn line buffer, so the number of instruction cycles "stolen" from the main array is significantly reduced.


```
begin
  for (i = 1) to (i = number of rows in the main array) do
    for (j = 1) to (j = number of columns in the main array) do
      Shift data value into corner-turn line buffer from east side.
      end  /* for j */
    Move data in corner-turn line buffer from EW registers to CM registers.
    for (k = 1) to (k = number of bits in a data value) do
      (1)  Move bit position k - 1 of data values already in the
           main array from RAM to the CM registers.
      (2)  Shift the contents of the CM registers in both the
           main array and corner-turn line buffer north one position.
      (3)  Move bit position 'k-1' of data in the main array from
           the CM registers in the main array to RAM.
      end  /* for k */
    end  /* for i */
  end  /* begin */
```

Figure 1. Descriptive Algorithm for Shift Register Corner-turning.

Shift Register Corner-Turning

This corner-turning method is summarized by the general algorithm in Figure 1. The corner-turn line buffer and main array are assumed to be connected as shown in Figure 2. A "row" is a set of processor elements which are in an east/west line, and a "column" is a set of processor elements which are in a north/south line. For data values with N-bits, bit position 0 is the least significant bit, and bit position N-1 is the most significant bit.

The size (in rows and columns of processor elements) of the main array and corner-turn buffer is determined as follows. The number of rows and columns in the main array is arbitrary (as long as they are a multiple of the number of rows and columns per chip), and is generally dictated by the requirements of the algorithm being executed in the main array, which we will call the "main algorithm." The number of columns in the corner-turn line buffer must be the same as the number of columns in the main array. The CM communication bus lines of the main array and coner-turn line buffer are connected in a "cylindrical" fashion, which means that each CMN (CM North) output of the main array is connected to the CMS (CM South) input of the same column in the corner-turn line buffer. Likewise, each CMN output of the corner-turn line buffer is connected to the CMS input of the same column in the main array. The number of rows in the corner-turn line buffer must be equal to, or greater than, the number of bits in a data value. Usually, there are less than 12 bits per value, and the corner-turn line buffer is a single east/west line of chips, but more chips may be used for larger data values.

The corner-turn operation is shared between the main array and the corner-turn line buffer. At any given time in the corner-turn process, the lines of data which have already been input are stored in the southmost rows of the main array. After a new line of data is brought into the corner-turn line buffer, it is shifted into the southmost row of the main array, and the other lines of data are shifted north one row.

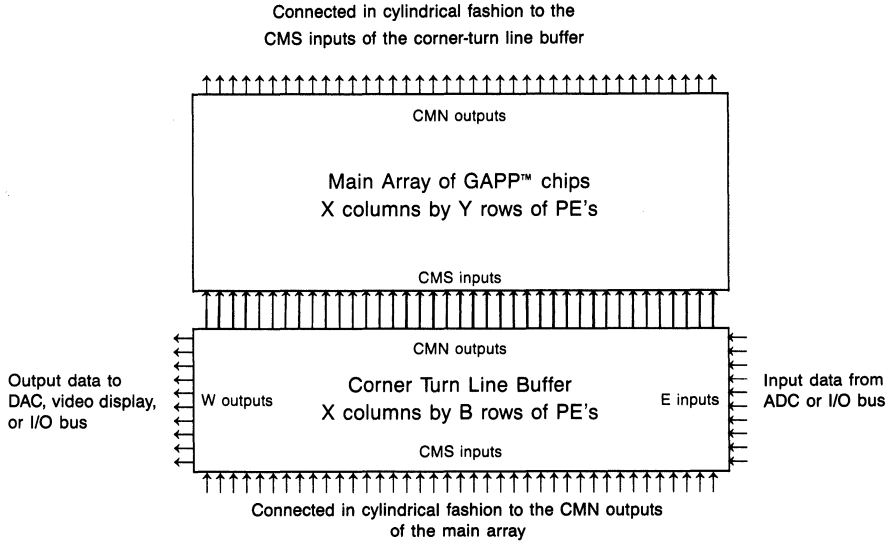


Figure 2. Corner-turning: Main Array and Corner-turn Line Buffer Arrangement.

Note: the N, S, E, and W I/O lines of the main array are not shown. The N and S lines and the E and W lines can be wrapped-around in any desired pattern as required by the algorithm executing in the main array. If no wrap-around is needed, these lines should be tied to ground or the V_{dd} rail through approximately 10KΩ resistors to prevent floating inputs and unnecessary power dissipation.

Note: the N and S I/O line of the corner-turn line buffer are not shown. These should be grounded through approximately 10KΩ resistors to prevent floating inputs.

Since the main array must share part of the reformatting process, system performance is degraded. Consider the operations marked (1), (2), and (3) in Figure 1. Each of these operations will require one instruction in an optimal implementation of a corner-turn operation. Operation (2) only requires the CM register control lines of the GAPP chip. The CM registers are primarily used for I/O and should never be needed for computation during execution of the main algorithm. Accordingly, operation (2) of the corner-turn process may be executed at the same time as an instruction of the main algorithm. However, operations (1) and (3) require the RAM control lines, which are almost always needed during algorithm execution, and so two instruction cycles must be "stolen" from the main algorithm to perform these operations. For a 512 by 512 array with 8-bit data values, this means that 8192 instruction cycles are stolen from main algorithm execution, which is 2.5% of the available processing time (assuming 30 video frames per second). The degradation gets worse as the number of bits in the data values gets larger.

Ram-Based Corner-Turning

High-performance I/O is achieved by doing most, if not all, of the corner-turn reformatting in the corner-turn line buffer. Ram-based corner-turning uses the same configuration for a main array and a corner-turn line buffer as shown in Figure 2. A general algorithm for RAM-based corner-turning is given in Figure 3.

The RAM in the GAPP processor elements is used to do almost all of the reformatting in the corner-turn line buffer. As each row of data is input to the corner-turn line buffer, it is stored in the CTLB RAM instead of being immediately shifted into the main array. When all rows of data have been stored, the least significant bit position is extracted from each row and shifted into the CMS inputs of the main array, followed by the next bit position, etc. For now, it is assumed that the corner-turn line buffer has an unlimited number of RAM addresses in which to store the rows of data. This is not actually the case when the corner-turn line buffer is constructed with GAPP devices, and this restriction forces some minor modifications of the basic algorithm which is covered on page 372.

```
begin
  for (i = 1) to (i = number of rows in the main array) do
    for (j = 1) to (j = number of columns in the main array) do
      Shift data value into corner-turn line buffer from east side.
      end  /* for j */
    Store row of data in corner-turn RAM.
    end  /* for i */
  for (k = 1) to (k = number of bits in data value) do
    for (i = 1) to (i = number of rows in the main array) do
      Read data row i in the corner-turn line buffer
      from RAM into the CM registers.
(1)   Shift the contents of the CM registers in both
      the main array and corner-turn line buffer north
      one position.
      Store the shifted data for row 'i' in the corner-turn
      line buffer from the CM registers to RAM.
      end  /* for i */
(2) Write the data for bit position 'k-1' to main array RAM
    from the CM registers.
    end  /* for k */
  end  /* begin */
```

Figure 3. Descriptive Algorithm for RAM-based Corner-turning.

```
corner_turn (dest, X, Y, B)
image     dest;          /* destination image variable */
int       X,             /* number of columns in main array */
          Y,             /* number of rows in main array    */
          B;             /* number of bits in input data    */
{
  int       i, j, k;     /* standard counter variables */
 /* ************************************************** */
 /*    Input Section: shift data values into corner-turn line buffer   */
 /* ______________________________________________________ */
   for (i = 0; i < Y; i++)
     {
      for (j = 0; j < X; j++) ew:=e ns:=0 c:=0;
        ram (:i) :=sm;
     }
/* ********************************************************************** */
/*    Output Section: shift data from corner-turn line buffer into main array    */
/* ______________________________________________________________________ */
   for (k = 0; k < B; k++)
     {
       for (i = 0; i < Y; i++)
        {
         cm := ram(:i);
         cm := cms;              /* (1) main array: cm := cms */
         ram(:i) := cm;
        }
      nop;                       /* (2) main array: ram(dest:k) := cm */
     }
}
```

Figure 4. GAL Program Code for RAM-based Corner-turning.

Implementation

Figure 4 gives the program code used to implement the RAM-based corner-turn algorithm. The routine is written in NCR-GAL™, the GAPP Algorithm Language, which combines control statements similar to the C programming language with GAPP instruction mnemonics.

The routine in Figure 4 would be executed in each of the processor elements in the corner-turn line buffer. The image variable argument to the routine, 'dest', is an image variable which determines the RAM addresses where the data frame is to be stored in the main array. The other arguments to the routine are the integer variables 'X', 'Y', and 'B', which are respectively the number of columns in the main array, the number of rows in the main array, and the number of bits in the input data.

During the execution of the corner-turn routine in the corner-turn line buffer, the main array is free to execute the main algorithm. The technical where the corner-turn line buffer executes one program, while the main array executes another, is reasonable simple to implement in hardware.

Some complications arise due to the fact that the main algorithm and the RAM-based I/O routine are not completely independent. During the operations marked (1) and (2) in Figure 3, the main array must execute certain instructions under the control of the I/O routine, which are shown in comments in Figure 4. Operation (1) requires the use of the main array's CM register control lines. It is an easy matter to have the I/O routine always control the CM registers of the main array. This effectively moves 2 bits from the main array control store to the corner-turn line buffer control store.

Since the main algorithm does not need the CM registers for computation, this method of data input/output does not interfere with the execution of the main algorithm. The CM registers should only be used in the I/O routine.

Operation (2) requires the use of the main array RAM control lines. If both the main algorithm and the I/O routine try to simultaneously execute an instruction which uses the RAM field, the conflict must be resolved in the favor of the I/O routine. This is to allow the corner-turn line buffer to keep up with the input data. Rather than to try and detect such conflicts, and since almost all main algorithm instructions will make use of the RAM field, it is a much simpler matter to "steal" an instruction cycle from the main algorithm every time operation (2) occurs in the I/O routine.

This must be done before the microcode is loaded into the control store, and is most easily accomplished with a utility which takes the two files containing the main algorithm and the I/O routine, and merges them into one file which is the microcode for the single control store. Such a merge utility must know: 1) which instructions of the I/O routine are operation (2), 2) on these instructions it must select the main array RAM address and control bits from the I/O routine, and 3) insert a NOP instruction into the main array EW, NS, and C register control bits.

Performance

Table 1 gives a summary of the performance of the RAM-based corner-turning routine shown in Figure 4. The number of instruction cycles are shown as a function of the 'X', 'Y', and 'B' arguments to the 'corner-turn' routine. To obtain the execution time, the number of instructions is multiplied by the cycle time of the GAPP chip (100ns is assumed in this application note, corresponding to a system constructed of NCR45CG72-1 chips with a 10 MHz clock rate). The total time, 'Tt', is the total number of instruction cycles needed to input a frame of data into the main array through the corner-turn buffer. This time is divided into two components, the input time, 'Tl', which is the needed to shift the data into the corner-turn buffer and store it in RAM, and the output time, 'To', which is the time needed to shift the data out of the corner-turn buffer and store it into the main array RAM. The interference time, 'Tr', is the number of instruction cycles "stolen" from the algorithm executing in the main array (this is typically less than 0.5% of 'Tt', for real-time video frame rates).

Generally, input data to a GAPP system comes from a mass storage device such as a video frame buffer. In this case, the only important item from Table 1 is 'Tt'. This value will determine the maximum array size which will allow the system to achieve its throughput requirements (usually one frame every 33.3 ms, corresponding to real-time video input). It is also important that 'Tt' be no greater than the execution time of the main algorithm, otherwise there will be instruction cycles during which the main array could be processing data, but is actually idle while waiting for the I/O routine to complete. Due to the limit of the size of the GAPP RAM in the corner-turn buffer, a maximum value of 128 for 'y' is assumed. Techniques for getting around this limit are in the next section.

Item	Number of Instructions
Total Time:	Tt = Y(X + 1) + B(3Y +1)
Input Time:	Ti = Y(X +1)
Input Time per Line:	T1 = X +1
Output Time:	To = B(3Y +1)
Main Array RAM Interference:	Tr = B

Table 1. RAM-based Corner-Turning Performance

Modifications for Greater Than 128 Video Lines

As mentioned on previous page, the routine shown in Figure 4 will not work if 'Y' is greater than 128. However, this restriction can be avoided by making some slight modifications. There are three options here, giving the system designer a tradeoff between performance and hardware complexity.

```
group_corner_turn (dest, X, Y, B)
image    dest;          /* destination image variable */
int      X,             /* number of columns in main array */
         Y,             /* number of rows in main array    */
         B,             /* number of bits in input data    */
         G;             /* number of data lines in group   */
{
  int  i, j, k, m, p;   /* standard counter variables */
  /*****************************************************************/
  /* Do for each group of G data lines & for any "leftover" lines (Y mod G) */
  /* ------------------------------------------------------------- */
      for (m = 0; m < = Y / G; m + +)
       {
        /* ********************************************** */
        /*    'p' = number of data lines in this group (usually 'G', but  */
        /*      'p' = (Y mod G) if this is the last group of data lines    */
        /* ---------------------------------------------- */
        p = (m < Y / G) ? G : (Y % G);
        /* ********************************************************** */
        /*      Input Section: shift data values into corner-turn line buffer  */
        /* ---------------------------------------------------------- */
        for (i = 0; i < p; i + +)
          {
           for (j = 0; j < X; j + +)        ew: = e ns: = 0 c: = 0;
           ram (:i); = sm;
          }
        /*********************************************************/
        /*     Output Section: shift data from corner-turn line buffer  */
        /*                       into main array.                */
        /* ----------------------------------------------------- */
        for (k = 0; (p ! = 0) && (k < B); k + +)
          {
            for (i = 0; i < p; i + +)
              {
               cm: = ram(:i);          /* (1) main array: cm: = ram (dest:k)  */
               cm: = cms;              /* (2) main array: cm: = cms           */
               ram(:i): = cm;
              }
            nop;                       /* (3) main array: ram(dest:k): = cm   */
           }
        }
}
```

Figure 5. GAL Program Code for RAM-based Corner-turning with Data Line Grouping.

Item	Number of Instructions
Total Time:	Tt = Y(X + 1) + B(3Y + P + f)
Total Input Time:	Tit = Y(X + 1)
Group Input Time:	Tig = G(X + 1)
Input Time per Line:	T1 = X + 1
Total Output Time:	Tot = B(3Y + 1)
Group Output Time:	Tog = B(3G + 1)
Main Array RAM Interference:	Tr = 2(PB + f)

notes: 'f' = 0 if Y is a multiple of G, 'f' = 1 otherwise.
'G' = number of data lines per group
'P' = integer part of Y divided by G

Table 2. RAM-based Corner-Turning Performance: Data Line Grouping

Ram-based Corner-turning with Data Line Grouping

The first option is to repeat the RAM-based corner-turning process for each group of 128 lines of data. 128 lines of data are shifted into the corner-turn line buffer (the maximum that it will hold in RAM), which are then shifted into the main array and stored in the bottom 128 rows. Then, the next group of 128 lines is shifted in, stored in the corner-turn line buffer, and shifted into the bottom 128 rows of the main array as the first group is shifted up to the next 128 rows. Thus, we have a method which employs the basic idea behind shift register corner-turning, but operates on groups of data lines to reduce the interference with the main algorithm execution.

Figure 5 gives a GAL routine to implement RAM-based corner-turning with data line grouping. Table 2 gives a summary of the performance of the RAM-based corner-turning routine shown in Figure 5. The group input time, 'Tlg', is the time needed to input one group of data lines, and the group output time, 'Tog', is the time needed to output one group of data lines to the main array. The corner-turn line buffer alternates between two phases: input of a group, and output of a group. The total input and output time, 'Tit', and 'Tot', show how much time is spent in each phase.

Ram-based Corner-turning with Line Buffer Grouping

The second option for handling a main array larger than 128 rows is to divide the main array into groups of 128 (or less) rows and assign a corner-turn line buffer to each group. Each line buffer is only responsible for data belonging to the rows of the main array in its local group.

There are two schemes for implementing this option. The same implementation that was given on page 370 can be used for each group, with all of the corner-turn line buffers under control of the same address sequencer and executing the same routine. The data to all the line buffers arrives in parallel. This means that the performance of the system will be the same as given in Table 1, with 'Y' replaced by 'H', the number of main array rows per group. The drawback to this scheme is that a frame buffer must be designed which is large enough to hold an entire frame of data, and present the data in parallel to each corner-turn line buffer. If the data acquisition time of the frame buffer is slow (such as for real-time video data), then the frame buffer will be able to present data to the corner-turn line buffers much faster than it can acquire it. In this case the frame buffer will likely be the bottleneck in the system, and the througput will only be slightly better than the option given on this page, with greatly increased hardware costs. Therefore, this scheme is probably best suited for systems processing very high speed input data, such as from infrared sensors. The high data rates coming into the system already require a frame buffer to slow the data rate down to what the GAPP array can accept.

The second scheme avoids the use of a frame buffer by having each corner-turn line buffer accept its input data as it is available. The line buffer for the first group will begin inputing its data and storing it in RAM. All other corner-turn line buffers will be inactive. When the data for the first group has been input into the first line buffer, that buffer becomes inactive and the second line buffer begins inputing its data. This continues until all line buffers have stored their data in RAM. Only one corner-turn line buffer is active at any one time. After all the corner-turn line buffers are full, each corner-turn line buffer outputs its data into its section of the main array in parallel with all the other corner-turn line buffers.

The corner-turn routine needed to implement this scheme is made up ot two parts of the 'corner-turn' routine shown in Figure 4, with the variable 'Y' replaced by 'H', the number of main array rows per group. The first part is the 'Input Section', which is executed once for each corner-turn line buffer. The second part is the 'Output Section', which is executed by all corner-turn line buffers in parallel. The performance for the scheme is summarized in Table 3. Note that the input time and main array RAM interference is very similar to the corresponding figures for RAM-based corner-turning as shown in Tabel 1 (but this assumes that a GAPP chip with more than 128 bits of RAM is available). However, the output time is much less than shown in Table 1, and is closer to the values for RAM-based corner-turning with data line grouping.

Item	Number of Instructions
Total Time:	Tt = Y(X + 1) + B(3H + 1)
Total Input Time:	Tit = Y(X + 1)
Group Input Time:	Tig = H(X + 1)
Input Time per Line:	T1 = X + 1
Output Time:	To = B(3H + 1)
Main Array RAM Interference:	Tr = B

note: 'H' = number of main array rows per group

Table 3. Ram-based Corner-Turning Performance: Line Buffer Grouping

45GDS1 - GAPP PC DEVELOPMENT SYSTEM

General Description

The GAPP PC Development System is composed of two parts. The first is a hardware board which is compatible with the IBM-PC I/O bus and contains a 12 by 12 array of processor elements implemented with two GAPP devices. The second part is a software package which allows the user to program the GAPP array in a high-level language and interactively debug a program.

Hardware Features

- 12 by 12 array of GAPP Processor Elements (PE).
- 12 byte reformatting/corner-turn array for data input/output.
- Interface to IBM standard bus with TTL circuitry.
- Three 8-bit registers for GAPP control and address interface.
- Two I/O ports for data down-load/up-load to or from GAPP array.
- Register and port addresses are switch selected.
- PE array clock is software controlled through separate I/O port.
- Printed circuit board plugs into bus connector of IBM compatible personal computers.
- Cylindrical wrap: all East and West I/O lines are horizontally connected at the left and right edges of the array: all North and South I/O lines are vertically connected at the top and bottom edges of the array.

Software Package Features

- Menu driven with screen oriented displays.
- GAL™ (GAPP Algorithm Language) compiler.
- Simple text editor for program corrections.
- Debug routines allow user to:

  single step through GAPP instructions,

  execute an entire block of GAL program statements,

  execute entire program,

  stop at any time for program corrections/ re-compilation.

- GAPP PE editor allows user to:

  up-load/change/down-load contents of each PE RAM,

  up-load/change/down-load contents of PE registers,

  store or load any of the above data to/from a data file,

  data files can be edited with the text editor.

- Runs under the VENIX/86™ (NCR45GDS1-VX) operating system on any NCR Model 4 (with hard disk) or IBM PC-XT™ compatible. Also available in an MS-DOS™ version (NCR45GDS1-MS) for IBM compatible personal computers.

Features of GAL

The GAPP Algorithm Language is a subset of the C programming language with several features added to tailor the language to the GAPP. Features of the C programming language which have been implemented are:

- All arithmetic, logical, and assignment operators.
- Int variables, can contain the values from −32768 to 32767.
- Variables defined inside of a block (within {}'s) are automatic (storage space can be re-used outside of the block).

- The *if, if...else, for,* and *while* program statements are implemented.

- Support for sub-routines and *int* functions is provided. Arguments to the subroutine are the values of the variables which are used in the sub-routine call. The values of these variables are not changed by the sub-routine.

Additional features have been added which are unique to GAL:

- A new type of variable is used to refer to GAPP RAM addresses. An *image* variable is used to refer to a set of adjacent RAM locations starting at address X, with *n* number of bits. *Image* variables are declared by one of the following program statements:

  image SCRATCH:3:7;
  image SCRATCH:5;

  The first form defines an *image* named "SCRATCH" which starts at RAM address 3 and ends at RAM address 7. The second form also defines an *image* name "SCRATCH," but only specifies the number of bits (5). The starting address of the *image* is left up to the GAL compiler.

- *Image* names can be used to specify the address portion of a GAPP instruction. The programmer must specify the name of the *image* and an arithmetic expression which gives the offset within the *image*. The compiler adds the starting address of the *image* to the arithmetic expression to determine the GAPP RAM address. An example is:

  SCRATCH:i + 3

  Either the *image* name, or the arithmetic expression may be omitted, but not both. If the expression is omitted, the compiler uses 0 for the offset; if the *image* name is omitted, the compiler uses the expression for the address.

- The function *size*() is built into the compiler; the function accepts the name of an *image* as an argument and returns the number of bits in the *image*.

- A legal GAL program statement is a GAPP instruction made up of GAPP RAM address and a list of GAPP assembler mnemonics. Examples of GAPP instructions are:

  X:l cw: = c ram: = cy;
  ram (X:1): = c;
  cw: = ram (:2);

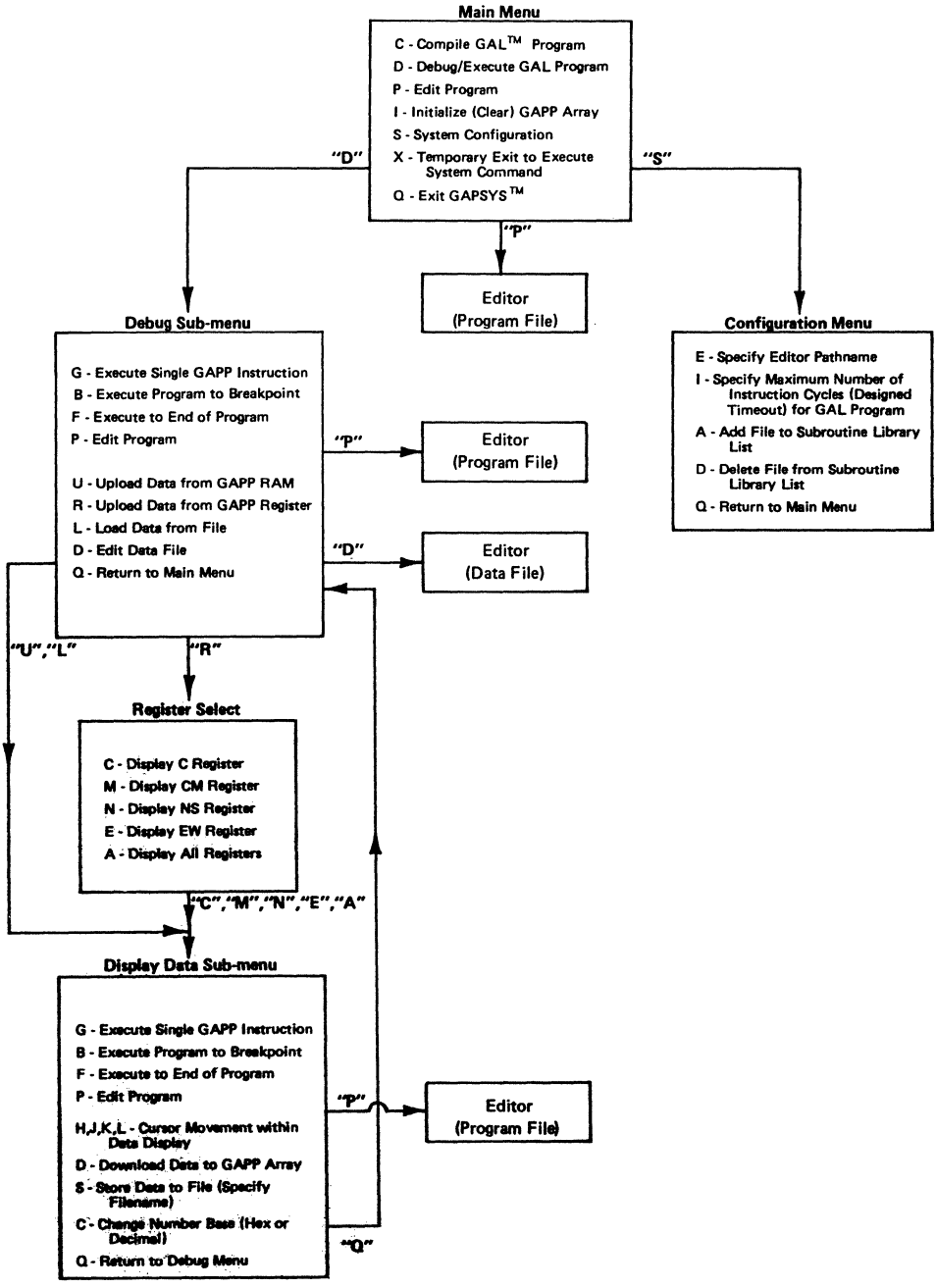
- In addition to using *int* variables as arguments to subroutine calls, the names of *images* may also be used. Both the starting address and the size of the *image* are put on the argument stack for the sub-routine to use.

- The status of the Global Output pin can be used as a criteria for conditional execution of GAL program statements. This is accomplished by the program statements:

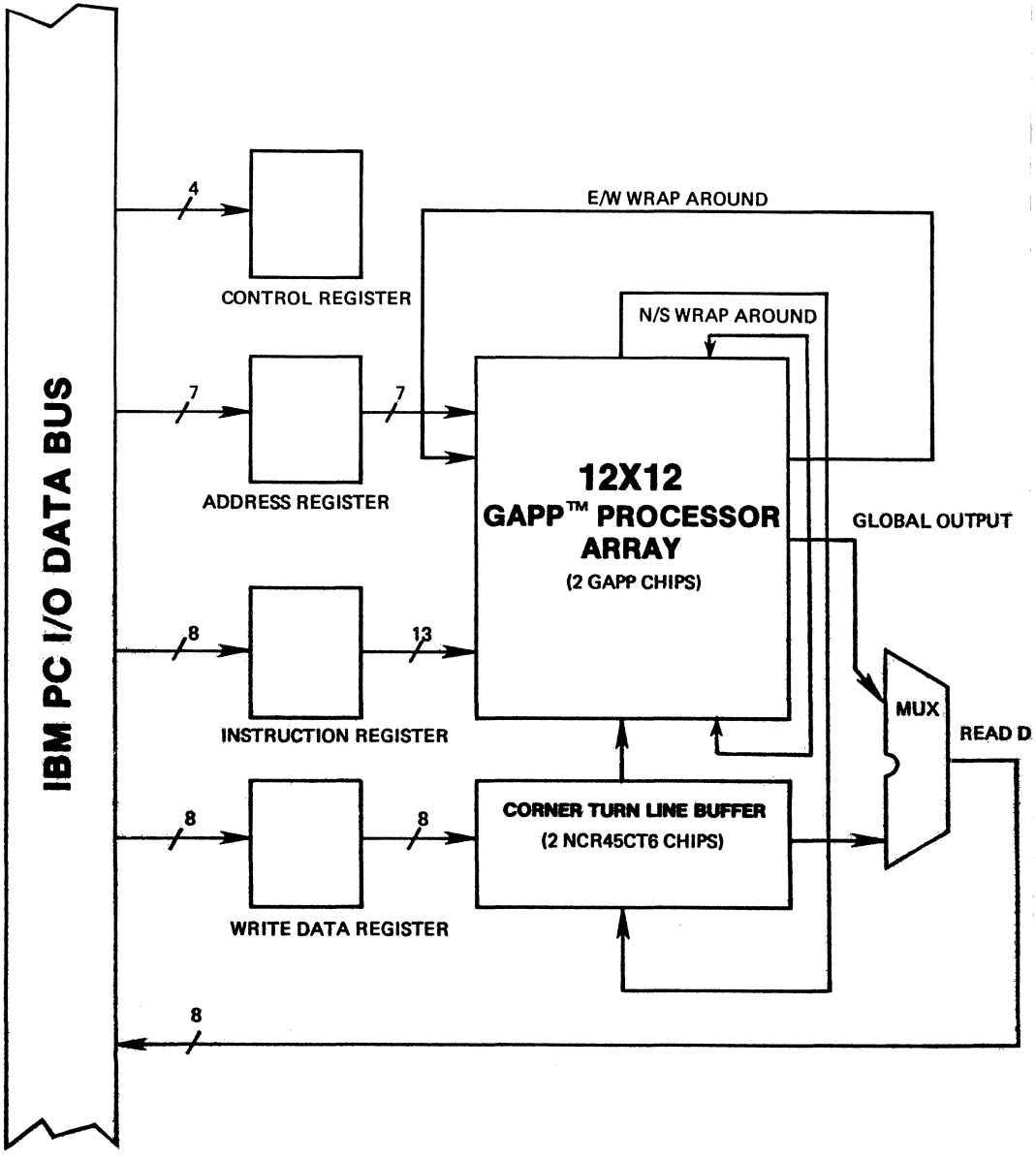
  if(goset)
  if(gocir)
  if(goset)...else
  if(gocir)...else
  while(goset)
  while(gocir)
  for (...;giset;...)
  for (...;gicir;...)

  The term *gocir* is non-zero (true) if the Global Output is low (one or more NS register contains a 1). The term *goset* is non-zero (true) if the Global Output is high (all NS registers contain 0).

GAPSYS Interactive Menu Structure

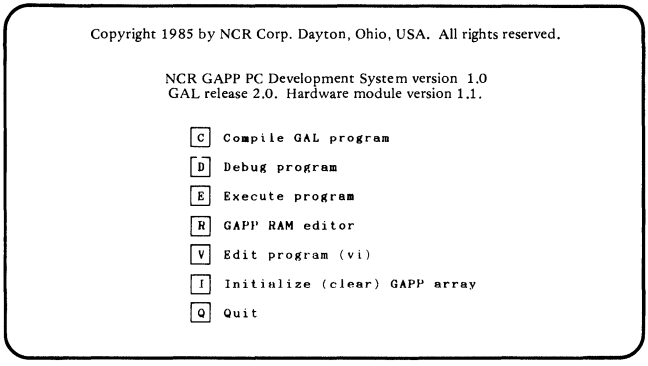


Block Diagram. GAPP PC Development System

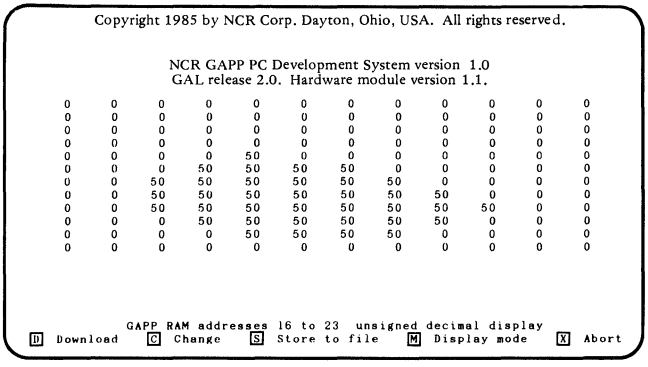


Using the GAPP PC Development System

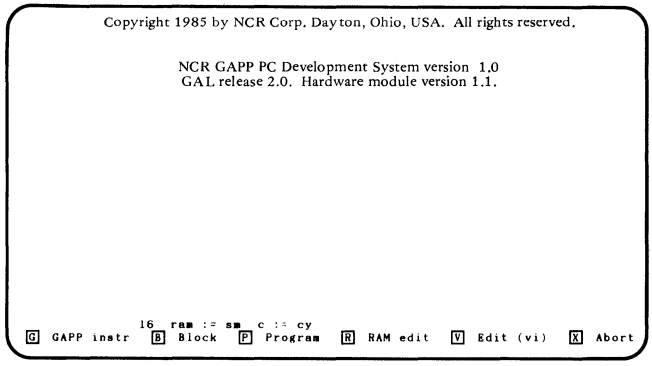
a) Displaying the main menu...



b) Reading from GAPP RAM...



c) Debugging a GAPP program...



d) Flagging a program error...

```
"edge.gal" line 19: syntax error.
"edge.gal" line 23: syntax error.

Hit RETURN to continue
```

e) Entering editor to correct program...

```
main()

{
    int     i;

    image   X:8,
            Y:8,
            Z:8;

    c := 0;
    for (i = 0; i < 8; i++) {
        X:i  ns := ram;              /*  Get bit from operand X  */
        Y:i  ew := ram;              /*  Get bit from operand Y  */
        Z:i  ram := sm  c := cy;     /*  Add bits and store in result    */
    }

    Z:i  ew := ram  ns := c  c := 1;

    for (i = 0; i < 8; i++) {
        Z:i + 1  ew := ram  c := cy;
        Z:i      ram := c  c:= 1;
    }
}

"edge.gal" 24 lines, 378 characters
```

45GS4 GAPP SYSTEM SIMULATOR PACKAGE

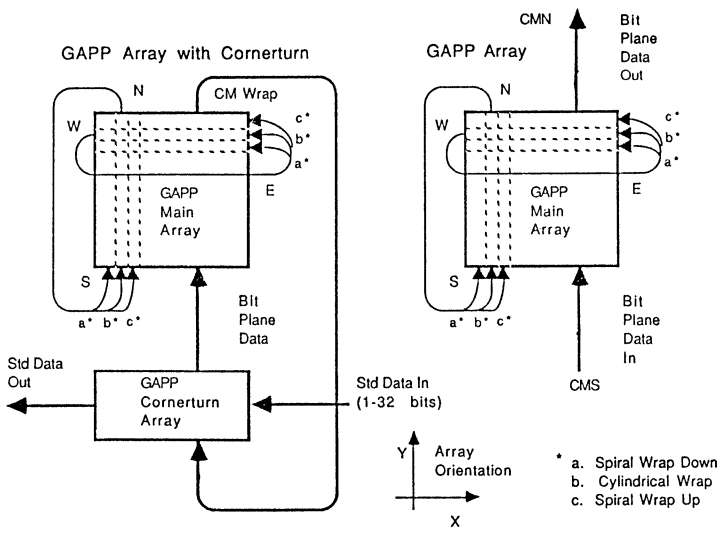
Quick Fact Sheet

The Geometric Arithmetic Parallel Processor (GAPP™) System Simulator Package (GSSP) is a software package that runs on a PC-AT™ compatible and on a VMS™ based VAX™ computer. The GAPP System Simulator Package provides a complete low cost development environment for writing, debugging and analyzing GAPP system algorithms and for analyzing GAPP related system parameters. It is comprised of four modules, the NCR GAL™ compiler, the GAPP microcode generator, the GAPP hardware simulator, and a library of coded routines. The GSSP can run solely on a main array of GAPP chips or can include a cornerturn input/output (I/O) array which properly reformats the I/O data for the GAPP Main Array.

Features:

- Develop GAPP programs in C-like language, NCR GAL
- Write and Debug GAPP Main Array Algorithms
- Write and Debug GAPP Cornerturn Algorithms
- Execute Algorithms on Array only or on Array with Cornerturn
- Inspect data in GAPP registers and GAPP RAM directly (Main array and Cornerturn)
- Variable I/O data path width (this can vary from 1 to 32 bits)
- Variable Array size (1-108 for N-S direction, 1-96 for E-W direction)
- Choice of 3 different edge wraps in N-S and E-W directions (Spiral Up, Cylindrical and Spiral Down)
- Load data files into GAPP Array
- Store GAPP Array data to files
- NCR GAL source files from the PC Development system, NCR45GDS1 are upward compatible to this system

GAPP Array Block Diagrams



GAPP is a trademark of NCR Corporation, GAL is a trademake of NCR Corporation, PC-AT is a trademark of IBM Corporation, VAX and VMS are trademarks of Digital Equipment Corporation

Command Descriptions

Simulator Configuration Commands

ctysize - specifies the size of the cornerturn in y direction
xsize - specifies the horizontal size of the GAPP Main and Cornerturn Arrays
xwrap - specifies the horizontal (E-W) wrap
ysize - specifies the vertical size of the GAPP Main Array
ywrap - specifies the vertical (N-S) wrap

Simulator Directive Commands

brkpoint - halts execution of a "do" file at a break point
cont - resumes execution of a "do" file or a microcode file after breakpoint
do - executes a file comprised of simulator commands and GAPP instructions
execsys - initiates execution of a microcode file
gostate - displays current Global Output signal state
greg - displays contents of Global Output Register
startaddr - specifies starting address of program

Data I/O Commands

load - loads data to GAPP Main Array RAM
rdarin - read array data from file (array only mode)
rdctin - read cornerturn data from file
store - store frame from GAPP Main Array RAM to file
wrarout - write GAPP Main Array data to file
wrctout - write cornerturn data to file

Simulator System Commands

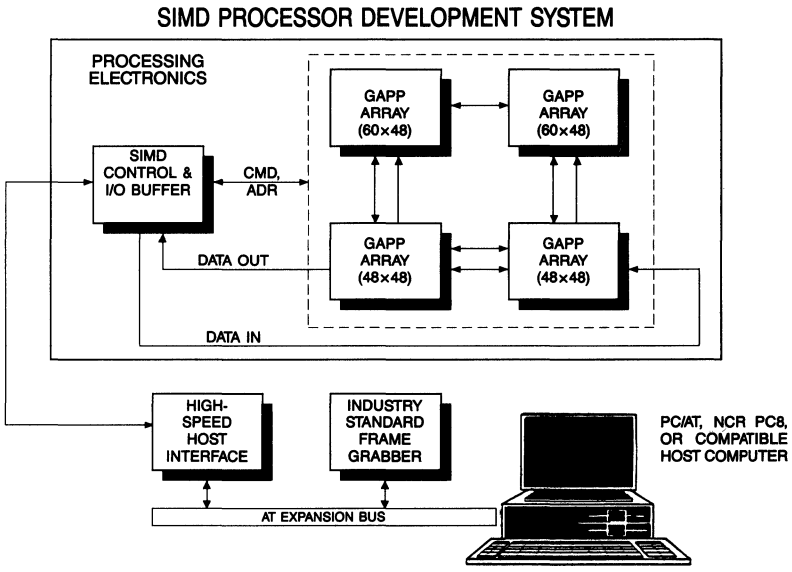
bye - ends simulation and returns to operating system
clear - initializes GAPP Main Array to zero
ctclear - initializes GAPP cornerturn to zero
ctpgrid - displays state of the cornerturn processors
ctrgrid - displays contents of cornerturn RAM
prgrid - displays state of the GAPP Main Array processors
rgrid - displays frame of GAPP Main Array RAM data

45SPDS - SIMD PROCESSOR DEVELOPMENT SYSTEM

System Features

- Provides a User-Friendly Development Environment for Geometric Arithmetic Parallel Processor (GAPP™) Based Image Processing Systems.
- Window-Oriented Menu-Driven User Interface
- NCR-GAL™ Compiler/Linker Allows GAPP Devices to be Programmed in a "C"-like Language
- Compatible with NCR-GAL Source Code from NCR's GAPP Development System (NCR45GDS1-MS)
- Source Resident Breakpoints
- GAPP RAM/Register Uploading-Downloading-Editing
- System Works in Conjunction with Industry Standard Frame Grabber to Acquire and Display RS-170 Video Images
- Direct Memory Access Connection between Host and SIMD Development System Provides High-Speed Transfer of Image and Program Data
- 10 Megahertz SIMD Controller with 64K Instruction Control Store and 256K byte Input/Output Buffer Memory
- Accommodates 1, 2, or 4 GAPP Processor Cards; Provides Processing Element Array Sizes of 48×48 to 108×96
- SIMD Controller Contains 256 bit Dynamic Constant Memory to Facilitate Data-Dependent, Adaptive Processing

System Block Diagram



Introduction

The SIMD Processor Development System (SPDS) facilitates the development of operational and algorithmic software for GAPP based systems. The SPDS is comprised of a software package which runs on a PC-AT™ compatible (host) computer such as the NCR PC8.

The software package contains the NCR GAPP Algorithm Language (NCR-GAL) Compiler and the GAPP Microcode Generator. These software tools are presented to the user of the four different operational modes customized for developing NCR-GAL software. These operational modes are accessed through a simple window oriented user interface. Also included is an on-line 'Help' feature which provides the user with menu-specific descriptions.

The SPDS hardware consists of one to four GAPP Array cards, an SIMD Controller card, a High-Speed Host Interface card and an optional frame grabber*. The High-Speed Host Interface card and the frame grabber reside in the host computer while the SIMD Controller card and the GAPP Array card(s) reside in a separate enclosure, the Processing Electronics station. The host computer and the Processing Electronics station are linked by a cable capable of transferring data at DMA rates.

SPDS System Overview

The basic components of a GAPP system include the following:

- Main Array of GAPP Processing Elements
- Corner Turn Array of GAPP Processing Elements
- Controller to provide instructions to the GAPP Array and to coordinate timing with other associated hardware.
- Host Computer to act as an interface between the user and the GAPP based system.

The SIMD Processor Development System (SPDS) provides a straight-forward, logical process for developing GAPP algorithms. Four development modes are structured to allow the user to focus on each individual step of the GAPP code generation without allowing concerns from other areas to interface.

The primary function of the SPDS system is to simplify two tasks:

1. Development of the Main Array Algorithm
2. Development of the Corner Turn Algorithm

To expedite task 2, a generic Corner Turn Algorithm is provided. (Thus task 2 may be skipped if desired.)

Four development modes for the various steps of designing NCR-GAL algorithms are presented in the order they are most likely to be used. The four modes are:

1. Algorithm Development
2. Image - No Corner Turn
3. Corner Turn Development
4. Image - With Corner Turn

The first two modes focus on modifying and testing the Main Array algorithm. The last two focus on developing a Corner Turn algorithm. Corner turning is a technique which properly reformats the data for the GAPP Array. If the algorithms are to be run only on the SPDS system, the development cycle may be concluded early by using the generic corner turn program furnished with the system.

Development Mode Descriptions

Features Common to all Modes

Source Resident Breakpoints - Breakpoints can be set at any line in the source code. When a breakpoint is encountered the sequencer halts and returns control to the user interface. Additionally, all breakpoints may be enabled or disabled without editing the source file via a menu command.

Global Output Register Inspection - The Global Output Register contains the last 16 samples of the GAPP Array Global Output Signal. This register may be inspected at any time. The current state of the Global Output Signal may also be inspected at any time.

*The optional industry standard frame buffer supported by the SIMD Processor Development Systems (SPDS) is the PG100-640/2 AT Frame Grabber from Imaging Technology. FG100 is a trademark of Imaging Technology, Inc.

Data Transfer - Data transfers are supported between the following items:

- GAPP Array contents
  - Main Array, Corner Turn RAM.
  - Main Array, Corner Turn registers.
  - Constant RAM.
- Frame Buffer
- Disk Files
- Windows

Image Display Control - The following image control functions are supported by the SPDS Frame Grabber interface.

- Canned and Camera Images
- Zooming
- Selection of a Region of Interest
- Look-Up Table Configuration

Window Control - The window-oriented user interface supports the following features.

- Overlapping
- Resizing and Moving
- Outputting Contents to Printer or ASCII Disk File

Single Keystroke or Cursor Select Pull Down Menus - Pull Down Menu items may be selected by pressing the first letter of the item desired or positioning the cursor over the item and pressing "Enter."

Operating System Access from within SPDS Software - The user interface allows DOS commands to be executed without exiting the SPDS user interface.

Mode Specific Descriptions

Algorithm Development

This mode allows the user to compile a Main Array NCR-GAL program, and execute it over a single array (one sub-image) of GAPP data. This is useful when verifying algorithm operation and when testing algorithms on several data sets.

Image - No Corner Turn

One or more compiled Main Array NCR-GAL programs are executed to process an entire image (Sub-imaging routines to split/join images are provided). This mode allows the user to focus attention on evaluating the final results of the Main Array algorithm without concern for image data I/O. This mode also allows the user to work with a "suite" of algorithms: the user has the option to run an entire set of algorithms (a suite) in sequence or to run any algorithm individually.

Corner Turn Development

In this mode the user runs all three key elements of an NCR-GAL program together. The three key elements are the Main Array algorithm, the Corner Turn algorithm, and the Takeover algorithm. Compiled versions of Main Array, Corner Turn, and Takeover NCR-GAL programs may be merged and the resulting program executed over a single GAPP array of data.

Image - With Corner Turn

A compiled Main Array NCR-GAL program may be merged with a standard "Corner-Turn" program provided with the SPDS to significantly increase execution speed. The resulting program may be executed to process an entire image. (Sub-imaging routines provided).

Hardware Component Descriptions

Processing Electronics Station

Within the Processing Electronics station is a card cage, a power supply and cooling fans. The SIMD Controller card and 1, 2 or 4 GAPP Array cards reside in the card cage.

GAPP Array

The GAPP Array consists of a two dimensional array of processing elements (PEs). This array of PEs is the computational heart of the system.

The SPDS GAPP Array consists of one or more GAPP Array cards. Each GAPP Array Card contains 40 GAPP chips (60 × 48 PEs) and is clocked at 10 Mhz.

The SPDS supports configurations of 1, 2 or 4 GAPP Array cards. (Additional array cards increase the overall throughput of the system by increasing the number of PEs.) Eight or twelve-bit I/O data is supported by all configurations. The signals along the periphery of each GAPP Array Card are brought out to connectors to support interconnection of GAPP array cards.

A four card configuration provides a 108 × 96 processing element GAPP array and a 12 × 96 element Corner Turn array. In the one and two GAPP array card configurations, Main Array sizes of 48 × 48 and 48 × 96 are provided respectively. The respective Corner Turn arrays are 12 × 48 and 12 × 96.

SIMD Controller

The SIMD Controller is contained on a single circuit card which resides in the Processing Electronics station. It communicates with the host computer via the High-Speed Host Interface board. The controller card performs the following functions:

Main GAPP Array Control Functions - The SIMD Controller card broadcasts commands and RAM address signals to the Main GAPP Array. A run-time adaptive Constant RAM and the SIMD Controller allows the system to perform operations such as adaptive thresholding and adaptive filtering.

Corner Turn GAPP Array Control Functions - The SIMD Controller card broadcasts commands and address signals to the Corner Turn array for data reformatting.

Data Transfer Functions - The SIMD Controller card contains a pair of high-speed buffers that support the host's DMA data transfer mode. Data may be transferred to/from the host using one buffer while data from the other buffer is simultaneously transferred to/from the GAPP Array.

Program Download Functions - The writable control store memory on the SIMD Controller card allows the host to download compiled NCR-GAL programs to the SPDS.

System Control Functions - The host monitors and controls the operation of the Processing Electronics station by accessing the status and control registers of the SIMD Controller card.

High-Speed Host Interface

The High-Speed Host Interface card is located in a 16-bit slot on the AT Expansion Bus. The High-Speed Host Interface provides a 16-bit direct memory access path or an 8/16-bit programmed input/output path between the host computer and the Processing Electronics Station. The High-Speed Host Interface works in conjunction with the host computer's direct memory access controller to achieve data transfers of approximately one million 16-bit words per second.

Frame Grabber

An Imaging Technology Series 100™ Frame Grabber card is located in a slot on the AT Expansion Bus. The Frame Grabber accommodates RS-170 video devices, thereby allowing the SPDS to capture and display images. Captured image data is available for processing by the SPDS. Processed images may be transferred back to the Frame Grabber for display. The Frame Grabber supports block DMA transfers to accelerate sub-image transfers. The Frame Grabber also supports pseudocolor output to enhance the visual appearance of processed images.

GAL LIBRARY FUNCTIONS

The GAL programs included in this listing are "library" functions which may be used in conjunction with the GAPSYS software package of the GAPP PC Development System (NCR45GDS1-MS). The intention is to aid the user in developing GAL (GAPP Algorithm Language) programs for the GAPP processor (NCR45CG72). These programs serve two purposes: they provide the user with some examples of how GAL subroutine programs may be written and they provide a collection of library functions which may be used for applications programming. Note that these programs have been written as "general purpose" routines. In many cases, it is possible to optimize (modify) the code for a specific application to obtain superior performance over these standard functions.

The following functions have been implemented as GAL sub-routines:

absval	Find absolute value of 2's complement number
acctc	Accumulate 2's complement numbers
addsm	Add signed magnitude numbers
addtc	Add 2's complement numbers
addum	Add unsigned magnitude numbers
compare	Compare unsigned or 2's complement numbers
dilate8	Dilate binary image in eight directions
dividesm	Divide signed magnitude numbers
dividetc	Divide 2's complement numbers
divideum	Divide unsigned magnitude numbers
edge	Extract edge of gray scale image
erode8	Erode binary image in eight directions
fswitch	Conditional reversal of two inputs (two output multiplexor)
median3	3 by 3 median filter
median31	3 by 3 median filter, line scan version
multsm	Multiply signed magnitude numbers
multtc	Multiply 2's complement numbers
multum	Multiply unsigned magnitude numbers
mux	Conditional selection of two inputs (multiplexor) using simple algorithm
mux2	Optimized multiplexor using "tricky" algorithm
sort	Add data to a sorted list
shift	Shift in any of 8 directions
skel	Extract topological information from an image of an object and produce a skeleton that represents the original topology
sm2tc	Signed magnitude to 2's complement converter
subsm	Subtract signed magnitude numbers
subtc	Subtract 2's complement numbers
tc2sm	2's complement to signed magnitude converter
thresh	Flag all pixels above a given threshold
xsobel	Find gradient in x direction using sobel xform
ysobel	Find gradient in y direction using sobel xform

A more detailed description of a function may be found in the header section of that functions's source listing. The header section will give a description of the function, the parameters, additional GAPP memory required within the sub-routine, execution times (in GAPP cycles), and any special notes about characteristics or usage of the function.

Each of these functions is found in a file listing which has the same name and a file extension of ".gal". For example, the "acctc" function is in a file named "acctc.gal."

Special Note:

All of the signed magnitude (sm) library functions allow (and produce) both positive and negative zero. If this is not desirable for a specific application, the results of these functions may be checked for the negative zero condition at the end of the sm function. If the result is negative zero, the sign bit should be inverted.

Only 5 mA while cycling through multiply/accumulate operations at a 5 MHz pace.

45CM16 CMOS 16 X 16 BIT SINGLE PORT MULTIPLIER/ ACCUMULATOR

Introduction

The NCR45CM16 multiplier/accumulator device is a small, low-power device which is compatible with 16-bit microprocessor systems. Input/output data is transferred through a 16-bit bi-directional data bus in signed two's complement form. The M16 is asynchronous, therefore does not require a clock. It can be attached to a system bus like a 16-bit wide static RAM with little additional circuitry and little space. The system sees it as a 200 nS, 16-bit wide static RAM.

It provides an inexpensive solution to multiplication operations as opposed to alternatives like the addition of an array processor board or redesigning the system.

The M16 is well suited to accelerating the multiply/accumulate operations associated with 16-bit controller or digital signal processing applications, releaving the processor of those operations and speeding up the system significantly.

The M16 is packaged in a 24-pin DIP as opposed to a conventional three-port multiplier/ accumulator device which comes in a 64-pin DIP package. The M16 can easily be placed in existing systems because of its small size and small power requirements.

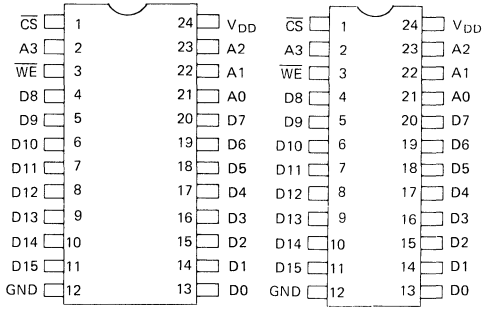
General Description

The NCR45CM16 is a 24 pin CMOS multiplier/accumulator for use with 16-bit microprocessor systems. All input and output data are transferred through a single 16-bit bidirectional data bus in signed two's complement format. This device is TTL/CMOS compatible and requires no clock due to its total static (asynchronous) operation. The device may be attached to the system bus in the same way as a 16-bit wide static RAM. A single 16x16 multiply and read 32-bit result requires 5 cycles (write X, write Y, multiply, read high-order result, read low-order result). Pipelined multiply/accumulate operations require only 2 cycles each.

Features

- 24 Pin Package
  —300 mil Ceramic "Skinny DIP"
  —600 mil Plastic DIP
- 40 bit Accumulator
  —Add Product to Accumulator
  —Subtract Product from Accumulator
- Cycle Time 190 ns (typ)
- Low Power CMOS
  —100μW Standby (max)
  —10mA Operating (max)
- Single 5 Volt $\pm$ 10% Supply
- Fully Static Operation—No Clock Required
- 3-state Bus Compatible Outputs

Pin Configuration



NCR45CM16-P
Plastic DIP

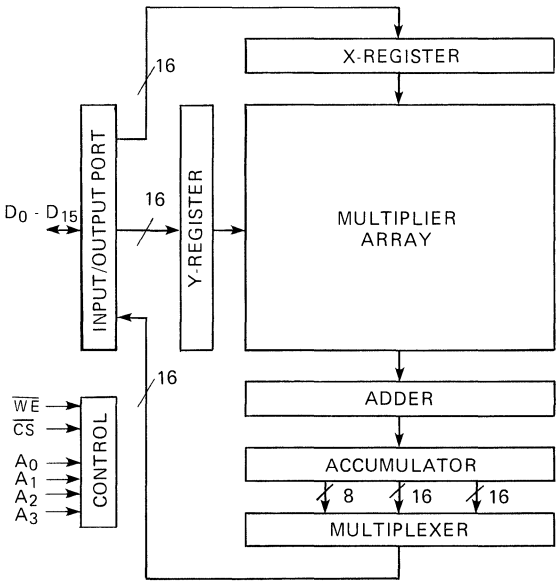
NCR45CM16-C
Ceramic DIP

Pin Names

A0 - A3	Address Inputs
D0 - D15	Data Inputs/Outputs
$\overline{CS}$	Chip Select
$\overline{WE}$	Write Enable
V_{DD}	5V $\pm$ 10% Supply Voltage

COMMUNICATIONS PRODUCTS

Functional Block Diagram



Absolute Maximum Ratings

Supply Voltage, V_{DD} ..... +7V
Voltage on any pin with respect to ground ..... −0.3 to V_{DD} +0.3V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operation conditions stipulated elsewhere in this specification is not implied.

Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{DD}	Supply voltage	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{DD}	Volts
V_{IL}	Input low level voltage	0		0.8	Volts
T_A	Operating ambient temperature	0		70	°C

Static Electrical Characteristics Over Recommended Operating Conditions

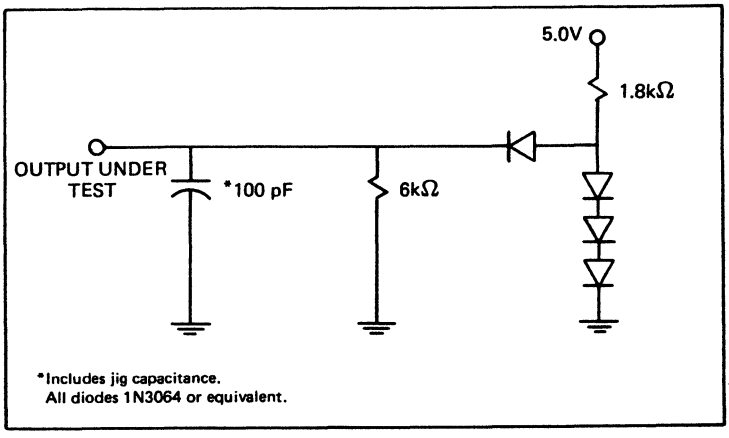
Symbol	Parameter	Condition	Min.	Typ.*	Max.	Units
I_{IN}	Input leakage current	V_{IN}=0V to V_{DD} max			± 10	μA
I_O	Output leakage current	V_O=0.4 to V_{DD} max, $\overline{CS}$=1			± 10	μA
V_{OH}	Output high voltage current	I_{OH}=400μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL}=2.1mA			0.4	Volts
I_{DD}	Supply current – Active	Outputs Open		5	15	mA
	Supply current – Standby	$\overline{CS}$ = 1, V_{IN} = 0V		0.01	0.1	mA

*Typical limits are V_{DD} = 5.0V, T_A = 25°C; typical parameters are not guaranteed

Capacitance

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
C_{IN}	Input capacitance	All pins except pin			5	pF
$C_{I/O}$	Input/Output capacitance	under test are tied to ground			10	pF

AC Test Load Circuit



CAUTION

1. **CMOS Devices are damaged by high energy electrostatic discharge. Devices must be stored in conductive foam or with all pins shunted.**

2. **Remove power before insertion or removal of this device.**

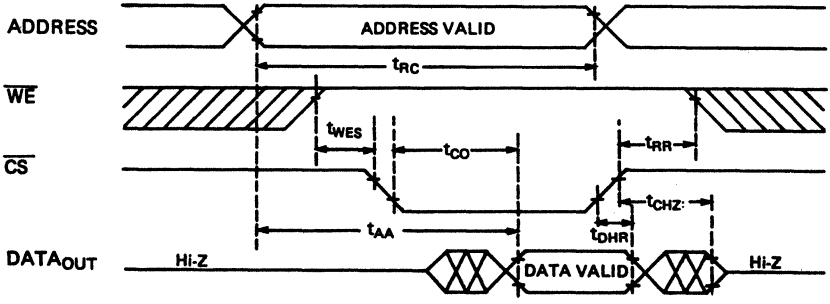
AC Characteristics

Read Cycle

SYMBOL	PARAMETER	NCR45CM16		
		Min.	Max.	Units
t_{RC}	Read Cycle Time	220		ns
t_{AA}	Address Access Time		120	ns
t_{CO}	Chip Select to Output Data Valid		70	ns
t_{WES}	Write Enable Set Up Before Select	0		ns
t_{RR}	Read Recovery Time	15		ns
t_{CHZ}	Chip Deselect to Output High-Z		60	ns
t_{DHR}	Data Hold from Read Time	0		ns

Read Cycle Timing Waveforms

The read operation is performed with $\overline{WE}$ = high. The falling edge of $\overline{CS}$ latches the address and initiates the read process.

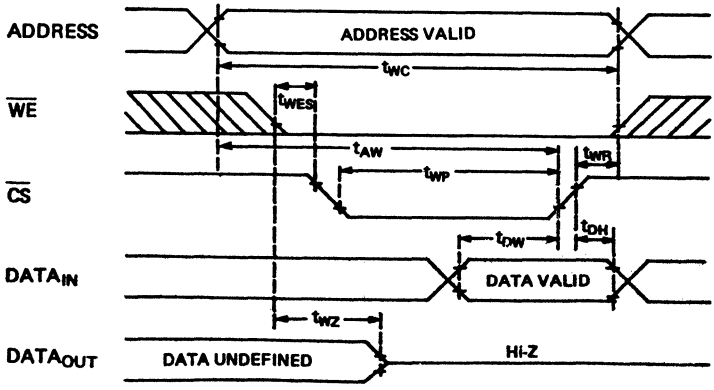


Write Cycle

SYMBOL	PARAMETER	NCR45CM16		
		Min.	Max.	Units
t_{WC}	Write Cycle Time	220		ns
t_{AW}	Address Valid to End of Write	170		ns
t_{WES}	Write Enable Set Up Before Select	0		ns
t_{WR}	Write Recovery Time	15		ns
t_{WP}	Write Pulse Width	120		ns
t_{DW}	Data Set Up to Write Time	70		ns
t_{DH}	Data Hold From Write Time	30		ns
t_{WZ}	Write Enable to Output Hi-Z	0	50	ns

Write Cycle Timing Waveforms

The write operation is performed with $\overline{WE}$ = low. The falling edge of $\overline{CS}$ latches the address and the rising edge of $\overline{CS}$ latches the data in.



Read Operations ($\overline{WE}$ = 1)

A_3	A_2	A_1	A_0	OPERATION
X	0	0	0	Read bits 0 through 15 of result from accumulator
X	0	0	1	Read bits 16 through 31 of result from accumulator
X	0	1	X	Read bits 32 through 47 of result* from accumulator

Divide by 2 and Read ($\overline{WE}$ = 1)

A_3	A_2	A_1	A_0	OPERATION
X	1	0	0	Read bits 1 through 16 of result from accumulator
X	1	0	1	Read bits 17 through 32 of result from accumulator
X	1	1	X	Read bits 33 through 48 of result from accumulator*

X = Don't care

*NOTE: Accumulator accumulates to 40 bits. Thus bits 0 - 39 are valid, while bits 40 - 48 are a sign extension of bit 39.

Write Operations ($\overline{WE}$ = 0)

A_3	A_2	A_1	A_0	Operation	A_3	A_2	A_1	A_0	Operation
0	0	0	0	Clear Accumulator Retain X and Y	1	0	0	0	Retain Accumulator Retain X and Y (NOP)
0	0	0	1	Clear Accumulator Write new data to Y	1	0	0	1	Retain Accumulator Write new data to Y
0	0	1	0	Clear Accumulator Write new data to X	1	0	1	0	Retain Accumulator Write new data to X
0	0	1	1	Clear Accumulator Write new data to X and Y	1	0	1	1	Retain Accumulator Write new data to X and Y
0	1	0	0	Add X Y to Accumulator Retain X and Y	1	1	0	0	Subtract X Y from Accum. Retain X and Y
0	1	0	1	Invalid Operation	1	1	0	1	Invalid Operation
0	1	1	0	Add X Y to Accumulator Write new data to X	1	1	1	0	Subtract X Y from Accum. Write new data to X
0	1	1	1	Add X Y to Accumulator Write new data to X and Y	1	1	1	1	Subtract X Y from Accum. Write new data to X and Y

Example Operations

1. Multiply two 16-bit numbers, read 32-bit result

Instruction	$\overline{WE}$	Operation
0010	0	Clear A, Write X
0001	0	Clear A, Write Y
0100	0	Add X · Y to A
0000	1	Read low order result
0001	1	Read high order result

2. Multiply two 16-bit numbers and accumulate, repeat five times (five point digital filter), read 40-bit result = $X_1Y_1 + X_2Y_2 + X_3Y_3 + X_4Y_4 + X_5Y_5$

Instruction	$\overline{WE}$	Operation
0010	0	Clear A, Write X1
0001	0	Clear A, Write Y1
0110	0	$A = X_1 \cdot Y_1$, Write X_2
1001	0	Write Y_2
0110	0	$A = X_1 \cdot Y_1 + X_2 Y_2$, Write X_3
1001	0	Write Y_3
0110	0	$A = X_1 \cdot Y_1 + X_2 Y_2 + X_3 Y_3$, Write X_4
1001	0	Write Y_4
0110	0	$A = X_1 \cdot Y_1 + X_2 Y_2 + X_3 Y_3 + X_4 Y_4$, Write X_5
1001	0	Write Y_5
0100	0	$A = X_1 \cdot Y_1 + X_2 Y_2 + X_3 Y_3 + X_4 Y_4 + X_5 Y_5$
0010	1	Read most significant bits (32-47) of result
0001	1	Read bits 16-31 of result
0000	1	Read bits 0-15 of result

3. Half of sum of squares = $1/2 (B_1^2 + B_2^2)$

Instruction	$\overline{WE}$	Operation
0011	0	Clear A, Write B_1 to Registers X and Y
0111	0	$A = B_1^2$, Write B_2 to Registers X and Y
0100	0	$A = B_1^2 + B_2^2$
0101	1	Divide A by 2 and read most significant 16 bits

4. Scale a series of numbers by a constant

Instruction	$\overline{WE}$	Operation
0001	0	Clear A, Write Constant to Y
1010	0	NOP A, Write X_1
0100	0	$A = X_1 Y$
0001	1	Read high order result
0000	1	Read low order result
0010	0	Clear A, Write X_2
0100	0	$A = X_2 \cdot Y$
0001	1	Read high order result
0000	1	Read low order result
0010	0	Clear A, Write X_3
0100	0	$A = X_3 \cdot Y$
0001	1	Read high order result
0000	1	Read low order result

45CM16 APPLICATION NOTE NUMBER 1

As many assembly language programmers can attest, performing multiplication operations with a microprocessor can take a great amount of time. The unaided microprocessor is especially slowed down by repeated multiply-accumulate operations that are common in process control or digital signal processing applications. This reduced performance limits the maximum bandwidth signal that the general purpose microprocessor can handle.

The alternatives, however, for improving the effective throughput of the processor are expensive. Previously the system designer could add a special purpose array processor board to his system, or redesign his system to use a special purpose DSP microprocessor. Both of these options require high expense or extensive engineering which may not be justified for many applications.

Another solution, that of adding an expensive three port multiplier chip with the associated latches and logic required to interface it to the system, can take up a large amount of system board space and consume an inordinate amount of power.

On the other hand, a small, low power multiplier chip that could be interfaced to the system with little additional circuitry would be an attractive solution to the throughput problem. NCR has developed a microprocessor bus compatible, 16 x 16 multiplier (NCR45CM16) which is designed specifically as a microprocessor "multiplication accelerator." It is packaged in a small 24-pin DIP and typically consumes only 5mA while cycling through multiply-accumulate operations at a 5 MHz pace. One important feature of the device is its simple system interface. The NCR45CM16 attaches to the microprocessor bus and appears to the system as a 200-ns, 16-bit wide static RAM. Figure 1 shows the size of the NCR45CM16 package next to a conventional three-port multiplier. The small size of the single port device will allow its incorporation into many existing microcomputer boards.

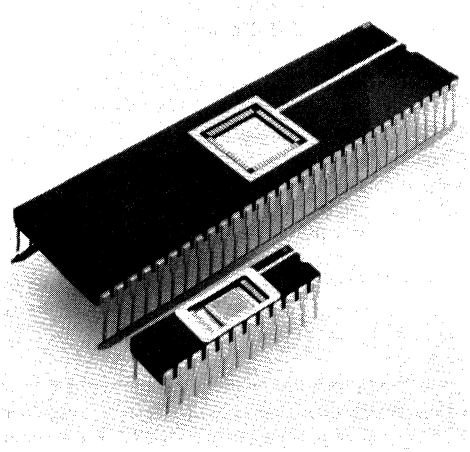


Figure 1. Comparison between the NCR45CM16 (below) and conventional three-port multiplier/accumulator chip (above) clearly shows that the bulky three-port does not size up for space limited microprocessor board designs.

Use in a System

The multiplier chip is most easily used if it is mapped directly into the processor's memory space. This is because the device has Chip enable (CE) and Write enable (WE) pins that perform the same functions as they would for a static RAM. When the device is not enabled, the I/O pins will go into a high impedance state that effectively disconnects the multiplier from the system bus. As shown in Figure 2, the chip has input registers X-REG and Y-REG that are written to through the single port bus interface. The product of these registers is then available for an accumulate operation on the next cycle. This 'product' may be added to or subtracted from the 40-bit accumulator while the X register is simultaneously updated. The result in the 40-bit accumulator may be read 16-bits at a time: least significant 16, most significant 16, or high significant 16. The latter is produced only with repeated multiply-accumulates that create a result greater than 32-bits. Figure 3 provides details of the multiplier operation. Control of the input registers, output registers and accumulator operation is determined by bits of the address bus (A_0-A_3).

For a series of multiply-accumulate operations (such as an FIR filter computation), the device can operate as a two cycle pipeline (Write to X-REG and accumulate, Write to Y-REG). After the last arithmetic operation, three read operations would be required to obtain the full precision output. A multiplier-aided-68000 or 8086 will be approximately three times faster than an unaided 68000 or 8086 microprocessor using only the internal multiply instruction.

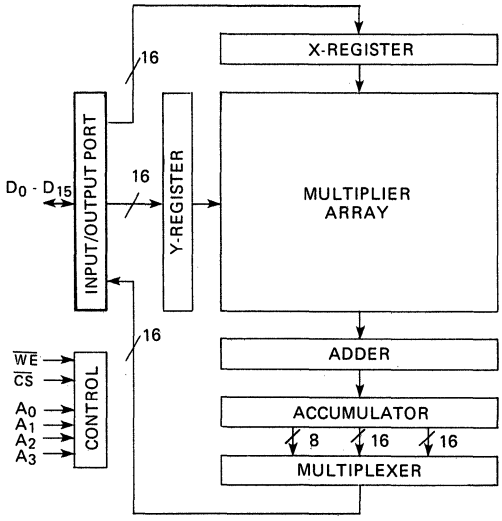


Figure 2. Functional Block Diagram

Microprocessor Interface

The NCR45CM16 is easily interfaced to both the 68000 and 8086 microprocessors. Typical interface circuitry for both micros can be seen in Figures 4a-4c. Examples of 68000 and 8086 assembly code used with the multiplier are included at the end of this application note.

Read Operations ($\overline{WE}$ = 1)

A_3	A_2	A_1	A_0	Operation
X	0	0	0	Read bits 0 through 15 of result from accumulator
X	0	0	1	Read bits 16 through 31 of result from accumulator
X	0	1	X	Read bits 32 through 47 of result* from accumulator

*NOTE: Accumulator accumulates to 40 bits. Thus bits 0-39 are valid while bits 40-47 are a sign extension of bit 39.

Divide By 2 and Read ($\overline{WE}$ = 1)

A_3	A_2	A_1	A_0	Operation
X	1	0	0	Read bits 1 through 16 of result from accumulator
X	1	0	1	Read bits 17 through 32 of result from accumulator
X	1	1	X	Read bits 33 through 48 of result* from accumulator

X = Don't care

NOTE: Accumulator accumulates to 40 bits. Thus bits 1-39 are valid, while bits 40-48 are a sign extension of bit 39.

Figure 3a. Read operations of the NCR45CM16 are determined by 4 address pins (A_0-A_3) and the write enable ($\overline{WE}$) pin.

Write Operations ($\overline{WE}$ = 0)

A_3	A_2	A_1	A_0	Operation	A_3	A_2	A_1	A_0	Operation
0	0	0	0	Clear Accumulator Retain X and Y	1	0	0	0	Retain Accumulator Retain X and Y (NOP)
0	0	0	1	Clear Accumulator Write new data to Y	1	0	0	1	Retain Accumulator Write new data to Y
0	0	1	0	Clear Accumulator Write new data to X	1	0	1	0	Retain Accumulator Write new data to X
0	0	1	1	Clear Accumulator Write new data to X and Y	1	0	1	1	Retain Accumulator Write new data to X and Y
0	1	0	0	Add X Y to Accumulator Retain X and Y	1	1	0	0	Subtract X Y from Accum. Retain X and Y
0	1	0	1	Invalid Operation	1	1	0	1	Invalid Operation
0	1	1	0	Add X Y to Accumulator Write new data to X	1	1	1	0	Subtract X Y from Accum. Write new data to X
0	1	1	1	Add X Y to Accumulator Write new data to X and Y	1	1	1	1	Subtract X Y from Accum. Write new data to X and Y

A = Accumulator
X = Data latched into X-register
Y = Data latched into Y-register

Figure 3b. Write operations of the NCR45CM16 are determined by 4 address pins (A_0-A_3) and the write enable ($\overline{WE}$) pin.

MC6800 to 45CM16 Interface

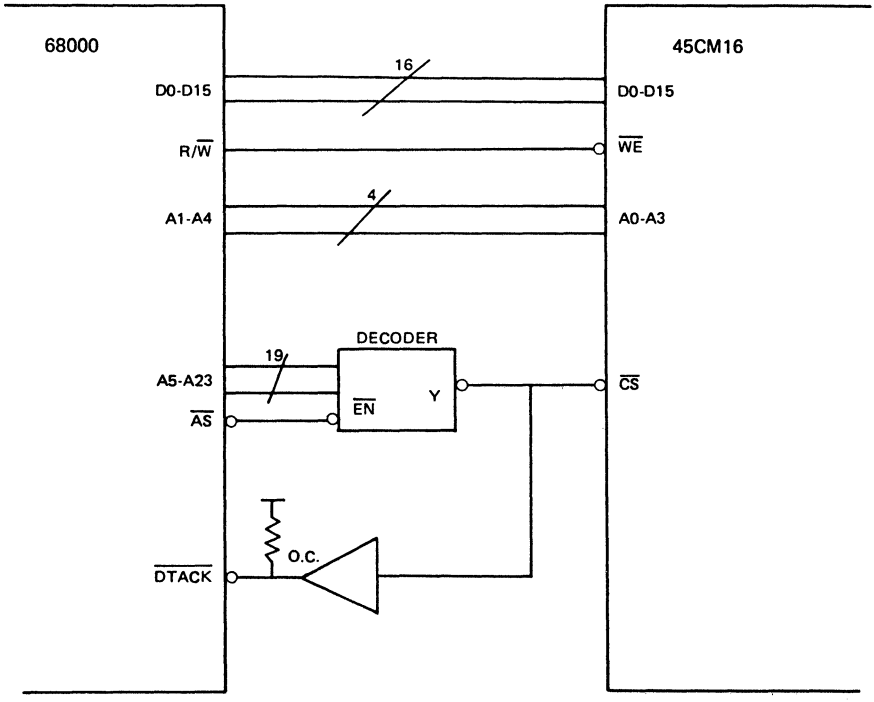


Figure 4a.

8086 to 45CM16 Interface

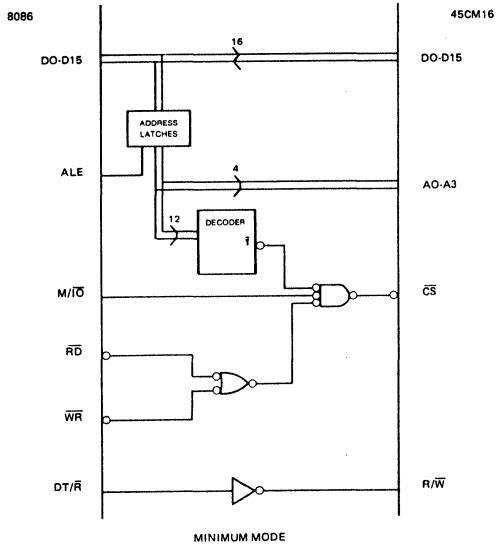


Figure 4b.

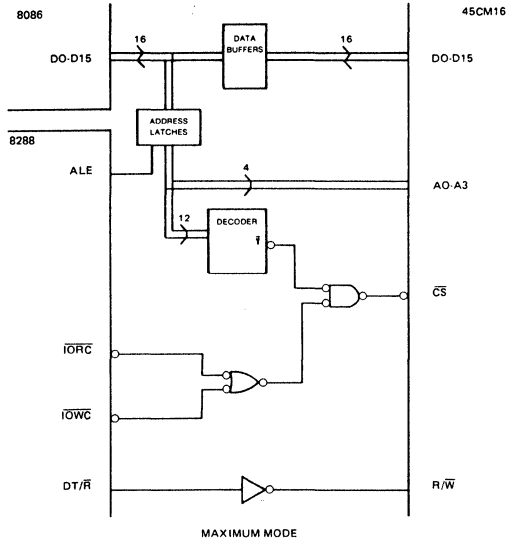


Figure 4c.

Example:

Using the NCR45CM16 for Assembly Language Multiplication/Addition - 68000 Application

The NCR45CM16 can speed up compute-bound problems on 16 bit microprocessors. One application that benefits from adding a 45CM16 is the computation of the sum of products:

A = (X1) * (Y1) + (X2) * (Y2) + . . . + (Xn) * (Yn).

Code for implementing this algorithm on the MC68000 is given below:

- SUBROUTINE: Sum Products
- A0 points to the first element in the X list
- A1 points to the first element in the Y list
- D0 contains the number of products to be summed
- Result of product sum returned in low byte of D3 plus D2
- Define memory mapping for relevant 45CM16 instructions:

```
NCRAREA     EQU    xxxx                    BASE ADDRESS FOR 45CM16 I/)
• Offsets for writes
WXYCLRA     EQU    $3                      WRITE TO BOTH X, Y; CLEAR A
ADDXYWX     EQU    $6                      ADD X * Y TO A; PUT NEW DATA IN X
WRITE_Y     EQU    #9                      WRITE NEW DATA TO Y
• Offsets for reads
A_LOW       EQU    $0                      LOW WORD OF 40 BIT ACCUM.
A_MID       EQU    $1                      BITSX 16-31 OF A
A_HIGH      EQU    $2                      BITS 32-47 (40-47 EXTENDED)
Executable code:
START       MOVE . W #NCRAREA, A2
            CLR . W WXYCLRA * 2 (A2)       SEND 0's TO X, Y, AND A
LOOP        MOVE . W (A0) +, ADDXYWX * 2 (A2)   MULTIPLY/ACCUMULATE, NEXT X
            MOVE . W (A0) +, ADDXYWX * 2 (A2)   WRITE NEXT Y
            MOVE . W (A1) +, WRITE_Y * 2 (A2)
            DBF    D0, LOOP                LST MULTIPLY/ACCUMULATE
            MOVE . W D0,ADDXYWX * 2 (A2)    FETCH LOW WORD IN ACCUMULATOR
            MOVE . W A_LOW * 2 (A2), D1     FETCH BITS 16-31 IN A
            MOVE . W A_MID * 2 (A2), D2     MOVE MIDDLE A WORD TO HIGH D2
            SWAP   D2                       CONVERT TO SINGLE 32-BIT
            MOVE . W D1, D2                 FETCH HIGH ACCUMULATOR WORD
            MOVE . W A_HIGH * 2 (A2), D3
            RTS
```

Of course, the sub-routine can be made to execute even faster by using separate address registers to hold write and read locations instead of using offsets. But even in the above, register-conserving approach it is clear that using the 45CM16 to do the multiply-and-accumulate loop greatly reduces the overhead and shortens the code of the corresponding loop for an unaided 68000. Without the 45CM16 a programmer would have to use the 68000's own signed multiply instruction and a 32 bit addition even to accumulate to just 32 bits. The requires 82 machine cycles of execution time for the unaided 68000 versus either 24 or 32, depending on the addressing mode, for the same operations done through the 45CM16 in a loop.

The largest disadvantage to the unaided approach, however, is the overhead required to do accumulation. With the 45CM16 at least 256 products can safely be added and the high byte of the 40 bit accumulator can be fetched at the end by the 68000. Without the 45CM16, adding 32 bit quantities in succession requires overflow checking and updating the bits in a second data register on each addition. This results in still further delay in the loop.

In short, there is more code and more delay in the unaided mutliply-and-accumulate loop than in a similar loop executed through the 45CM16. With the multiplication accelerator, much code becomes unnecessary and the only additional code required for communicating with the 45CM16 is that which fetches the result from the accumulator at the end.

Using the NCR45CM16 For Assembly Language Multiplication/Addition on the 8086

The NCR45CM16 MAC can be interfaced with the Intel 8086 bus in two ways, memory mapped mode or I/O mode. In the memory mapped mode, the 45CM16 acts as a 16-bit wide RAM device connected to the 8086 bus. Data transfer between the 8086 and the 45CM16 is achieved by using one of the 23 different addressing modes available with the MOV instruction. In the I/O mapped mode, the 45CM16 acts as a 16 bit wide I/O device or peripheral connected to the 8086 bus. Data transfer between the 8086 and the 45CM16 is achieved by using only two I/O instructions, IN and OUT.

The advantage of connecting the 45CM16 in the memory mapped mode is the availability of a large number of addressing modes for data transfer. However, all of these modes, except one, have higher execution times than the simple IN and OUT instructions associated multiply/accumulate operations on the 8086, the I/O interfacing mode is used in this example. Comparing the unaided 8086 to the 45CM16 used in the I/o mode indicates a speedup of approximately 3X for the multiply operation.

The following assembly language subroutine is used to calculate SUM as
SUM = X (1) * Y (1) + X (2) * Y (2) +
X (3) * Y (3) +.....+ X (N) * Y (N)

The 45CM16 and data arrays are mapped into the memory space OO hex to FF hex. An Intel Intellec IV development system was used to write the subroutine.

```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE SUM + OF + PRODUCTS
OBJECT MODULE PLACED IN : F1: NCR. OBJ
NO INVOCATION LINE CONTROLS

LOC    OBJ                    LINE        SOURCE

                                1                NAME    SUM + OF + PRODUCTS
                                2        ;
                                3        ;       THIS SUBROUTINE CALCULATES THE SUM OF PRODUCTS AS
                                4        ;       SUM=X(1)*Y(1)+X(2)*Y(2)+.....+X(N)*Y(N:
                                5        ;       IT ASSUMED THAT NO OVERFLOW OCCURES DURING THIS CONFIRMATION
                                6        ;
                                7        ;       THE CALLING PROGRAM SETS THE VALUE OF BP REGISTER TO THE BASE
                                8        ;       ADDRESS OF THE DATA REQUIRED  FOR THIS SUBROUTINE
                                9        ;
                               10        ;       N = NUMBER OF MULTIPLICATIONS IN THE SUM
                               11        ;       X AND Y = ARRAYS TO BE MULTIPLIED AND ADDED
                               12        ;       SUM + LOW, SUM + MID, SUM + HIGH = VARIABLES WHERE THE SUM IS STORED
                               13        ;
                               14        ;
                               15        ;                  \<--------- 16 BITS---------->\
                               16        ;                   ------------------------------
                               17        ;       BP-------->\              N               \ 2 BYTES
                               18        ;                   ------------------------------
                               19        ;                1 \           X ARRAY            \
                               20        ;                2 \                              \
                               21        ;                  \                              \ 2N
                               22        ;                . \                              \ BYTES
                               23        ;                . \                              \
                               24        ;                  \                              \
                               25        ;                N \                              \
                               26        ;                   ------------------------------
                               27        ;                1 \           Y ARRAY            \
                               28        ;                2 \                              \
                               29        ;                  \                              \
                               30        ;                . \                              \ 2N
                               31        ;                . \                              \ BYTES
                               32        ;                  \                              \
                               33        ;                N \                              \
                               34        ;                   ------------------------------
                               35        ;                  \       SUM + LOW              \
                               36        ;                  \       SUM + MID              \ 6 BYTES
                               37        ;                  \       SUM + HIGH             \
                               38        ;                   ------------------------------
                               39        ;
                               40                      ASSUME            CS: CODE
                               41        ;
```


COMMUNICATIONS PRODUCTS

```
LOC   OBJ         SOURCE

----                         CODE           SEGMENT
                  ;
0000                         SUM            PROC           NEAR
                  ;
 00E4                        CLA← WRX1      EQU            11100100B
 00E2                        CLA← WRY1      EQU            11100010B
 00EC                        ACCXY← WRX     EQU            11101100B
 00F2                        NOP ← WRY      EQU            11110010B
 00E8                        ADDXY ← NOP    EQU            11101000B
 00E0                        RD ← LOW       EQU            11100000B
 00E2                        RD ← MID       EQU            11100010B
 00E4                        RD ← HIGH      EQU            11100100B
                  ;
                  ;      THE 8 EQUATE STATEMENTS DEFINE THE ADDRESSES REQUIRED
                  ;      FOR VARIOUS OPERATIONS OF NCR 45CM16. THESE ARE BASED ON
                  ;      THE I/O DECODING SCHEME SHOWN IN THE ENCLOSED REPORT.
                  ;
                  ;
                  ;
0000 8BF5                         MOV      SI,BP              ; SI HOLDS THE
                                                                BASE ADDRESS
                  ;
0002 8B0C                         MOV      CX,[SI]            ; CX=N
0004 8B09                         MOV      BX,CX              ; BX=N
0006 03DB                         ADD      BX,BX              ; BX=2N
0008 46                           INC      SI
0009 46                           INC      SI                 ; NOW SI POINTS TO
                  ;                                             FIRST ELEMENT OF
                  ;                                             X ARRAY
                  ;
                  ;
000A 8B04                         MOV      AX, [SI]           ; TAKE X(1) TO AX
000C E7E4                         OUT      CLA← WRX1, AX      ; CLEAR ACC OF 45CM16
                                                                AND WRITE X(1)
0005 8B00                         MOV      AX, [SI+BX]        ; TAKE Y(1) TO AX
0010 E7E2                         OUT      CLA← WRY1, AX      ; CLEAR ACC OF 45CM16
                                                              ; AND WRITE Y(1)
                  ;
0012 49                           DEC      CX                 ; CX=CX-1
                  ;
0013 46                  REPEAT  : INC     SI                 ;
0014 46                           INC      SI                 ; SI=SI+2
                  ;
0015 8B04                         MOV      AX, [SI]           ; TAKE NEXT X TO AX
0017 E7EC                         OUT      ADDXY← WRX, AX     ; MULTIPLY X AND Y
                                                              ; AND ADD TO ACC
                                                              ; WRITE X TO 45CM16
```

```
LOC  OBJ                  LINE      SOURCE

0019 8B00                   88                        MOV     AX, [SI+RX]         ; TAKE NEXT Y TO AX
                            89      ;
001B E7F2                   90                        OUT     NOP+WRY, AX         ; WRITE NEXT Y TO 45CM16
                            91      ;
001D E2F4                   92                        LOOP    REPEAT              ; DECREMENT CX AND
                            93                                                    ; GOTO REPEAT IF
                            94      ;                                               CX NOT EQUAL TO 0
001F E7E8                   95                        OUT     ADDXY+NOP, AX       ; DO X(N)*Y(N) AND
                            96                                                    ; ADD TO ACC
0021 E5E0                   97                        IN      AX, RD+LOW          ; READ LOW 16 BITS
0023 894002                 98                        MOV     [SI+BX+2], AX       ; STORE AT SUM+LOW
0026 E5E2                   99                        IN      AX, RD+MID          ; READ MID 16 BITS
0028 894004                100                        MOV     [SI+BX+4], AX       ; STORE AT SUM+MID
002B E5E4                  101                        IN      AX, RD+HIGH         ; READ  HIGH 16 BITS
002D 894006                102                        MOV     [SI+BX+6], AX       ; STORE AT SUM+HIGH
                           103      ;
0030 C3                    104                        RET                         ; RETURN FROM SUBROUTINE
                           105      ;
                           106      ;          SUM             ENDP
                           107      ;
----                       108      ;          CODE            ENDS
                           109      ;
                           110      ;
                           111      ;          THIS SUBROUTINE IS WRITEN BY KAYLAN P. GOKHALE AT DEPARTMENT OF
                           112      ;          ELECTRICAL AND COMPUTER ENGINEERING, UNIVERSITY OF MISSOURI,
                           113      ;          COLUMBIA, MO 65211
                           114      ;
                           115                        END

ASSEMBLY COMPLETE: NO ERRORS FOUND
```

45CF8 FINITE IMPULSE RESPONSE FILTER

Introduction

The NCR45CF8 Finite Impulse Response Filter is a directly cascadable device which is designed for the implementation of video speed FIR filters with either linear or non-linear phase characteristics. Each chip contains four 9×8 parallel multipliers, along with adders and delays to implement a four-stage general FIR filter.

A unique architecture allows the NCR45CF8 to be cascaded to any number of stages in either the linear or nonlinear phase modes. For example, a 24-tap filter requires only three NCR45CF8 devices.

Multiple coefficients of each chip can be changed during each clock cycle. Two's complement or unsigned magnitude format is selected independently for each coefficient and each data input. Input data is nine bits and the coefficients are eight bits, with multiplication results rounded to 14 bits. The output is summed to 18 bits.

Two versions of the NCR45CF8 have been designed to allow implementation of FIR digital filters with both even and odd numbers of taps. In the linear phase mode, port B is connected to port C, which wraps around the partial sums. The NCR45CF8E (even) repeats the center coefficient while the NCR45CF8O (odd) does not repeat the center coefficient.

General Description

The NCR45CF8 is a directly cascadable finite impulse response (FIR) filter chip designed for the implementation of video speed FIR filters with either linear or non-linear phase characteristics. Each chip contains four 9×8 parallel multipliers, along with adders and delays, to implement either a four stage general FIR filter (with non-symmetric coefficients) or an eight stage linear phase FIR filter (with symmetric coefficients). A unique architecture allows the NCR45CF8 to be cascaded to any number of stages in either the linear or nonlinear phase modes. For example, a 24-tap linear phase FIR filter requirs only three NCR45CF8 devices.

For maximum versatility, one coefficient of each chip can be changed during each clock cycle. Two's complement or unsigned magnitude format is selected independently for each coefficient and each data input. Data inputs are nine bits and coefficients are eight bits. Multiplication results are rounded to fourteen bits and the output is summed to eighteen bits.

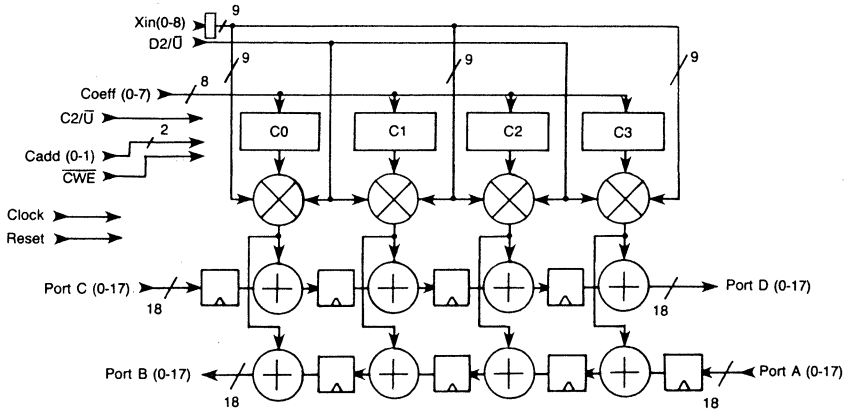
Features

- Directly cascadable with no external components
- 14.5 Mhz throughput rate
- Linear or non-linear phase operation
- 4 stages per chip in non-linear phase mode
- 8 stages per chip in linear phase mode
- TTL compatible
- Low power CMOS

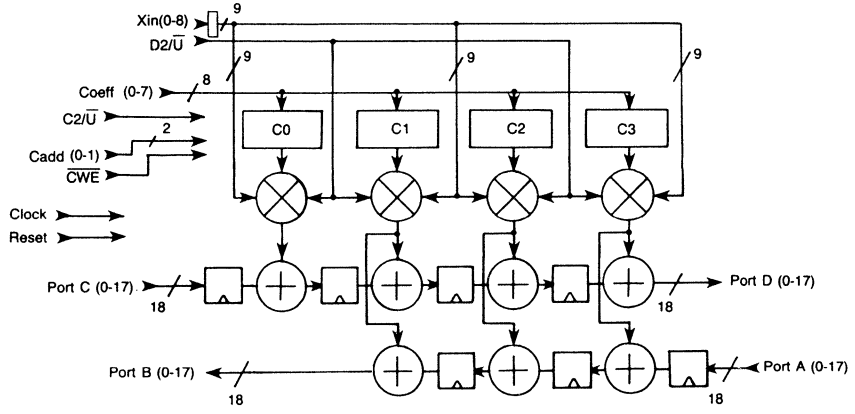
Applications

- Digital Video Filters
- 2-D Filtering
- Multi-bit Correlation
- Adaptive Filters

NCR45CF8E Block Diagram

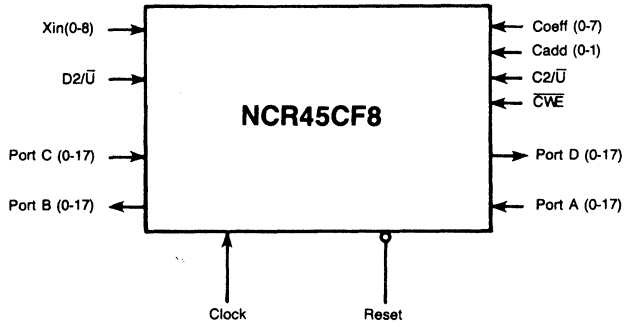


NCR45CF8O Block Diagram



Two versions of the NCR45CF8 are available. This is to allow linear phase digital filters with either an even or an odd number of taps to be implemented. In the linear phase mode, port B is connected to port C to wrap around the partial sums. The NCR45CF8E (even) repeats the center coefficient while the NCR45CF8O (odd) does not repeat the center coefficient.

Logic Diagram



PIN NAME	FUNCTION
Data	Data Input
D2/U	Data Format Select
Coeff	Coefficient Input
Cadd	Coefficient Tap Address
C2/U	Coefficient Format Select
CWE	Coefficient Write Enable
Port A, C	Input Ports
Port B, D	Output Ports

NOTE: BIT 0 of the Input Ports, Output Ports, Data Input Bus and Coefficient Input Bus is the Least Significant bit.

The individual coefficients and data values can be input to the NCR45CF8 in either unsigned magnitude or two's complement format. For coefficient input, the C2/$\overline{U}$ input is held high for two's complement and low for unsigned magnitude format. For data value input, the D2/$\overline{U}$ input is held high for two's complement and low for unsigned magnitude format. These control lines may be changed independently for every clock cycle as a data value or a coefficient is input. Coefficients are updated by placing the new value on the coefficient input lines with the appropriate tap address on the Cadd lines and taking the $\overline{CWE}$ line low. One coefficient may be updated during each cycle. Coefficient storage is fully static, so the coefficients can be loaded once and will remain stable as long as the power is on. A reset line is provided to insure complete device initialization in a known state. When the reset line is pulled low, all internal registers, including data and coefficient registers, are initialized to zero. This assures proper filter output during the first few cycles of operation when data has not yet propagated the full length of the filter. A four cycle latency exists from data input to sum output due to internal pipelining.

The canonical FIR filter structure shown in figure 1a is equivalent to the structure of figure 1b. The second structure has the advantage of regularity and is easily cascaded. When used with non-symmetric coefficients, two NCR45CF8 chips will implement the structure shown in figure 1b. Four taps, or stages, of such a filter are implemented in each chip. In many instances, a linear phase FIR filter is desired. Linear phase is achieved by using coefficients which are symmetric about the center tap, or taps, of the filter. Figure 2 shows a linear phase filter.

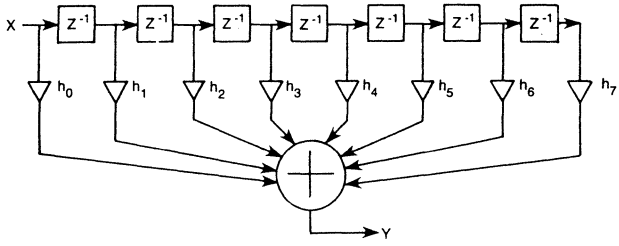


Figure 1a. Direct form (canonical) FIR filter structure.

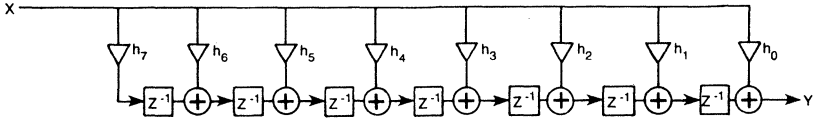


Figure 1b. Equivalent FIR filter structure.

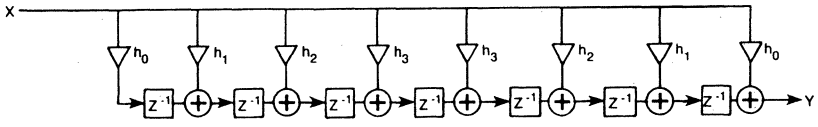


Figure 2. Linear phase FIR filter.

A significant savings in hardware can be realized by sharing the multipliers with the same coefficients. This reduces the number of multipliers required for a length-N filter from N to N/2. A structure that takes advantage of the symmetric coefficient characteristic of the linear phase mode, while maintaining cascadeability, is shown in figure 3.

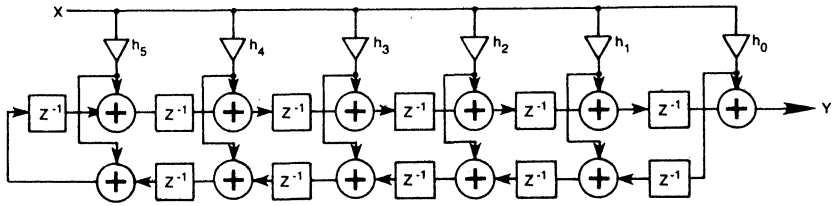


Figure 3. Linear phase FIR architecture using N/2 multipliers.

The NCR45CF8 device can be used as a four stage or tap, section of a FIR filter with arbitrary coefficients, such as previously shown in figure 1b. It can also be used to implement eight stages, or taps, of a linear phase FIR filter by externally connecting port B to port C. Multiple NCR45CF8 devices can be cascaded in either the linear or non-linear phase mode to create longer filters. This is shown in figure 4. Any number of NCR45CF8 devices can be cascaded electrically, but caution must be exercised to scale the coefficients so that overflow does not occur in the adder strings.

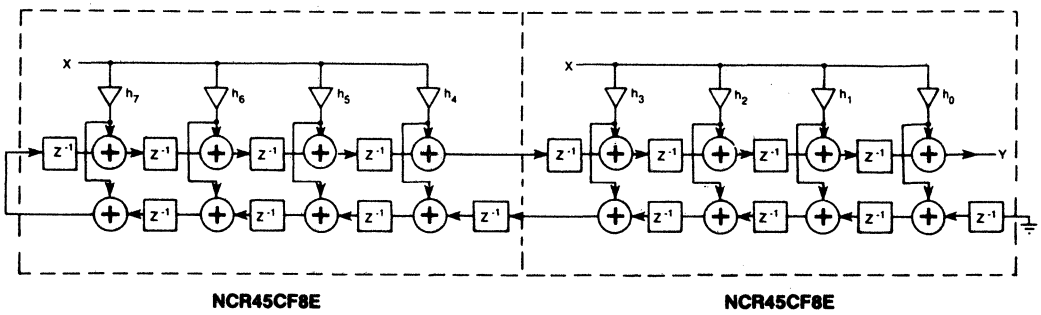


Figure 4. Cascading two NCR45CF8 chips to form a length-16 linear phase FIR filter.

System requirements can dictate a filter design to have either an odd or an even number of taps. For an odd tap filter with non-symmetric coefficients, the NCR45CF8E can be used with the coefficient of the first (leftmost) tap of the first (leftmost) chip set to zero. This neutralizes the effect of that tap since it will produce a zero product that will have no impact on the overall filter output. Doing this effectively cuts one tap off the filter, resulting in a filter with an odd number of taps.

Odd tap FIR filters with symmetric coefficients are implemented with a unique (non-repeated) center coefficient. This structure is shown in figure 5. The NCR45CF8E (even) is used for even tap filters and for cascaded sections of odd tap filters, while the NCR45CF8O (odd) is used for the leftmost device in an odd tap linear phase filter. Figure 6 shows both devices cascaded to form a 15-tap linear phase FIR filter. Note that the leftmost device must be the odd tap version since the center coefficient is not repeated in this filter.

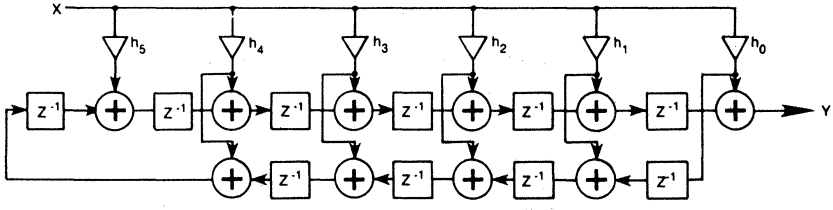


Figure 5. Odd tap linear phase FIR structure.

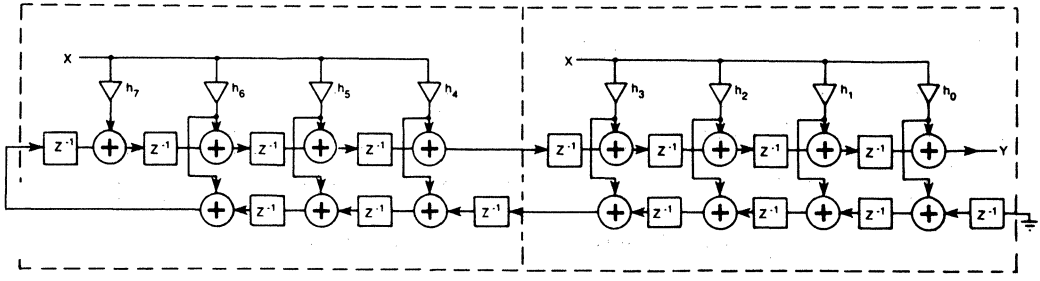


Figure 6. Cascading two NCR45CF8 chips to form a length-15 linear phase FIR filter.

Interleaving for Higher Speed Operation

By interleaving two (or more) filters the effective data rate can be doubled. The 8-tap FIR filter of figure 7 is limited to 14.5 Mhz operation. The same filter, implemented in four sections using twice the hardware, is shown in figure 8. The proper addition of the output of the four sections will cause a filter with 29 Mhz throughput to result. Figure 9 shows the interleave operation. Essentially, this scheme doubles the filter throughput by distributing computation between two sets of hardware.

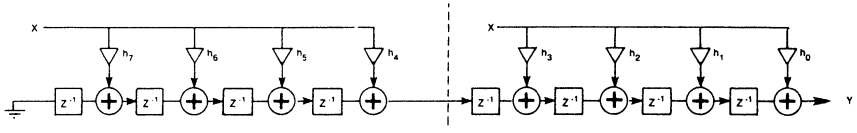


Figure 7. Eight tap filter using two NCR45CF8 devices (non-linear phase mode 14.5 MHz data rate).

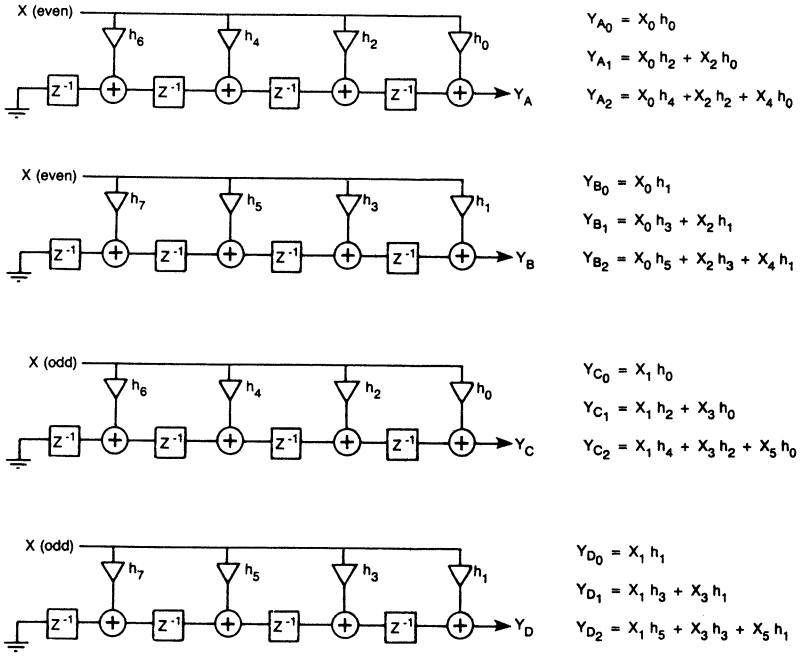


Figure 8. Four section equivalent FIR filter showing distribution of data and coefficients (14.5 MHz operation within sections).

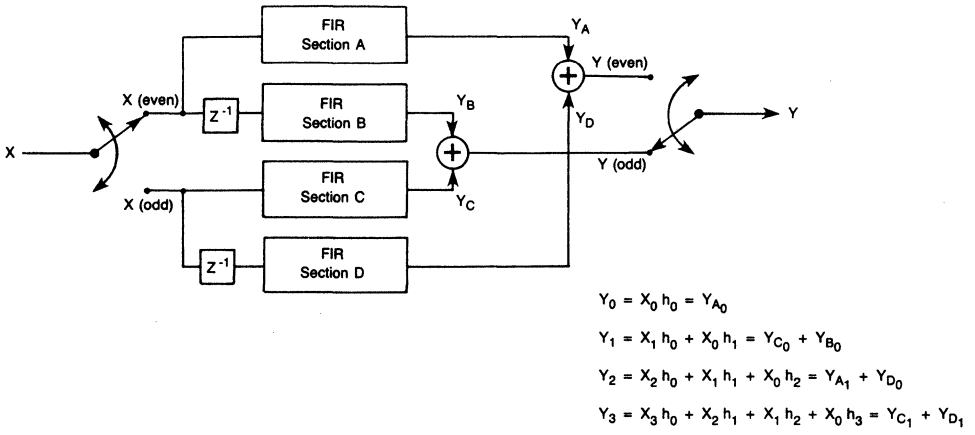


Figure 9. 29 MHz throughput interleaved FIR filter (14.5 MHz components).

Another application of the double throughput system is for interpolating FIR filters. A filter like the one shown in figure 10, where every other coefficient is zero, is often desired. By distributing the computation between two NCR45CF8 filter sections, a 29 Mhz interpolating FIR filter can be constructed. Figure 11 shows such a system.

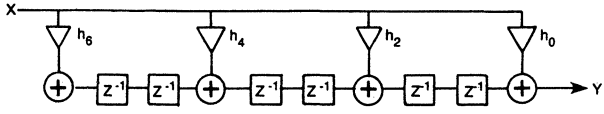


Figure 10. Interpolating FIR filter.

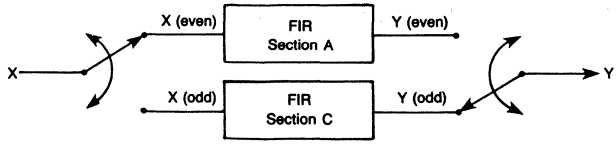
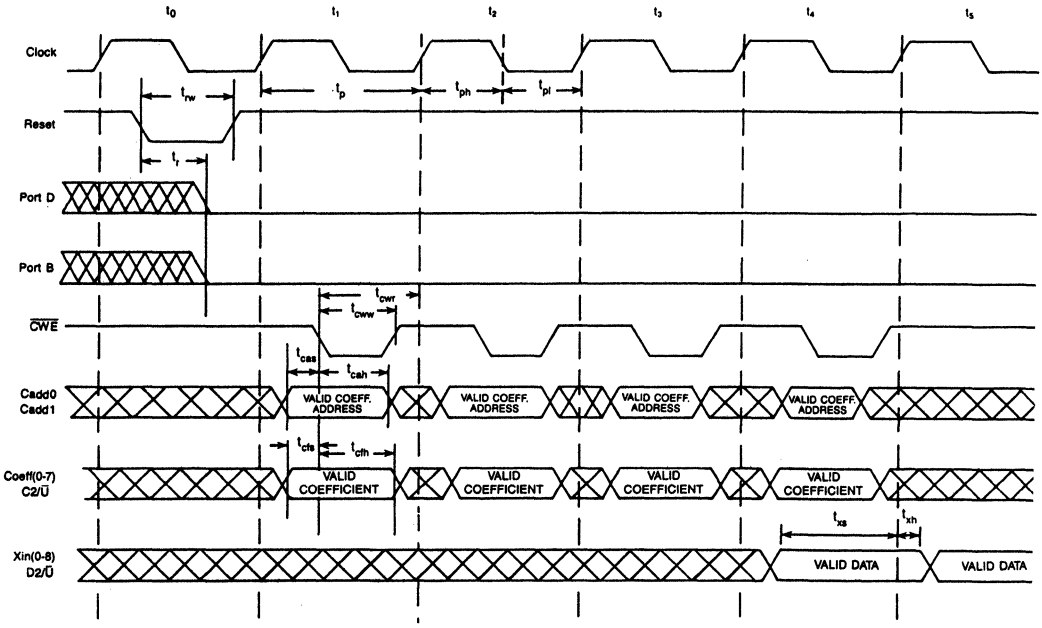


Figure 11. 29 MHz interpolating FIR filter using interleaved filter sections (14.5 MHz components).

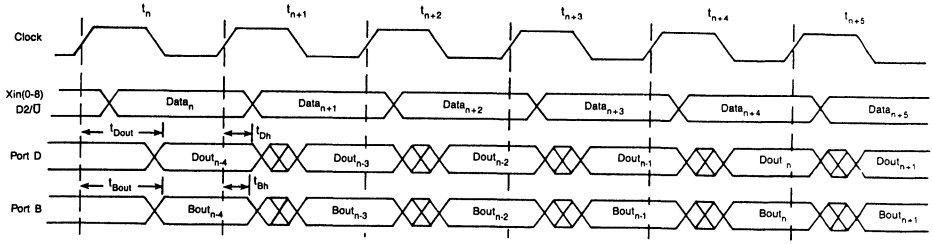
AC Characteristics over Recommended Operating Conditions

Symbol	Parameter	Min.	Max.	Units
t_p	clock period	69		ns
t_{ph}	clock high time	29		ns
t_{pl}	clock lot time	29		ns
t_r	reset time		50	ns
t_{rw}	reset pulse width	30		ns
t_{xh}	data hold time	12.5		ns
t_{xs}	data setup time	5		ns
t_{cww}	coefficient write pulse width	12.5		ns
t_{cas}	coefficient address setup time	0		ns
t_{cah}	coefficient address hold time	15		ns
t_{cfs}	coefficient setup time	5		ns
t_{cfh}	coefficient hold time	12.5		ns
t_{cwr}	coefficient write recovery time	25		ns
t_{Dout}	clock to Port D valid		55	ns
t_{Dh}	clock to Port D invalid	20		ns
t_{Bout}	clock to Port B valid		55	ns
t_{Bh}	clock to Port B invalid	20		ns
t_{As}	Port A setup time	0		ns
t_{Ah}	Port A hold time	15		ns
t_{Cs}	Port C setup time	0		ns
t_{Ch}	Port C hold time	15		ns

Timing Diagram I - Coefficient Initialization Sequence

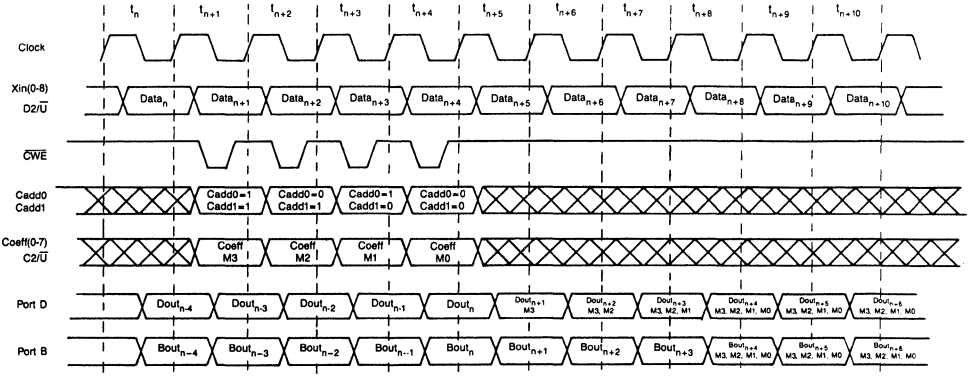


Timing Diagram II - Free Running Mode with Stable Coefficients



This shows the filter in a free-running mode with coefficients already loaded and stable. $Dout_n$ and $Bout_n$ are the first outputs using the $Data_n$. Thus, a latency of four cycles is present in this filter. Both CWE and RESET are assumed to be held high (inactive) in this mode. The C2/U and Coeff (0-7) inputs are don't cares.

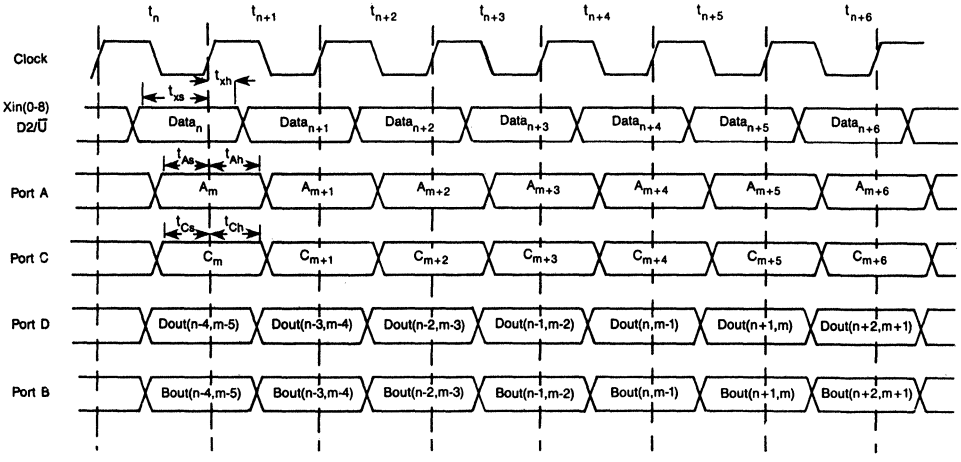
Timing Diagram III - Free Running Mode with Changing Coefficients



Timing diagram III shows the latency effects on filter output from changing the coefficients while the filter is running. One coefficient can be changed during each cycle. It should be noted that a four cycle latency exists between a coefficient update and output data effected by the update. This latency is due to internal delays in the multipliers. The sequence of the coefficient update is arbitrary and is controlled by the Cadd0 and Cadd1 lines. The M3, M2, M1, M0 notation on the Coeff(0-7) inputs and the Port B and Port D outputs shows how the output data is effected by this particular sequence of coefficient updates. It should also be noted that with this sequence of coefficient updates (M3, M2, M1, M0), Port D sees the effect on output one at a time, while Port B sees the effect all at once.

Timing Diagram IV - Input Port Timing

Timing diagram IV shows the relationship of the Port B and Port D outputs to the filter inputs: Xin, Port A, and Port C. At time t_n, Datan is input through Xin, while A_m and C_m are input through Ports A anc C, respectively. The four cycle latency discussed previously is evident since the Port B and D outputs do not depend on $Data_n$ until t_{n+4}. A_m and C_m both experience a five cycle latency before contributing to the Port B and D outputs, respectively, at t_{n+5}. This diagram shows a FIR filter used as a "non-end" chip in a cascaded, multi-chip linear phase configuration. If it was the lead chip, Port A would be grounded. If it was the left end chip (Port B tied to Port C of the same NCR45CF8E), then an input A_m at t_n would not contribute to the Port D output unitl t_{n+9}.



Absolute Maximum Ratings

Supply Voltage, V_{DD} ..... +7V

Voltage on any pin with respect to ground ..... −0.3 to VDD + 0.3V

Storage temperature ..... −65 °C to 150 °C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operation conditions stipulated elsewhere in this specification is not implied.

Recommended Operating Conditions

Symbol	Parameter	Min.	Max.	Units
V_{DD}	Supply voltage	4.75	5.25	Volts
V_{IH}	Input high level voltage (All inputs except Input Ports)	2.0	V_{DD}	Volts
V_{IHC}*	Input high level voltage, Parts A and C.	.7 V_{DD}	V_{DD}	Volts
V_{IL}	Input low level voltage	0	0.8	Volts
T_A	Operating ambient temperature	0	70	°C

*Port A and Port C use CMOS Input levels, all other pins use TTL levels.

Static Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Condition	Min.	Max.	Units
I_{IN}	Input leakage current	V_{IN}=0V to V_{DD}max		±10	μA
I_O	Output leakage current	V_O=0.4 to V_{DD}max		±10	μA
V_{OH}	Output high voltage	I_{OH}=4.0mA	2.4		Volts
V_{OL}	Output low voltage	I_{OL}=4.0mA		0.8	Volts
I_{DD}	Supply current — Active	Outputs Loaded		130	mA
	Supply current — Standby	Clock Inactive		10	mA
I_{SC}	Short circuit Output current †	One Output Shorted to ground or power		75	mA

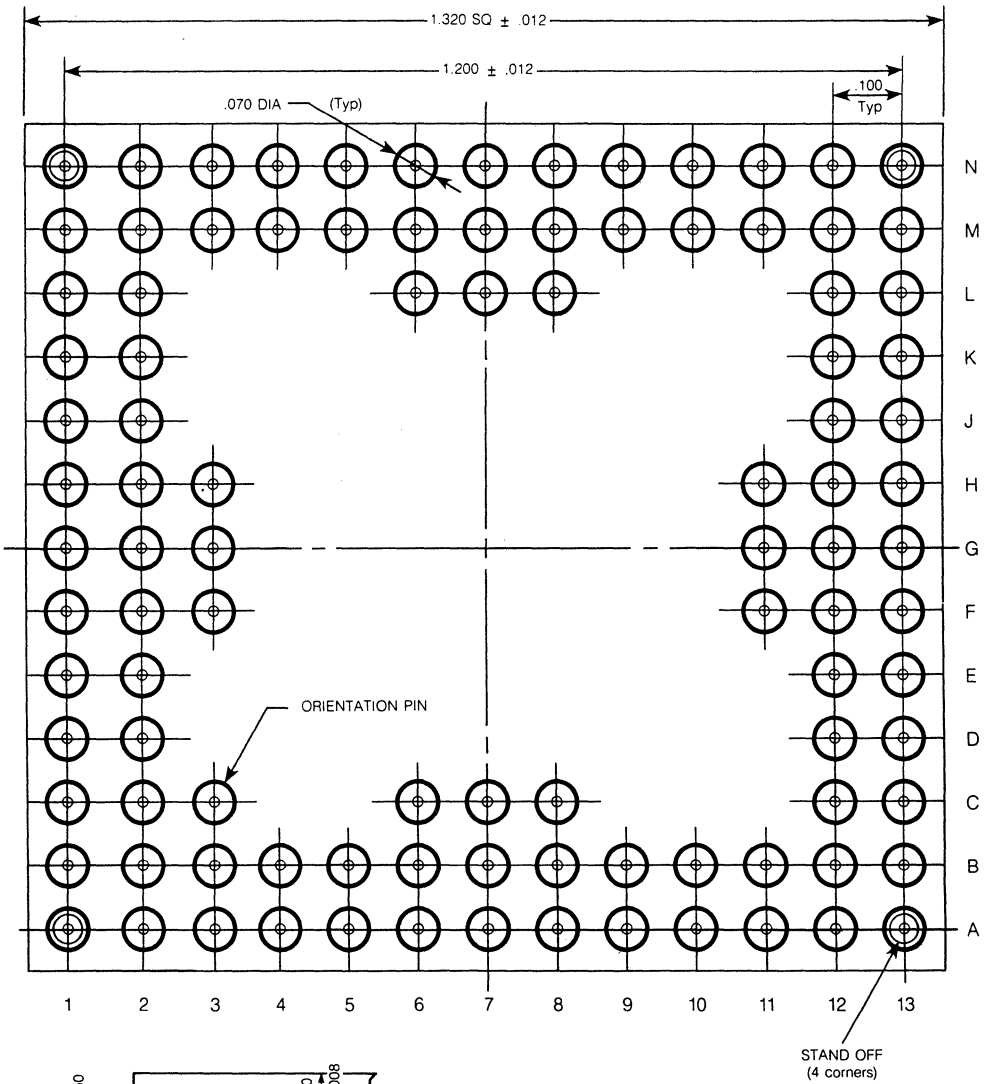
† Operating the device with Outputs Shorted to power or ground may result in permanent damage to the chip.

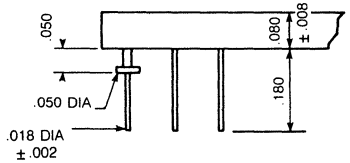
Capacitance TA = 25 °C, f = 1 MHz

Symbol	Parameter	Condition	Min.	Max.	Units
C_{IN}	Input capacitance	All pins except pin		5	pF
C_O	Output capacitance	under test are tied to ground		10	pF

Plastic Pin Grid Array Package

Bottom View





Side View

NOTE: ALL DIMENSIONS ARE IN INCHES

Table of Pin Numbers vs. Signal Labels for Plastic Pin Grid Array Package

PIN #	LABEL	PIN #	LABEL	PIN #	LABEL
A1	C11	E1	C3	M1	B8
A2	C13	E2	C4	M2	B5
A3	C14	E12	D13	M3	B3
A4	C16	E13	D14	M4	B1
A5	Xin5	F1	C0	M5	$\overline{CWE}$
A6	Xin8	F2	C1	M6	C2/$\overline{U}$
A7	D2/$\overline{U}$	F3	C2	M7	Coeff6
A8	Xin3	F11	D15	M8	Coeff3
A9	Xin0	F12	D16	M9	Coeff0
A10	D1	F13	D17	M10	A16
A11	D3	G1	B17	M11	A13
A12	D5	G2	Reset	M12	A11
A13	D7	G3	Vss	M13	A10
B1	C9	G11	Vss	N1	B6
B2	C10	G12	A0	N2	B4
B3	C12	G13	Clock	N3	B2
B4	C15	H1	B16	N4	B0
B5	C17	H2	B15	N5	Cadd0
B6	Xin7	H3	B14	N6	Coeff7
B7	Xin4	H11	A3	N7	Coeff5
B8	Xin2	H12	A2	N8	Coeff4
B9	D0	H13	A1	N9	Coeff1
B10	D2	J1	B13	N10	A17
B11	D4	J2	B12	N11	A15
B12	D6	J12	A5	N12	A14
B13	D9	J13	A4	N13	A12
C1	C7	K1	B11		
C2	C8	K2	B10		
C3	***	K12	A7		
C6	Xin6	K13	A6		
C7	Vdd	L1	B9		
C8	Xin1	L2	B7		
C12	D8	L6	Cadd1		
C13	D10	L7	Vdd		
D1	C5	L8	Coeff2		
D2	C6	L12	A9		
D12	D11	L13	A8		
D13	D12				

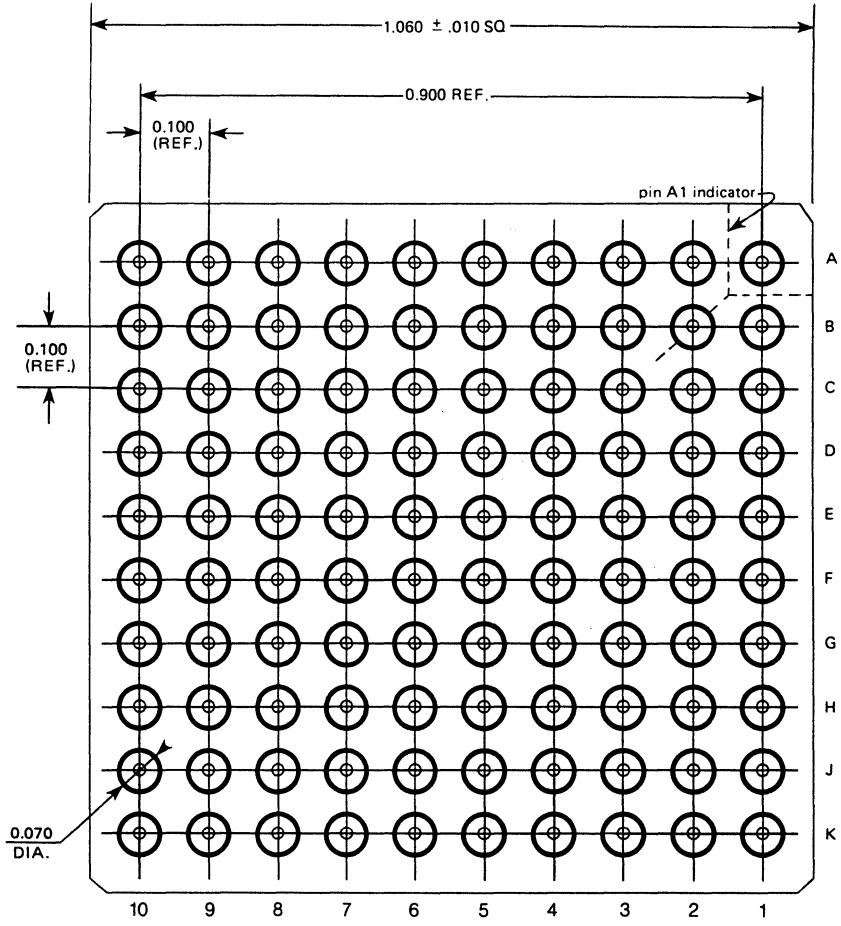
*** PIN NUMBER C3 is for ORIENTATION PURPOSES ONLY. This PIN is electrically floating.

Table of Signal Names vs. Pin Numbers for Plastic Pin Grid Array Package

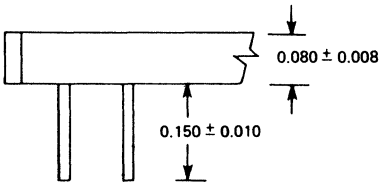
SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #
A0	G12	B7	L2	C14	A3	Coeff3	M8
A1	H13	B8	M1	C15	B4	Coeff4	N8
A2	H12	B9	L1	C16	A4	Coeff5	N7
A3	H11	B10	K2	C17	B5	Coeff6	M7
A4	J13	B11	K1	D0	B9	Coeff7	N6
A5	J12	B12	J2	D1	A10	Xin0	A9
A6	K13	B13	J1	D2	B10	Xin1	C8
A7	K12	B14	H3	D3	A11	Xin2	B8
A8	L13	B15	H2	D4	B11	Xin3	A8
A9	L12	B16	H1	D5	A12	Xin4	B7
A10	M13	B17	G1	D6	B12	Xin5	A5
A11	M12	C0	F1	D7	A13	Xin6	C6
A12	N13	C1	F2	D8	C12	Xin7	B6
A13	M11	C2	F3	D9	B13	Xin8	A6
A14	N12	C3	E1	D10	C13	Cadd0	N5
A15	N11	C4	E2	D11	D12	Cadd1	L6
A16	M10	C5	D1	D12	D13	C2/$\overline{U}$	M6
A17	N10	C6	D2	D13	E12	D2/$\overline{U}$	A7
B0	N4	C7	C1	D14	E13	$\overline{CWE}$	M5
B1	M4	C8	C2	D15	F11	Clockin	G13
B2	N3	C9	B1	D16	F12	Reset	G2
B3	M3	C10	B2	D17	F13	Vdd	C7
B4	N2	C11	A1	Coeff0	M9	Vdd	L7
B5	M2	C12	B3	Coeff1	N9	Vss	G3
B6	N1	C13	A2	Coeff2	L8	Vss	G11

Ceramic Pin Grid Array Package

Bottom View



All dimensions are in inches



Side View

Table of Pin Numbers vs. Signal Labels for Ceramic Pin Grid Array Package

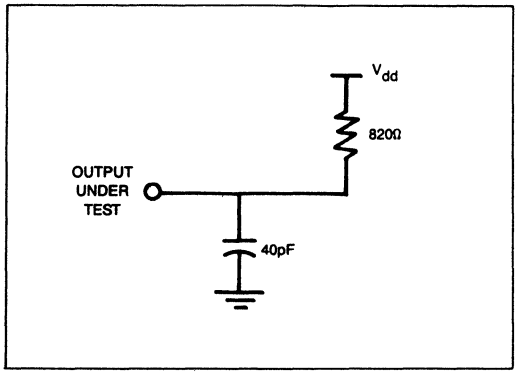
PIN #	LABEL	PIN #	LABEL	PIN #	LABEL
A1	D11	E1	Xin4	J1	Xin7
A2	D16	E2	Xin3	J2	C12
A3	D17	E3	Xin2	J3	C8
A4	Clock	E4	Xin1	J4	C5
A5	D14	E5	D0	J5	C0
A6	A0	E6	A5	J6	B14
A7	A4	E7	A14	J7	B12
A8	A8	E8	V_{dd}	J8	B11
A9	A12	E9	Coeff2	J9	B7
A10	A16	E10	Coeff1	J10	B6
B1	D7	F1	Xin5	K1	C15
B2	D8	F2	Xin6	K2	C11
B3	D12	F3	V_{dd}	K3	C7
B4	D13	F4	C13	K4	C3
B5	D15	F5	C4	K5	Reset
B6	A1	F6	$\overline{CWE}$	K6	B13
B7	A6	F7	Cadd1	K7	B17
B8	A9	F8	C2/$\overline{U}$	K8	B16
B9	A13	F9	Coeff7	K9	B15
B10	Coeff3	F10	Coeff6	K10	B10
C1	D3	G1	D2/$\overline{U}$		
C2	D4	G2	C17		
C3	D6	G3	C14		
C4	D10	G4	C9		
C5	V_{ss}	G5	C2		
C6	A2	G6	B8		
C7	A7	G7	B4		
C8	A11	G8	B1		
C9	A17	G9	B0		
C10	Coeff4	G10	Cadd0		
D1	Xin0	H1	Xin8		
D2	D1	H2	C16		
D3	D2	H3	C10		
D4	D5	H4	C6		
D5	D9	H5	C1		
D6	A3	H6	V_{ss}		
D7	A10	H7	B9		
D8	A15	H8	B5		
D9	Coeff0	H9	B3		
D10	Coeff5	H10	B2		

NCR45CF8 Finite Impulse Response Filter

Table of Pin Numbers vs. Signal Labels for Ceramic Pin Grid Array Package

SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #	SIGNAL LABEL	PIN #
A0	A6	B7	J9	C14	G3	Coeff3	B10
A1	B6	B8	G6	C15	K1	Coeff4	C10
A2	C6	B9	H7	C16	H2	Coeff5	D10
A3	D6	B10	K10	C17	G2	Coeff6	F10
A4	A7	B11	J8	D0	E5	Coeff7	F9
A5	E6	B12	J7	D1	D2	Xin0	D1
A6	B7	B13	K6	D2	D3	Xin1	E4
A7	C7	B14	J6	D3	C1	Xin2	E3
A8	A8	B15	K9	D4	C2	Xin3	E2
A9	B8	B16	K8	D5	D4	Xin4	E1
A10	D7	B17	K7	D6	C3	Xin5	F1
A11	C8	C0	J5	D7	B1	Xin6	F2
A12	A9	C1	H5	D8	B2	Xin7	J1
A13	B9	C2	G5	D9	D5	Xin8	H1
A14	E7	C3	K4	D10	C4	Cadd0	G10
A15	D8	C4	F5	D11	A1	Cadd1	F7
A16	A10	C5	J4	D12	B3	C2/$\overline{U}$	F8
A17	C9	C6	H4	D13	B4	D2/$\overline{U}$	G1
B0	G9	C7	K3	D14	A5	$\overline{CWE}$	F6
B1	G8	C8	J3	D15	B5	Clock	A4
B2	H10	C9	G4	D16	A2	Reset	K5
B3	H9	C10	H3	D17	A3	V_{dd}	E8
B4	G7	C11	K2	Coeff0	D9	V_{dd}	F3
B5	H8	C12	J2	Coeff1	E10	V_{ss}	C5
B6	J10	C13	F4	Coeff2	E9	V_{ss}	H6

AC Test Load Circuit



CAUTION

1. CMOS Devices are damaged by high energy electrostatic discharge. Devices must be stored in conductive foam or with all pins shunted.
2. Remove power before insertion or removal of this device.

Application Notes

The ground pins (Vss) of the NCR45CF8 are not connected internally. In all applications, both ground pins must be connected externally to ground. We also recommend connecting both power pins (Vdd) externally to power in all applications. The NCR45CF8 device is a high speed CMOS integrated circuit, which will dissipate very little static power but will also consume significant dynamic current. We recommend the use of at least one 01.μf deglitching capacitor per NCR45CF8 chip to reduce switching noise on the system power buss. To prevent ringing, care should be taken to follow standard engineering practice for minimizing power and ground lead inductance.

45CF8 CMOS FIR FILTER CHIP

Quick Fact Sheet

The NCR45CF8 is a high speed digital FIR filter optimized for real time video and image processing applications. It's unique architecture allows linear and nonlinear phase operation for noise reduction filtering, anti-aliasing/reconstruction filtering, 2-D filtering, and chroma/luminance separation. In addition, the design of the NCR45CF8 allows multiple chips to be cascaded to form arbitrary length FIR filters, without external components or "glue" logic. Even and odd tap versions are available (NCR45CF8E and NCR45CF8O) for implementation of even or odd length linear phase FIR filters.

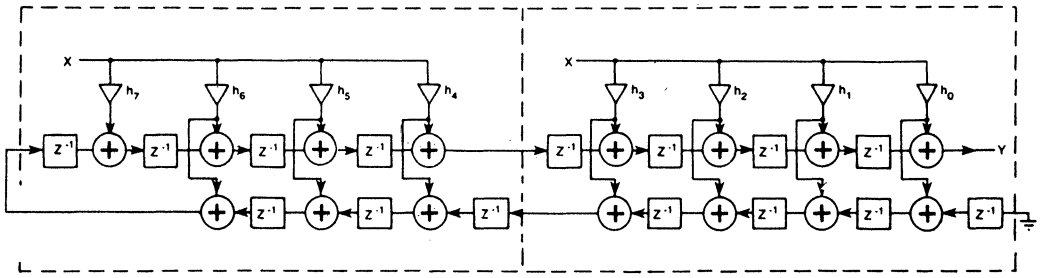
Features:

- 14.5 MHz data throughput rate
- 9 bit data words
- 8 bit coefficients
- 8 stages/chip in linear phase mode
- 4 stages/chip in nonlinear phase mode
- Maximum 750mW power dissipation
- 100 pin PGA package

Applications:

- Broadcast video equipment
- Machine/robot vision systems
- Factory automation systems
- Image processing
- Video teleconferencing
- Medical imaging systems

Currrently available in plastic or ceramic pin grid array packages, the NCR45CF8 replaces board level FIR filters, reducing component cost, power dissipation and required board space. The 9 bit data and 8 bit coefficient widths provide the resolution needed for video and image processing applications. For applications requiring higher resolution such as medical imaging, NCR45CF8's can be used in parallel to obtain 14 bit data width. Designed for ease of use, the NCR45CF8 requires a minimum of support logic. A simulator (NCR45FSM) is also available, enabling the system designer to simulate the filters internal bit - widths and roundoff characteristics.



NCR45CF8O NCR45CF8E

The NCR45FSM simulator runs on MS-DOS personal computers with graphics capability. (CGA and Hercules standards are supported.) Your NCR field applications engineer or sales representative can arrange a demonstration.

Key Points to Remember about the NCR45CF8

- High Speed for Rea l Time Video/Image Processing
- Easy to Use
- Directly Cascadable to form Arbitrary Length FIR Filters
- The NCR45CF8 Replaces board level FIR filters

  Reducing chip count

  Reducing power dissipation

  Reducing board space requirements

  Increasing reliability

  All while still providing performance for real time processing!

MEMORY PRODUCTS

NMOS ROMS

Introduction

NCR offers NMOS ROMs ranging in sizes from 16K to 256K fabricated in a 2.75u technology. ROM patterns are accepted in EPROM. If additional information is needed contact: NCR - Fort Collins, C.O.T. Department.

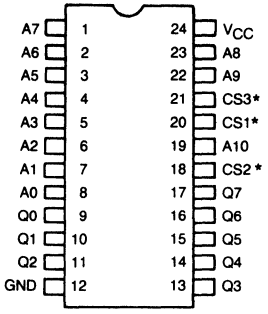
16K (2K X 8) ROM

2316

- **Fully Static Operation**
- **Silicon Gate NMOS Technology**
- **Programmable Chip Selects**
- **Maximum Access Time**
  - **2316-20 ..... 200ns**
  - **2316-25 ..... 250ns**
  - **2316-30 ..... 300ns**
- **3-State Outputs**
- **Fully TTL Compatible**
- **Single ±10% 5 Volt Supply**
- **Pin Compatible with 2716 and 2516 EPROM's and 2316 Type ROM's.**
- **0°C to 70°C Operating Range.**

The NCR2316 is a mask programmable read-only-memory with a 2K word by 8-bit organization. Designed for ease of use, this device requires only a 5 volt supply, is TTL compatible, and because of its totally static (asynchronous) operation requires no clock. The active level of the three-chip select inputs are programmable and are defined by the user. These ROMs are available in a 24 pin package and are pin compatible with industry standard EPROMs and ROMs.

PIN CONFIGURATION

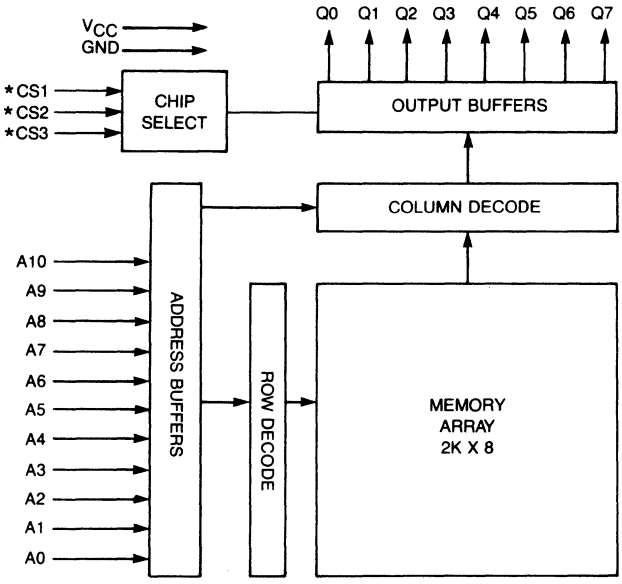


* Active Level (Hi, Low or Don't Care) of chip selects are defined by user.

PIN NAMES

A0 - A10	Address Inputs
Q0 - Q7	Data Outputs
CS1 - CS3	Programmable Chip Selects
V_{CC}	5V ± 10% Supply Voltage

FUNCTIONAL BLOCK DIAGRAM



* MASK PROGRAMMABLE

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground $-.5$ to $+7V$
Storage temperature . . . . . $-65°C$ to $150°C$

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Supply voltage*	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.5		0.8	Volts
T_A	Operating ambient temperature	0		70	°C

* V_{CC} must be applied at least 100μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IN}	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max, Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH}=−160μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL}=1.6mA			0.4	Volts
I_{CC}	Supply current	Outputs Open			75	mA

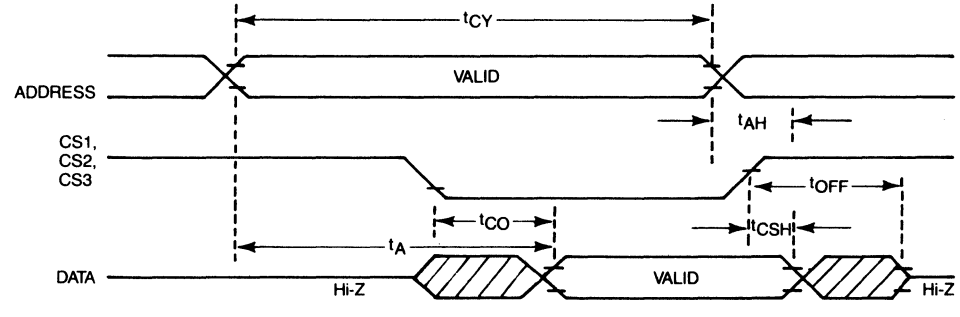
CAPACITANCE, T_A = 25°C, f=1 MHZ

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
C_{IN}	Input capacitance	All pins except pin under			7	pF
C_O	Output capacitance	test are tied to ground			12.5	pF

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

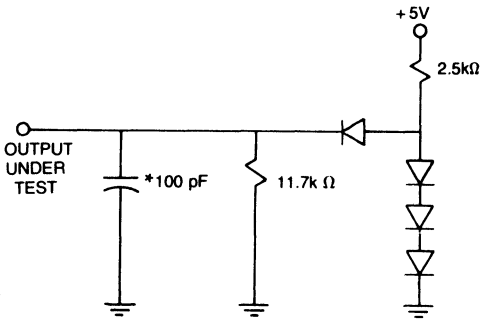
Symbol	Parameter	2316-20		2316-25		2316-30		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CY}	Cycle time	200		250		300		ns
t_A	Address access time		200		250		300	ns
t_{CO}	Chip select access time		100		125		150	ns
t_{OFF}	Chip select to data off (Hi Z)		100		125		150	ns
t_{CSH}	Chip select to data hold	0		0		0		ns
t_{AH}	Address to data hold	0		0		0		ns

AC WAVEFORMS



The chip select lines are assumed to be active low.

AC TEST LOAD CIRCUIT



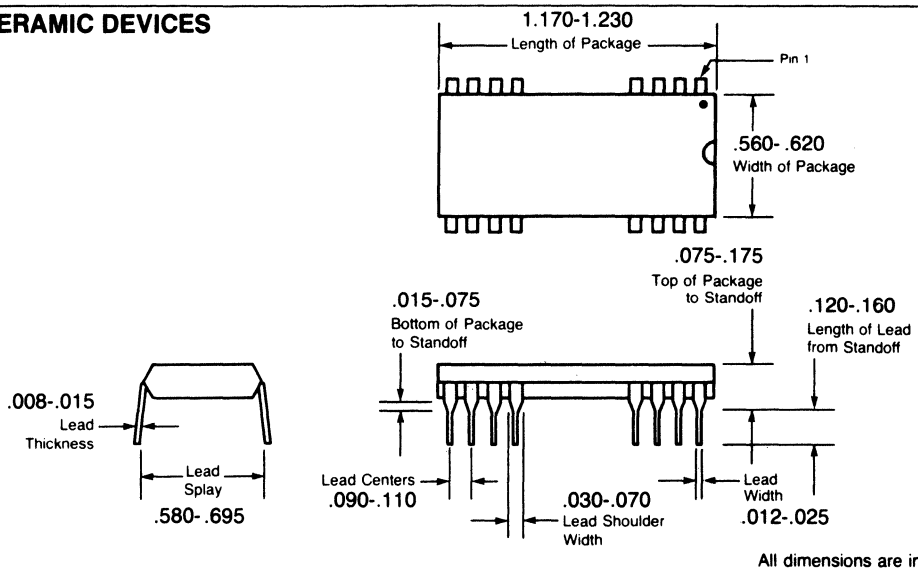
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

A.C. CONDITION OF TESTS

Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times ..... 10 ns
Output Timing Levels..... 0.8 Volts to 2.0 Volts

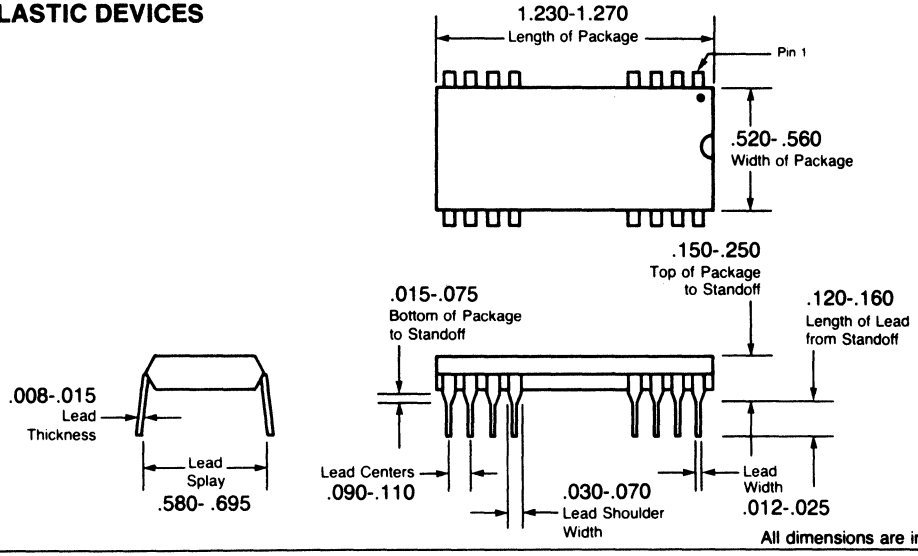
MECHANICAL DATA 24 PIN

CERAMIC DEVICES



All dimensions are inches.

PLASTIC DEVICES



All dimensions are inches.

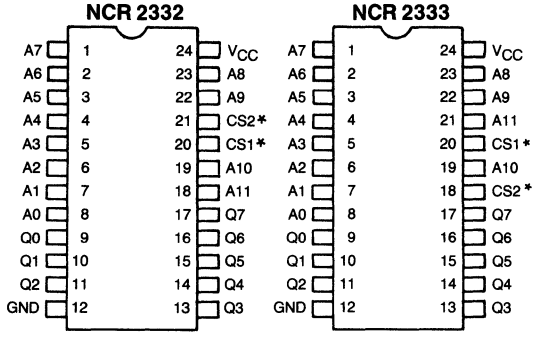
32K (4K X 8) ROM

2332/33

- **Fully Static Operation**
- **Silicon Gate NMOS Technology**
- **Programmable Chip Selects**
- **Maximum Access Time**
  - **2332/33-20 ..... 200ns**
  - **2332/33-25 ..... 250ns**
  - **2332/33-30 ..... 300ns**
- **3-State Outputs**
- **Fully TTL Compatible**
- **Single ± 10% 5 Volt Supply**
- **2332 Pin Compatible with 2532 EPROM's.**
- **2333 Pin Compatible with 2732 EPROM's.**
- **0°C to 70°C Operating Range.**

The NCR2332 and 2333 are mask programmable read-only-memories with 4K word by 8-bit organizations. Designed for ease of use, these devices require only a 5 volt supply, are TTL compatible, and because of their totally static (asynchronous) operation require no clock. The active level of the two chip select inputs are programmable and are defined by the user. These ROMs are available in a 24 pin package with industry standard byte-wide JEDEC pin-outs. The NCR 2332 is pin compatible with 2532 EPROM's and the NCR 2333 is pin compatible with 2732 EPROMS's.

PIN CONFIGURATION

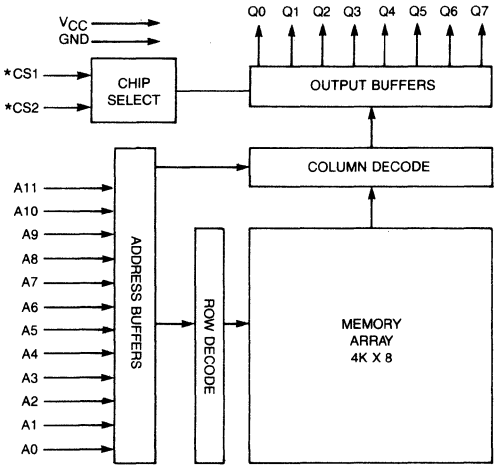


* Active Level (Hi, Low or Don't Care) of chip selects are defined by user.

PIN NAMES

A0 - A11	Address Inputs
Q0 - Q7	Data Outputs
CS1 - CS2	Programmable Chip Selects
V_{CC}	5V ± 10% Supply Voltage

FUNCTIONAL BLOCK DIAGRAM



*MASK PROGRAMMABLE

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground $-.5$ to $+7V$
Storage temperature ..... $-65°C$ to $150°C$

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min.	Typ.	Max.	Units
V_{CC}	Supply voltage*	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.5		0.8	Volts
T_A	Operating ambient temperature	0		70	°C

* V_{CC} must be applied at least 100μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IN}	Input leakage current	V_{IN}=OV to V_{CC} max			10	μA
I_O	Output leakage current	V_O=0.2 to V_{CC} max, Chip Deselected			±10	μA
V_{OH}	Output high voltage	$I_{OH}=-160\mu A$	2.4			Volts
V_{OL}	Output low voltage	I_{OL}=1.6mA			0.4	Volts
I_{CC}	Supply current	Outputs Open			75	mA

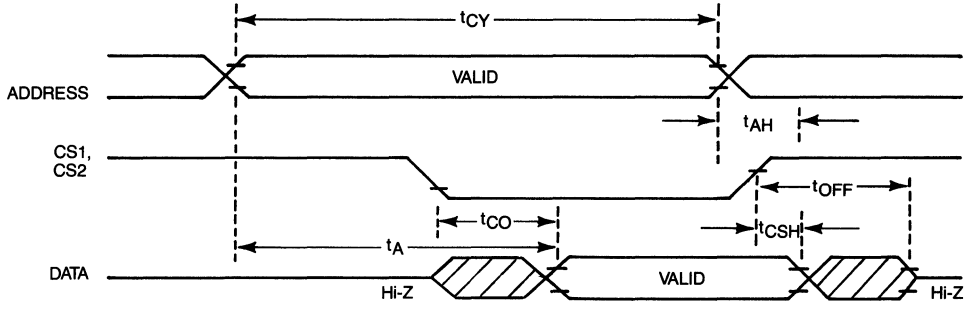
CAPACITANCE, T_A = 25°C, f=1 MHZ

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
C_{IN}	Input capacitance	All pins except pin under			7	pF
C_O	Output capacitance	test are tied to ground			12.5	pF

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

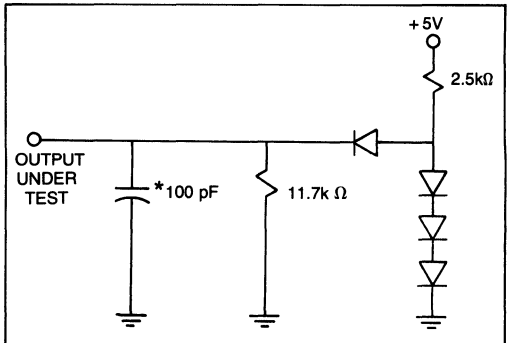
Symbol	Parameter	2332/33-20		2332/33-25		2332/33-30		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CY}	Cycle time	200		250		300		ns
t_A	Address access time		200		250		300	ns
t_{CO}	Chip select access time		100		125		150	ns
t_{OFF}	Chip select to data off (Hi Z)		100		125		150	ns
t_{CSH}	Chip select to data hold	0		0		0		ns
t_{AH}	Address to data hold	0		0		0		ns

AC WAVEFORMS



The chip select lines are assumed to be active low.

AC TEST LOAD CIRCUIT



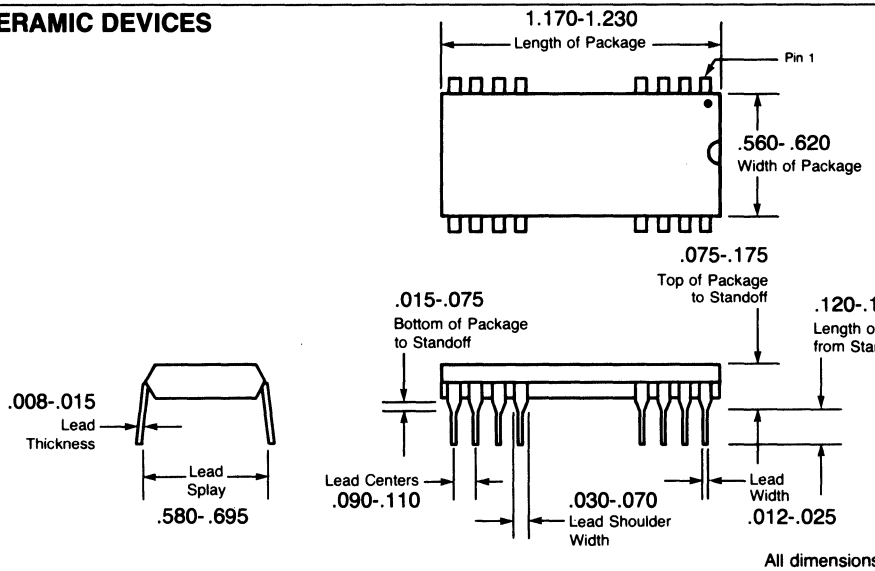
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

A.C. CONDITION OF TESTS

Input Pulse Levels . . . . . .0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times . . . . . .10 ns
Output Timing Levels . . . . . .0.8 Volts to 2.0 Volts

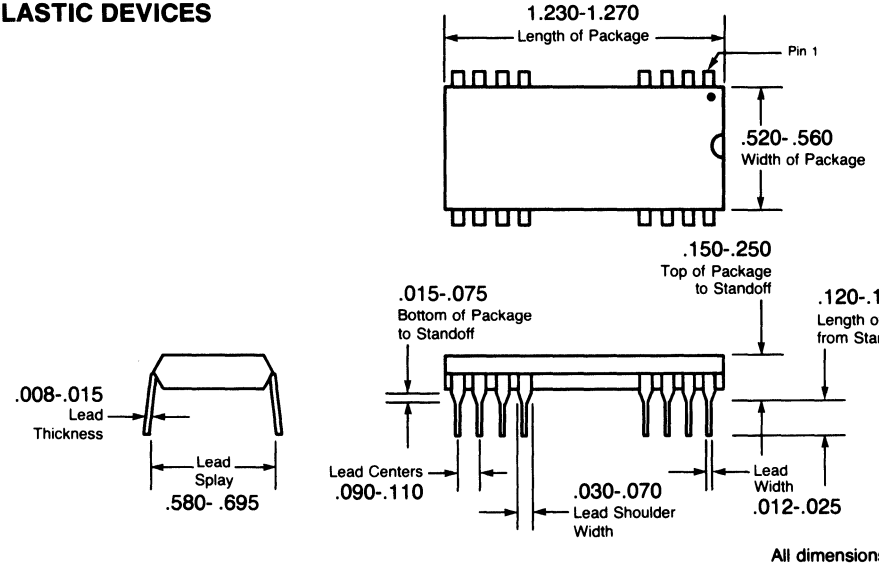
MECHANICAL DATA 24 PIN

CERAMIC DEVICES



All dimensions are inches.

PLASTIC DEVICES



All dimensions are inches.

64K (8K X 8) ROM

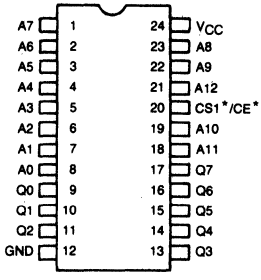
2364/65

- **2364/65 ..... Non-Power Down**
- **2364S/65S ..... Automatic Power Down**
- **Fully Static Operation**
- **Silicon Gate NMOS Technology**
- **Maximum Access Time**
  - **2364/65-20 ..... 200ns**
  - **2364/65-25 ..... 250ns**
  - **2364/65-30 ..... 300ns**
- **Programmable Chip Selects**
- **3-State Outputs**
- **Fully TTL Compatible**
- **Single ± 10% 5 Volt Supply**
- **Pin Compatible with 2564 EPROM's**
- **Available in 3 Temperature Ranges**
  - **2364/65 (Commercial) ..... 0°C to 70°C**
  - **2364I/65I (Industrial) ..... −40°C to 85°C**
  - **2364HR/65HR (Military) ..... −55°C to 125°C**

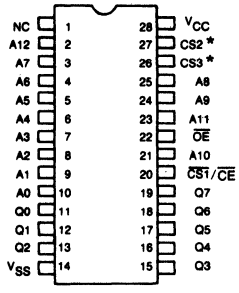
The NCR 2364 and 2365 are mask programmable read-only-memories with 8K word by 8 bit organizations. Designed for ease of use, these devices require only a 5-volt supply, are TTL compatible, and because of their totally static (asynchronous) operation require no clock. These memory devices are available in two versions. The 2364 and 2365 are non-power down versions where the active level of chip selects CS1 (on the 2364), and CS2 and CS3 (on the 2365) are programmable and defined by the user to facilitate system memory expansion. The 2364S and 2365S are standby versions offering an automatic power-down feature controlled by the chip enable CE input. When CE goes high, the device automatically powers down and remains in a low power standby mode as long as CE remains high. Also to provide easier system implementation, the active level of chip enable CE (on the 2364S), and chip selects CS2 and CS3 (on the 2365S) is programmable. The NCR 2364 is packaged in a 24 pin DIP, and the 2365 is packaged in a 28 pin DIP, both with industry standard byte-wide JEDEC pin-outs. Optionally, the 2365 is available in a space saving 28 pin surface mounted plastic leaded chip carrier.

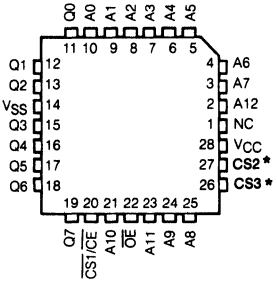
PIN CONFIGURATION

NCR 2364/S



NCR 2365/S





* Active Level (Hi, Low or Don't Care) defined by user.

PIN NAMES

A0 - A12	Address Inputs
Q0 - Q7	Data Outputs
CS1, $\overline{CS1}$ CS2, CS3	Programmable Chip Select
CE, $\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
V_{CC}	5V ± 10% Supply Voltage

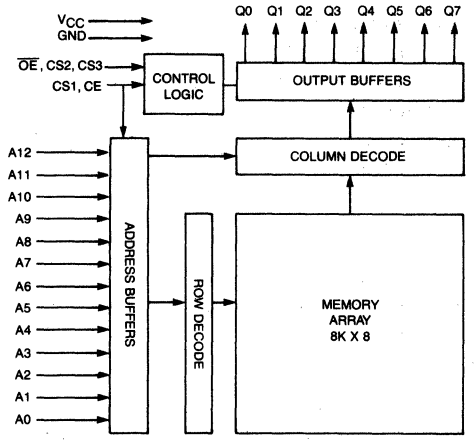
MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground ..... −0.5 to +7V
Storage temperature ..... −65 °C to 150 °C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

FUNCTIONAL BLOCK DIAGRAM



RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	2364/65			2364I/65I			2364HR/65HR			Units
		Min.	Typ.	Max.	Min.	Typ.	Max	Min.	Typ.	Max	
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	2.2		V_{CC}	Volts
V_{IL}	Input low level voltage	-0.5		0.8	-0.5		0.8	-0.5		0.8	Volts
T_A	Operating ambient temperature	0		70	-40		85	-55		125	°C

* V_{CC} must be applied at least 100μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{IN}	Input leakage current	V_{IN}= 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max, Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = −200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{cc}	Supply current - active	Outputs Open			60	mA
I_{SB}*	Supply current - standby	Chip Deselected			10	mA

* Applies to 2364S/65S power down versions only

CAPACITANCE,[1] T_A = 25 °C, f = 1 MHz

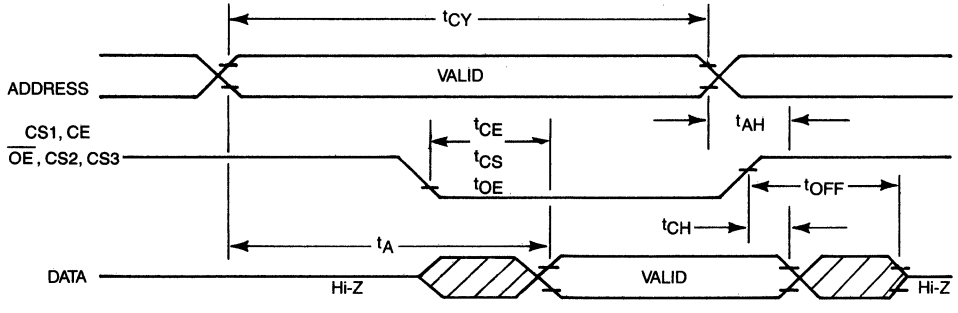
Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
C_{IN}	Input capacitance	All pins except pin under			7	pF
C_O	Output capacitance	test are tied to ground			12.5	pF

Note: 1. Characteristics are the same for all operating temperature ranges.

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

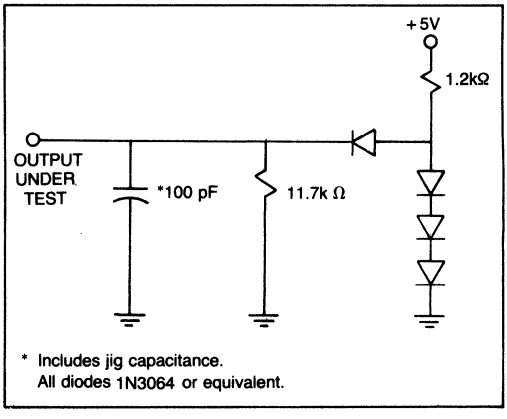
Symbol	Parameter	2364/65-20, 2364I/65I-20. 2364HR/65HR-20			2364/65-25, 2364I/65I-25, 2364HR/65HR-25			2364/65-30, 2364I/65I-30, 2364HR/65HR-30			Units
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
t_{CY}	Cycle time	200			250			300			ns
t_A	Address access time			200			250			300	ns
t_{CE}	Chip enable access time	200			250			300			ns
t_{CS}	Chip select access time			100			120			150	ns
t_{OFF}	Chip select to data off (Hi Z)			100			120			150	ns
t_{CH}	Data hold time from control	0			0			0			ns
t_{AH}	Data hold time from address	0			0			0			ns

AC WAVEFORMS



The chip select line is assumed to be active low.

AC TEST LOAD CIRCUIT



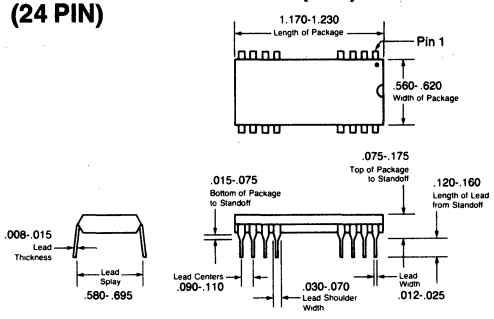
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

A.C. CONDITION OF TESTS

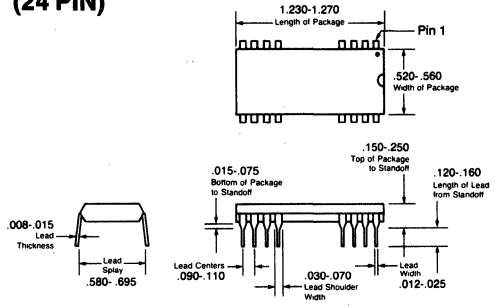
Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times ..... 10 ns
Output Timing Levels..... 0.8 Volts to 2.0 Volts

MECHANICAL DATA

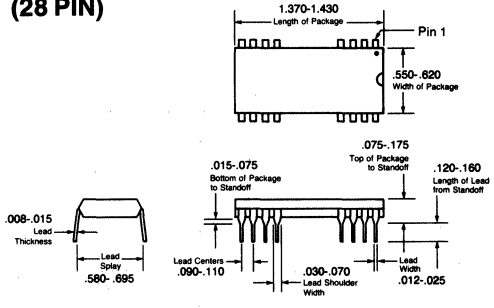
CERAMIC DUAL IN LINE (DIP) (24 PIN)



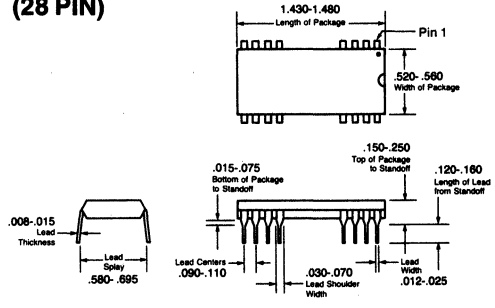
PLASTIC DUAL IN LINE (DIP) (24 PIN)



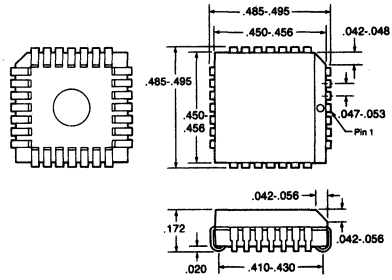
CERAMIC DUAL IN LINE (DIP) (28 PIN)



PLASTIC DUAL IN LINE (DIP) (28 PIN)



PLASTIC LEADED CHIP CARRIER (PLCC) (28 PIN)



All dimensions are in inches.

128K (16K X 8) ROM

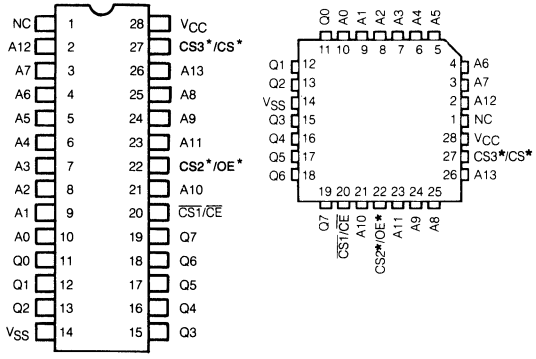
23128

- **23128..... Non-Power Down**
- **23128S ..... Automatic Power Down**
- **Fully Static Operation**
- **Silicon Gate NMOS Technology**
- **Maximum Access Times**
  - **23128/23128S-15 ..... 150ns**
  - **23128/23128S-20 ..... 200ns**
  - **23128/23128S-25 ..... 250ns**
- **Fully TTL Compatible**
- **5 Volt Only Operation**
- **Byte-Wide Industry Standard JEDEC Pin-Out**
- **Available in Three Temperature Ranges**
  - **23128 (Commercial)..... 0°C to 70°C**
  - **23128I (Industrial) ..... −40°C to 85°C**
  - **23128HR (Military) ..... −55°C to 125°C**

The NCR 23128 is a mask programmable read-only-memory with a 16K word by 8 bit organization. Designed for ease of use, this device requires only a 5 volt supply, is TTL compatible, and because of its totally static (asynchronous) operation, requires no clock. This memory device is available in two versions. The NCR 23128 is a non-power down version where the active level of chip selects CS2 and CS3 is programmable and is defined by the user to facilitate system memory expansion. The NCR 23128S offers an automatic power down feature (standby) controlled by the chip enable $\overline{CE}$ input. When $\overline{CE}$ goes high, the device automatically powers down and remains in a low power standby mode as long as $\overline{CE}$ remains high. Also, on the 23128S, the active level of chip select CS and output enable OE is programmable, thereby providing easier system implementation. The NCR 23128 is packaged in a 28 pin DIP with an industry standard byte-wide JEDEC pin-out. Optionally, this device is available in a space saving 28 pin surface mounted plastic leaded chip carrier.

PIN CONFIGURATION

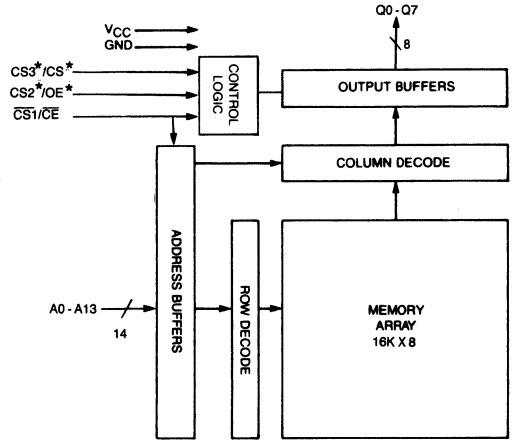
NCR 23128/S



* Programmable Active High or Low

NOTE:
23128 Only: Chip Selects $\overline{CS1}$, CS2 and CS3 are normally AND'd, ie, ($\overline{CS1}$•CS2•CS3). At the option of the user, $\overline{CS1}$ and CS2 may be internally OR'd and then AND'd with CS3 ie, [($\overline{CS1}$ + CS2)•CS3]

FUNCTIONAL BLOCK DIAGRAM



PIN NAMES

A_0 - A_{13}	Address Inputs	$\overline{CE}$	Chip Enable
Q_0 - Q_7	Data Outputs	OE	Output Enable
$\overline{CS1}$, CS2, CS3, CS	Chip Selects	V_{CC}	5 Volt ± 10% Power Supply
NC	No Connection	V_{SS}	Ground

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground..... −0.5 to +7V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

		23128			23128I			23128HR			
Symbol	Parameter	Min.	Typ.	Max.	Min.	Typ.	Max	Min.	Typ.	Max	Units
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	2.2		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.5		0.8	−0.5		0.8	−0.5		0.8	Volts
T_A	Operating ambient temperature	0		70	−40		85	−55		125	°C

* V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{IN}	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max, Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = −200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}	Supply current — Active	Outputs Open			75	mA
I_{SB}*	Supply current — Standby	Chip Deselected			10	mA

*Applies to 23128S Power Down Version only.

CAPACITANCE,[1] T_A = 25°C, f = 1 MHZ

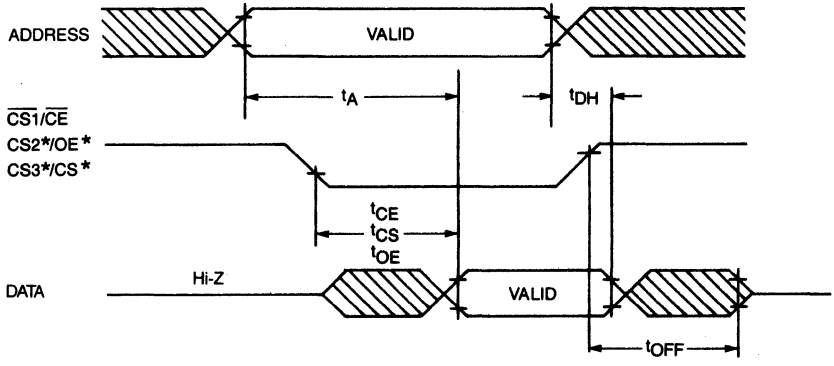
Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input capacitance	All pins except pin under test are tied to ground			7.0	pF
C_O	Output capacitance				12.5	pF

Note: 1. Characteristics are the same for all Operating Temperature Ranges.

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	23128-15, 23128I-15		23128-20, 23128I-20		23128-25, 23128I-25, 23128HR-25		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_A	Address access time		150		200		250	ns
t_{CE}	Chip enable access time		150		200		250	ns
t_{CS}	Chip select access time		75		100		120	ns
t_{OE}	Output enable access time		75		100		120	ns
t_{DH}	Data hold time	0		0		0		ns
t_{OFF}	CS active to data high Impedance		75		100		120	ns

AC WAVEFORMS

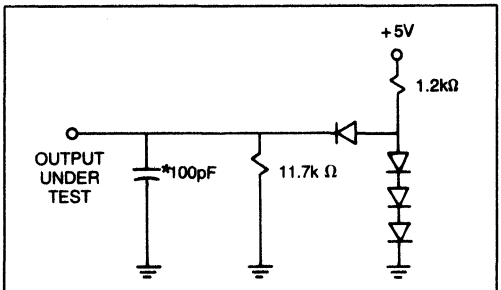


*Programmable Chip Selects CS, CS2 and CS3 and Output Enable OE are assumed to be active low for this example.

AC CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times ..... 10 ns
Output Timing Levels..... 0.8 Volts to 2.0 Volts

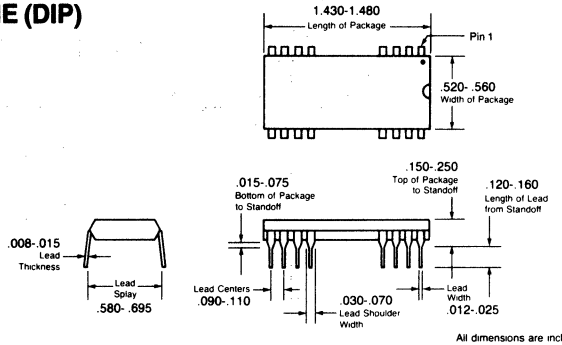
AC TEST LOAD CIRCUIT



* Includes jig capacitance.
All diodes 1N3064 or equivalent.

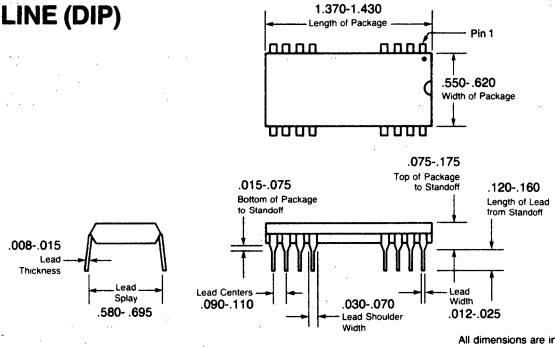
MECHANICAL DATA 28 PIN

PLASTIC DUAL IN LINE (DIP)



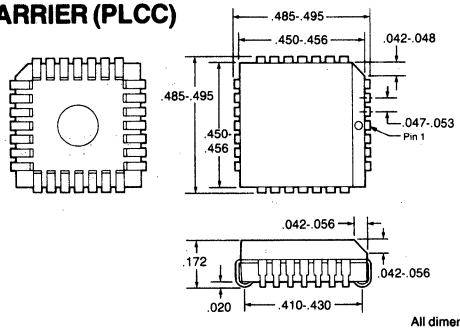
All dimensions are inches.

CERAMIC DUAL IN LINE (DIP)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (PLCC)



All dimensions are inches.

256K (32K X 8) ROM

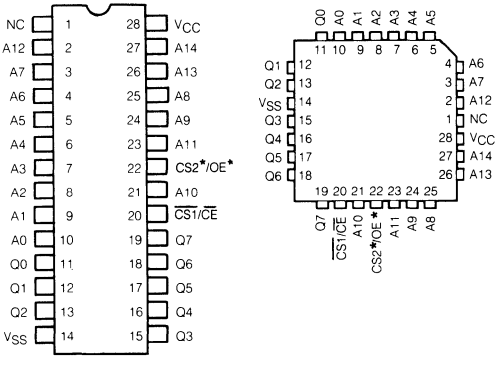
23256/57

- **23256/57 ..... Non-Power Down**
- **23256S/57S ..... Automatic Power Down**
- **Fully Static Operation**
- **Silicon Gate NMOS Technology**
- **Maximum Access Time**
  - **23256/57-15 ..... 150ns**
  - **23256/57-20 ..... 200ns**
  - **23256/57-25 ..... 250ns**
- **Pin compatible with 27256 EPROMS**
- **Fully TTL Compatible**
- **5 Volt Only Operation**
- **Byte-Wide Industry Standard JEDEC Pin-Out**
- **Available in three temperature ranges**
  - **23256/57 (Commercial) ..... 0°C to 70°C**
  - **23256I/57I (Industrial) ..... −40°C to 85°C**
  - **23256HR/57HR (Military) ..... −55°C to 125°C**

The NCR 23256 and 23257 are mask programmable read-only-memories with 32K word by 8 bit organizations. Designed for ease of use, these devices require only a 5 volt supply, are TTL compatible, and because of their totally static (asynchronous) operation, require no clock. These memory devices are available in two versions. The NCR 23256 and 23257 are non-power down versions where the active level of chip select CS2 is programmable and is defined by the user to facilitate system memory expansion. The NCR 23256S and 23257S are standby versions offering an automatic power down feature controlled by the chip enable CE input. When CE goes high, the device automatically powers down and remains in a low power standby mode as long as CE remains high. Also, on the 23256S and 23257S, the active level of output enable OE is programmable, thereby providing easier system implementation. The NCR 23256 is packaged in a 28 pin DIP with an industry standard byte-wide JEDEC pin-out. The 23257 is packaged in a 28 pin DIP with an alternate pin-out where pin 1 = A14 and pin 27 = NC. Optionally these devices are available in a space saving 28 pin surface mounted plastic leaded chip carrier.

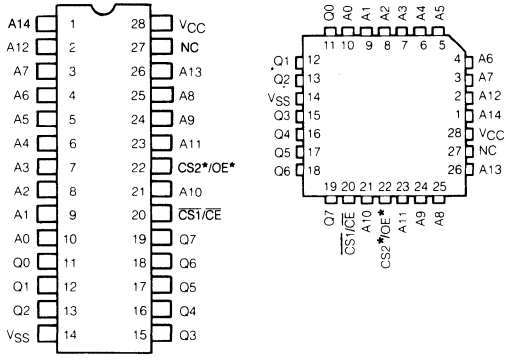
PIN CONFIGURATION

NCR 23256/S



*Programmable Active High, Low or Don't Care

NCR 23257/S



*Programmable Active High, Low or Don't Care

NOTE:
23256/57 Only. Chip Selects $\overline{CS1}$ and CS2 are normally AND'd, but may be internally OR'd at the option of the user.

PIN NAMES

A_0 - A_{14}	Address Inputs	OE	Output Enable
Q_0 - Q_7	Data Outputs	$\overline{CE}$	Chip Enable
$\overline{CS1}$, CS2	Chip Selects	V_{CC}	5 Volt ± 10% Power Supply
NC	No Connection		

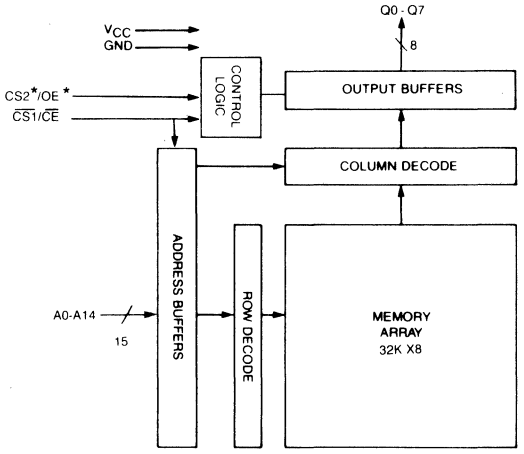
MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground ..... −0.5 to +7V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

FUNCTIONAL BLOCK DIAGRAM



RECOMMENDED OPERATING CONDITIONS

		23256/57			23256I/57I			23256HR/57HR			
Symbol	**Parameter**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max**	**Min.**	**Typ.**	**Max**	**Units**
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	2.2		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.5		0.8	−0.5		0.8	−0.5		0.8	Volts
T_A	Operating ambient temperature	0		70	−40		85	−55		125	°C

*V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Units**
I_{IN}	Input leakage current	V_{IN} = 0V to V_{CC} max,			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max, Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = −200 μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}	Supply current — Active	Outputs Open			75	mA
I_{SB}*	Supply current — Standby	Chip Deselected			10	mA

*Applies to 23256S/57S Power Down Versions Only

CAPACITANCE[1], T_A= 25°C, f = 1 MHZ

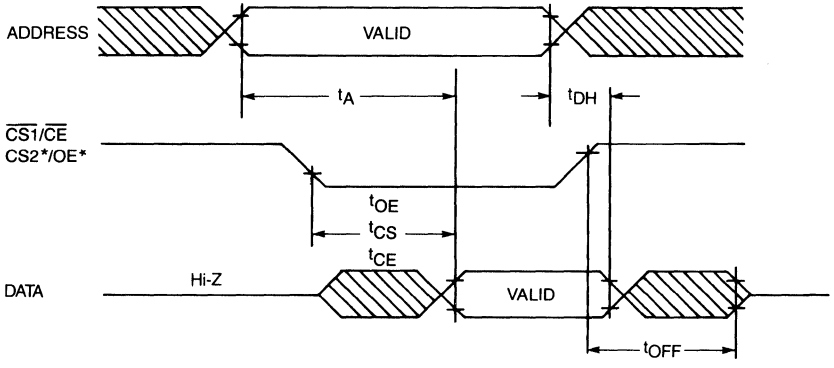
Symbol	**Parameter**	**Condition**	**Min**	**Typ**	**Max**	**Units**
C_{IN}	Input capacitance	All pins except pin under test are tied to ground			7.0	pF
C_O	Output capacitance				12.5	pF

Note: 1. Characteristics are the same for all Operating Temperature Ranges.

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS

		23256/57-15 23256I/57I-15		23256/57-20 23256I/57I-20		23256/57-25 23256I/57I-25		23256HR-27 23257HR-27		
Symbol	**Parameter**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_A	Address access time		150		200		250		275	ns
t_{CE}	Chip enable access time		150		200		250		275	ns
t_{CS}	Chip select access time		80		100		120		135	ns
t_{OE}	Output enable access time		80		100		120		135	ns
t_{DH}	Data hold time	0		0		0		0		ns
t_{OFF}	CS active to data high Impedance		80		100		120		135	ns

AC WAVEFORMS

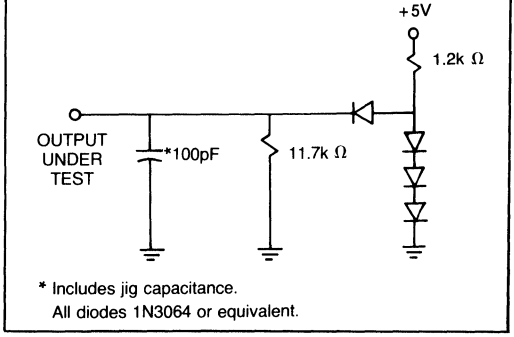


*Programmable Chip Select CS2 and Output Enable OE are assumed to be active low for this example.

AC CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise and Fall Times..... 10 ns
Output Timing Levels..... 0.8 Volts to 2.0 Volts

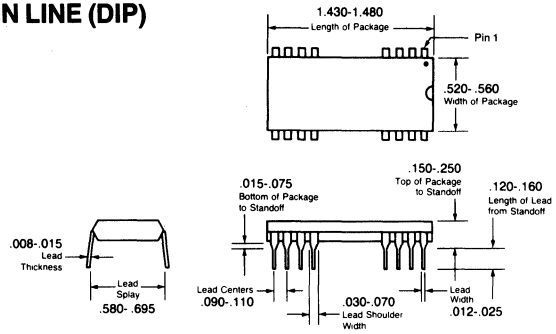
AC TEST LOAD CIRCUIT



* Includes jig capacitance.
All diodes 1N3064 or equivalent.

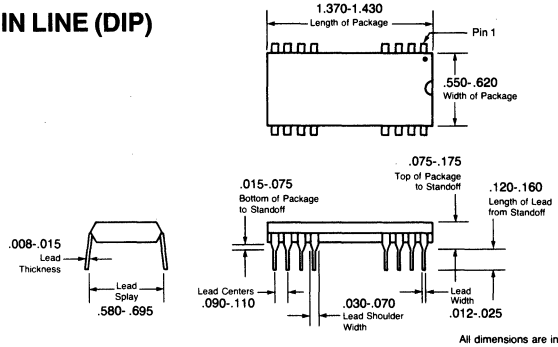
MECHANICAL DATA 28 PIN

PLASTIC DUAL IN LINE (DIP)



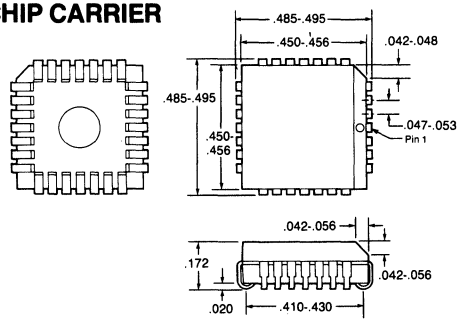
All dimensions are inches.

CERAMIC DUAL IN LINE (DIP)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (PLCC)



All dimensions are inches.

CMOS ROMS

Introduction

NCR offers CMOS ROMs ranging in sizes from 64K to 512K fabricated in a 2.75u technology. ROM patterns are accepted in EPROM. If additional information is needed contact: NCR - Fort Collins, C.O.T. Department.

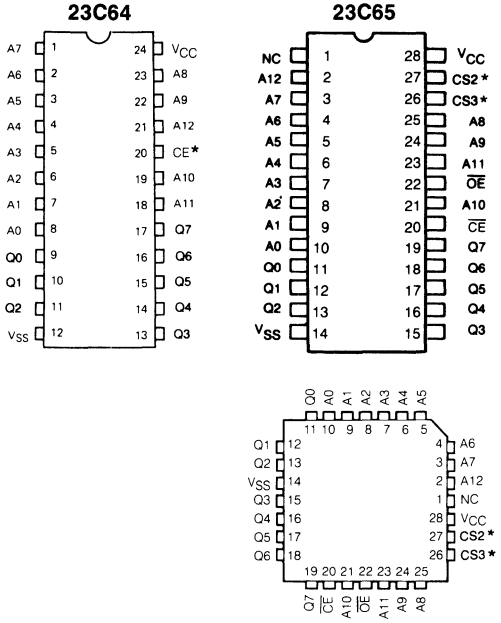
64K (8K X 8) CMOS ROM

23C64/65

- **Fully Static Operation**
- **Silicon Gate CMOS Technology**
- **Maximum Access Time**
  23C64/65-15 ..... 150ns
  23C64/65-20 ..... 200ns
  23C64/65-25 ..... 250ns
- **5 Volt Only Operation**
- **Byte-Wide Industry Standard JEDEC Pin-Out**
- **Available in Two Temperature Ranges**
  23C64/65 (Commercial) ..... 0°C to 70°C
  23C64/65I (Industrial) ..... −40°C to 85°C

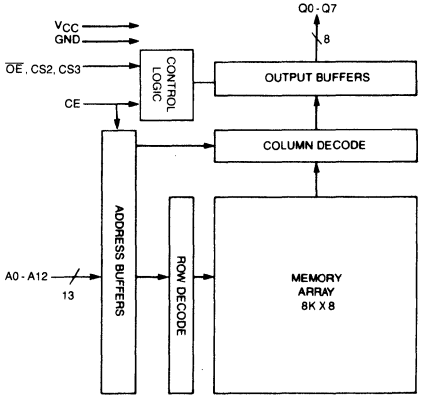
The NCR 23C64 and 23C65 are mask programmable read-only memories with 8K word by 8 bit organizations. Designed for ease of use, these devices require only a 5-volt supply, are TTL compatible, and because of their totally static (asynchronous) operation, require no clock. These CMOS ROMS offer very low power dissipation in the operational mode and have an automatic power down feature that significantly reduces power consumption in the standby mode. The NCR 23C64 is packaged in a 24-pin DIP and the 23C65 is packaged in a 28-pin DIP, both with industry standard byte-wide JEDEC pin-outs. Optionally, the 23C65 is available in a space saving 28-pin surface mounted plastic leaded chip carrier.

PIN CONFIGURATION



*Programmable High, Low or Don't Care

FUNCTIONAL BLOCK DIAGRAM



PIN NAMES

A0-A12	Address Inputs
Q0-Q7	Data Outputs
CE	Chip Enable
V_{CC}	5V @ 10% Supply Voltage
V_{SS}	Ground
CS2, CS3	Chip Selects
$\overline{OE}$	Output Enable

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground ..... −0.5 to +7V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	23C64/65			23C64I/65I			Units
		Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	-0.3		0.8	-0.3		0.8	Volts
T_A	Operating ambient temperature	0		70	-40		85	°C

*V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_N	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max Chip Deselected			± 10	μA
V_{OH}	Output high voltage	I_{OH} = -200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}[2]	Supply current—active	Outputs Open			25	mA
I_{SB}[3]	Supply current—standby	Chip Deselected			10	μA

CAPACITANCE[1], T_A = 25°C, f = 1 MHZ

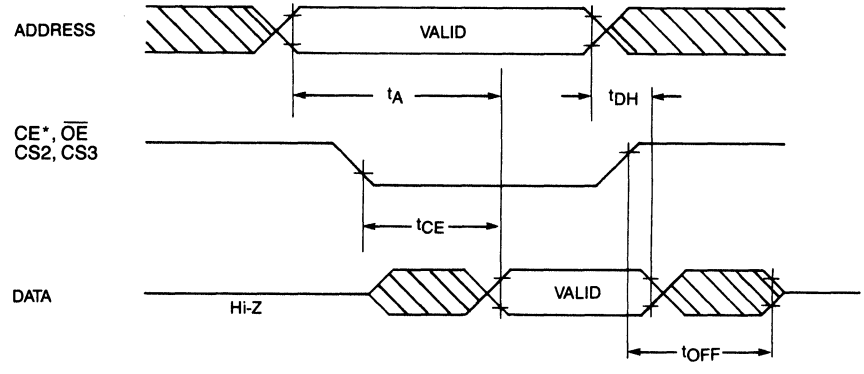
Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input capacitance	All pins except pin under test			10.0	pF
C_O	Output capacitance	are tied to ground			12.5	pF

NOTES: 1. Characteristics are the same for both Commercial and Industrial Operating Temperature Ranges.
2. Current is proportional to cycle rate. I_{CC} is measured at the specified minimum cycle time. Data outputs open.
$V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$
3. $\overline{CE} \geq V_{CC} - 0.3V$ or $CE \leq V_{SS} + 0.3V$

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1]

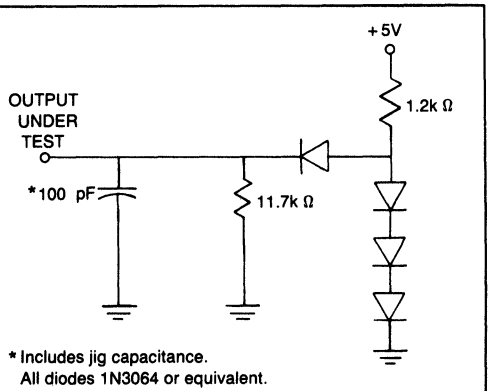
Symbol	Parameter	23C64/65-15		23C64/65-20		23C64/65-25		Units
		Min	Max	Min	Max	Min	Max	
t_A	Address Access Time		150		200		250	ns
t_{CE}	Chip Enable Access Time		150		200		250	ns
t_{DH}	Data Hold Time	0		0		0		ns
t_{OFF}	CS Active to Data High Impedance		80		80		80	ns

AC WAVEFORMS



*Programmable Chip Enable is assumed to be active low for this example.

AC TEST LOAD CIRCUIT

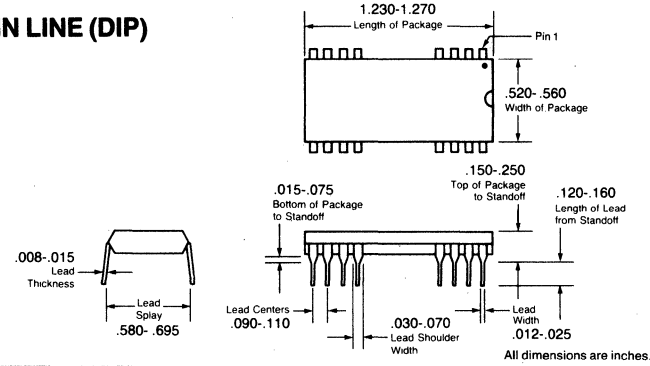


* Includes jig capacitance.
All diodes 1N3064 or equivalent.

AC CONDITIONS OF TEST

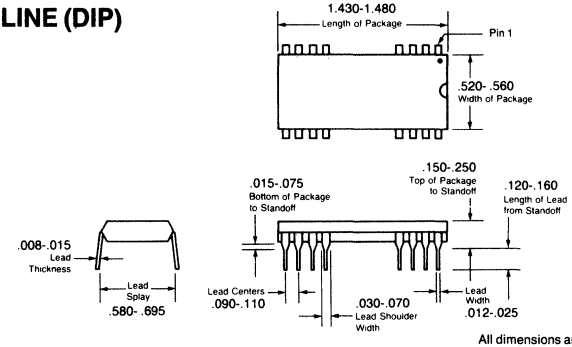
Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times ..... 10 ns
Output Timing Levels ..... 0.8 Volts to 2.0 Volts

PLASTIC DUAL IN LINE (DIP) (24 PIN)



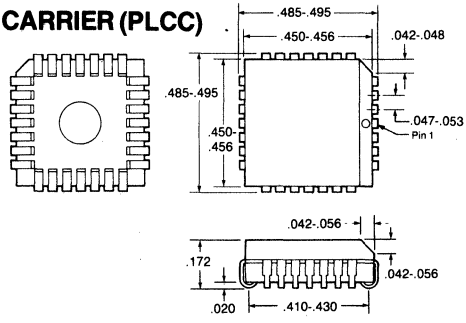
All dimensions are inches.

PLASTIC DUAL IN LINE (DIP) (28 PIN)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (PLCC)



All dimensions are inches.

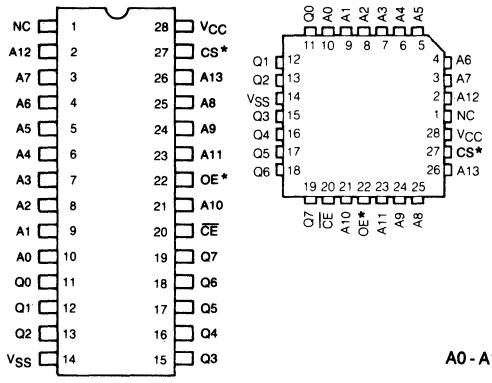
128K (16K X 8) CMOS ROM

23C128

- **Fully Static Operation**
- **Silicon Gate CMOS Technology**
- **Maximum Access Time**
  - **23C128-15 ..... 150ns**
  - **23C128-20 ..... 200ns**
  - **23C128-25 ..... 250ns**
- **Fully TTL Compatible**
- **5 Volt Only Operation**
- **Byte-Wide industry Standard JEDEC Pin-Out**
- **Available in Two Temperature Ranges**
  - **23C128 (Commercial) ..... 0° C to 70 °C**
  - **23C128I (Industrial) ..... −40 °C to 85 °C**

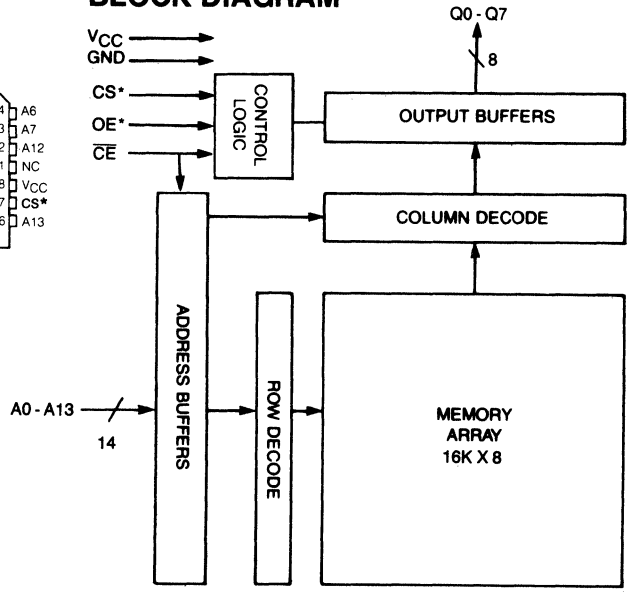
The NCR 23C128 is a mask programmable read-only-memory with a 16K word by 8 bit organization. Designed for ease of use, this device requires only a 5 volt power supply, is TTL compatible, and because of its totally static (asynchronous) operation, requires no clock. This CMOS ROM offers very low power dissipation in the operational mode and has an automatie power down feature that significantly reduces power consumption in the standby mode. The active level of output enable OE and chip select CS is programmable, thereby providing easier system implementation. The NCR 23C128 is packaged in a 28 pin DIP with an industry standard byte-wide JEDEC pin-out. Optionally, this device is available in a space saving 28 pin surface mounted plastic leaded chip carrier.

PIN CONFIGURATION



*Programmable Output Enable and Chip Select High, Low or Don't Care

FUNCTIONAL BLOCK DIAGRAM



PIN NAMES

A0-A13	Address Inputs	$\overline{CE}$	Chip Enable
Q0-Q7	Data Outputs	V_{CC}	5V ± 10% Supply Voltage
CS	Chip Select	V_{SS}	Ground
OE	Output Enable	NC	No Connection

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground −0.5 to +7V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	23C128			23C128I			Units
		Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	-0.3		0.8	-0.3		0.8	Volts
T_A	Operating ambient temperature	0		70	-40		85	°C

*V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_N	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = -200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}[2]	Supply current—active	Outputs Open			35	mA
I_{SB}[3]	Supply current—standby	Chip Deselected			40	μA

CAPACITANCE[1]; T_A = 25°C, f = 1 MHZ

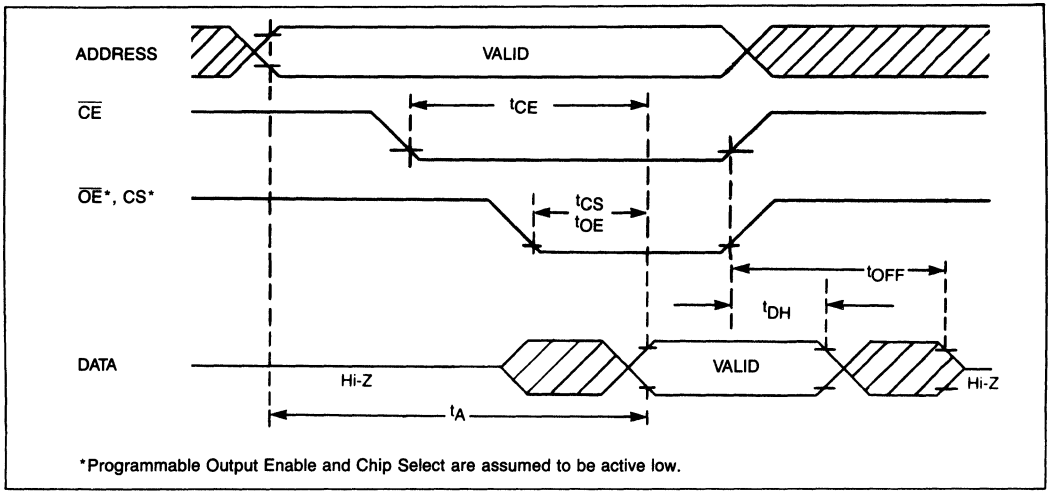
Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input capacitance	All pins except pin under test			10.0	pF
C_O	Output capacitance	are tied to ground			12.5	pF

NOTES: 1. Characteristics are the same for both Commercial and Industrial Operating Temperature Ranges.
2. Current is proportional to cycle rate. I_{CC} is measured at the specified minimum cycle time. Data outputs open.
$V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$
3. $\overline{CE} \geq V_{CC} - 0.3V$ or $CE \leq V_{SS} + 0.3V$

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1]

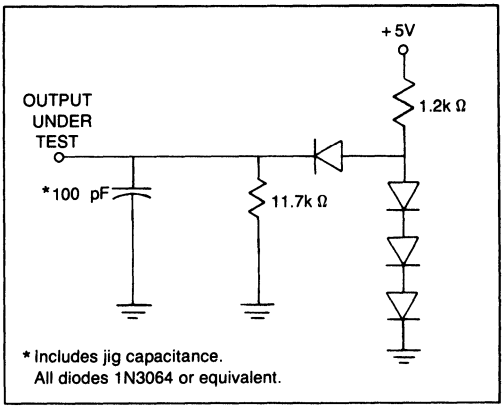
Symbol	Parameter	23C128-15		23C128-20		23C128-25		Units
		Min	Max	Min	Max	Min	Max	
t_A	Address Access Time		150		200		250	ns
t_{CE}	Chip Enable Access Time		150		200		250	ns
t_{CS}	Chip Select Access Time		75		100		125	ns
t_{OE}	Output Enable Access Time		75		100		125	ns
t_{DH}	Data Hold Time	0		0		0		ns
t_{OFF}	CS Active to Date High Impedance		80		80		80	ns

AC WAVEFORMS



*Programmable Output Enable and Chip Select are assumed to be active low.

AC TEST LOAD CIRCUIT



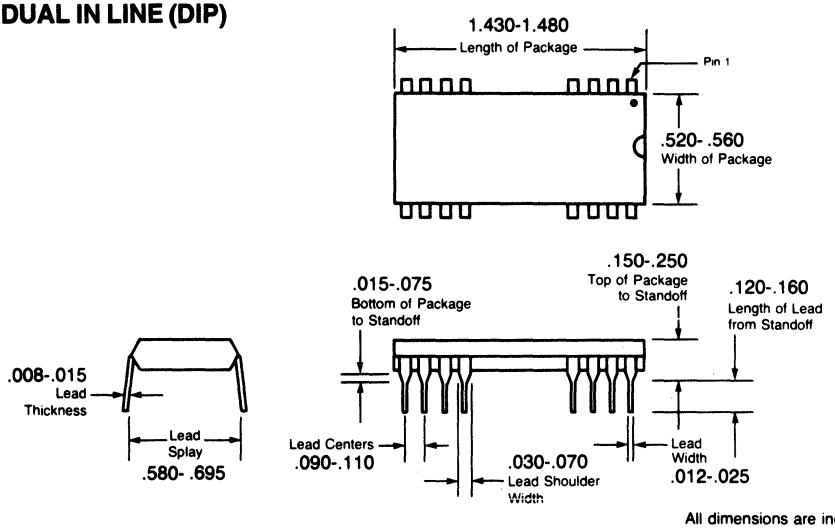
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

AC CONDITIONS OF TEST

Input Pulse Levels..... 0.8 volts to 2.0 volts
Inputs Rise & Fall Times..... 10 ns
Output Timing Levels..... 0.8 Volts to 2.0 Volts

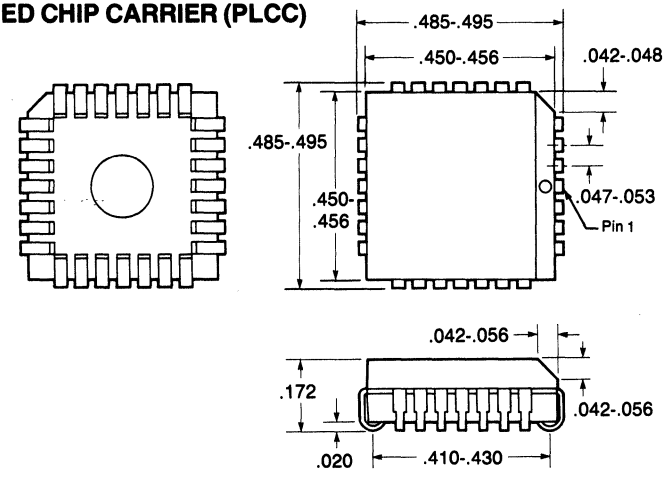
MECHANICAL DATA 28 Pin

PLASTIC DUAL IN LINE (DIP)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (PLCC)



All dimensions are inches.

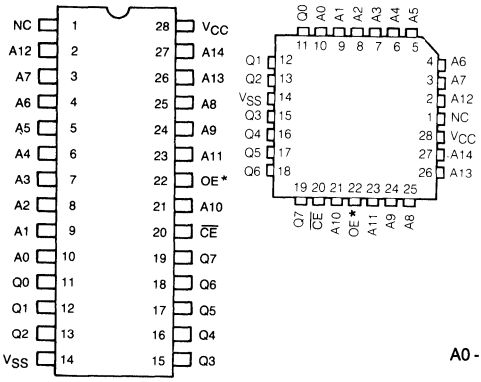
256K (32K X 8) CMOS ROM

23C256

- **Fully Static Operation**
- **Silicon Gate CMOS Technology**
- **Maximum Access Time**
  - **23C256-15 ..... 150ns**
  - **23C256-20 ..... 200ns**
  - **23C256-25 ..... 250ns**
- **Fully TTL Compatible**
- **5 Volt Only Operation**
- **Byte-Wide Industry Standard JEDEC Pin-Out**
- **Available in Two Temperature Ranges**
  - **23C256 (Commercial) ..... 0°C to 70°C**
  - **23C256I (Industrial) ..... −40°C to 85°C**

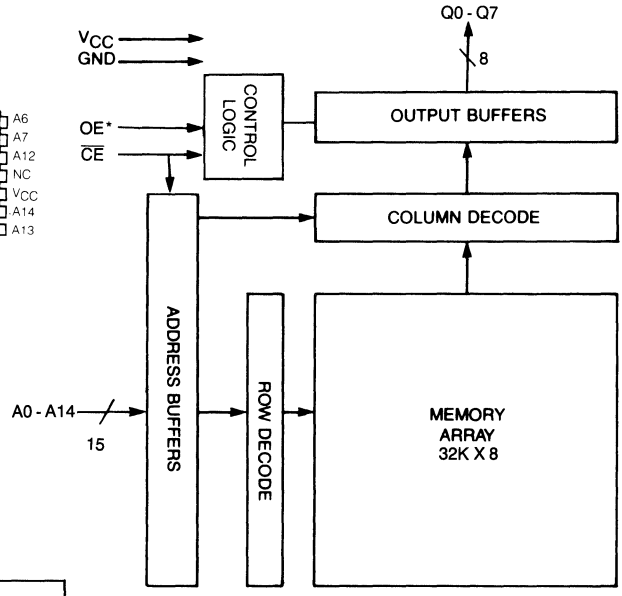
The NCR 23C256 is a mask programmable read-only-memory with a 32K word by 8 bit organization. Designed for ease of use, this device requires only a 5 volt power supply, is TTL compatible, and because of its totally static (asynchronous) operation, requires no clock. This CMOS ROM offers very low power dissipation in the operational mode and has an automatic power down feature that significantly reduces power consumption in the standby mode. The active level of output enable OE is programmable, thereby providing easier system implementation. The NCR 23C256 is packaged in a 28 pin DIP with an industry standard byte-wide JEDEC pin-out. Optionally, this device is available in a space saving 28 pin surface mounted plastic leaded chip carrier.

PIN CONFIGURATION



* Programmable Output Enable High, Low or Don't Care.

FUNCTIONAL BLOCK DIAGRAM



PIN NAMES

A0-A14	Address Inputs
Q0-Q7	Data Outputs
OE	Output Enable
$\overline{CE}$	Chip Enable
V_{CC}	5V ± 10% Supply Voltage
V_{SS}	Ground

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground ..... −0.5 to +7V
Storage temperature ..... −65°C to 150°C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	23C256			23C256I			Units
		Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.3		0.8	−0.3		0.8	Volts
T_A	Operating ambient temperature	0		70	−40		85	°C

*V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_N	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = −200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}[2]	Supply current—active	Outputs Open			35	mA
I_{SB}[3]	Supply current—standby	Chip Deselected			40	μA

CAPACITANCE[1], T_A = 25°C, f = 1 MHZ

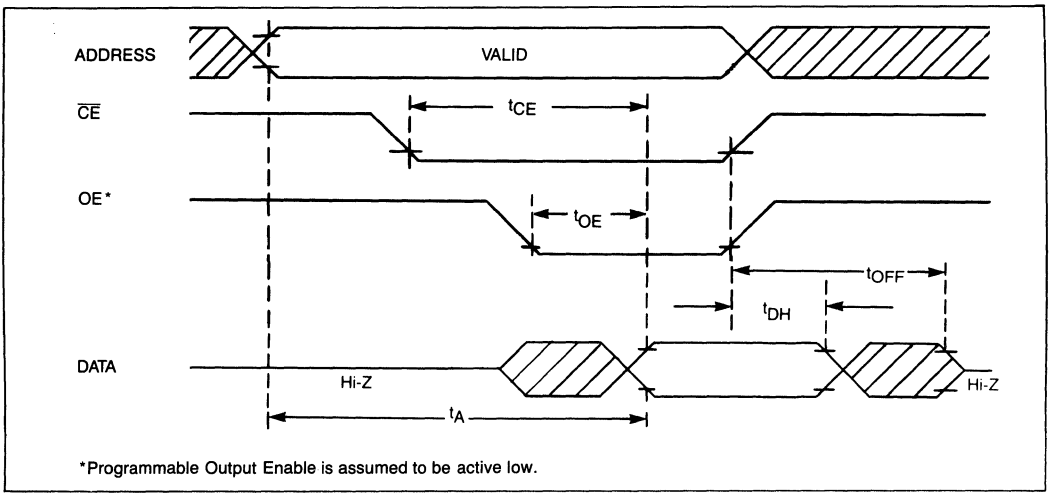
Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input capacitance	All pins except pin under test			10.0	pF
C_O	Output capacitance	are tied to ground			12.5	pF

NOTES: 1. Characteristics are the same for both Commercial and Industrial Operating Temperature Ranges.
2. Current is proportional to cycle rate. I_{CC} is measured at the specified minimum cycle time. Data outputs open.
$V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$
3. $\overline{CE} \geq V_{CC} - 0.3V$ or $CE \leq V_{SS} + 0.3V$

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1]

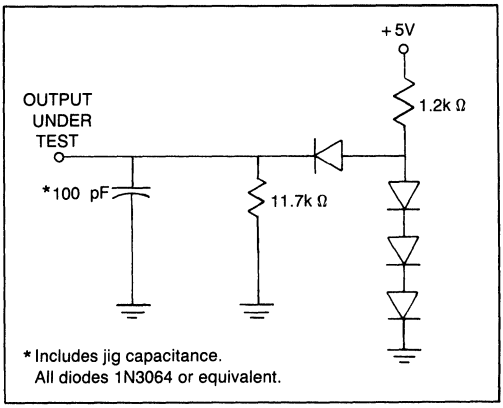
Symbol	Parameter	23C256-15		23C256-20		23C256-25		Units
		Min	Max	Min	Max	Min	Max	
t_A	Address Access Time		150		200		250	ns
t_{CE}	Chip Enable Access Time		150		200		250	ns
t_{OE}	Output Enable Access Time		75		100		125	ns
t_{DH}	Data Hold Time	0		0		0		ns
t_{OFF}	CS Active to Date High Impedance		80		80		80	ns

AC WAVEFORMS



*Programmable Output Enable is assumed to be active low.

AC TEST LOAD CIRCUIT



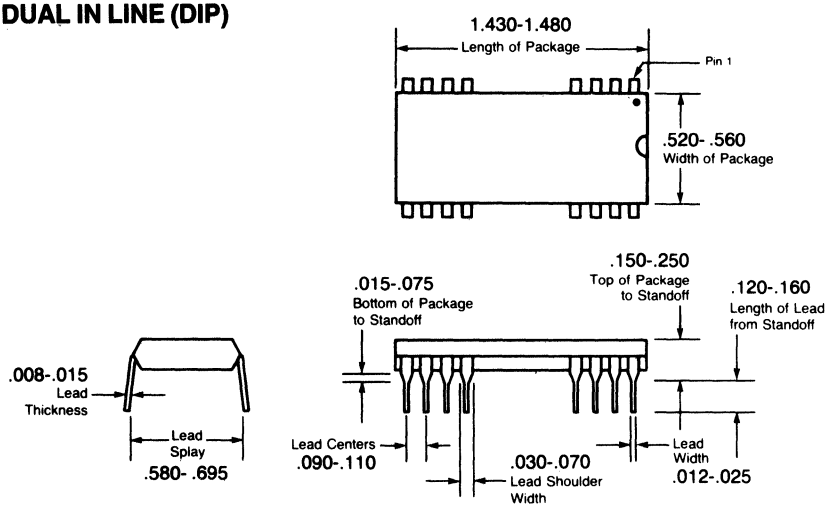
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

AC CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volts to 2.0 Volts
Inputs Rise & Fall Times ..... 10 ns
Output Timing Levels ..... 0.8 Volts to 2.0 Volts

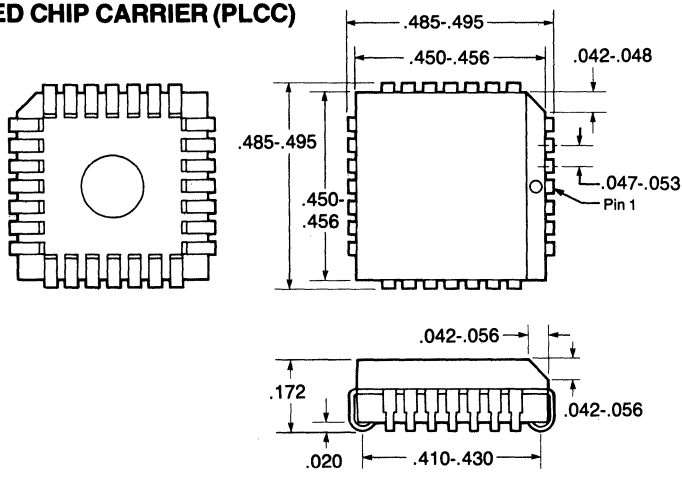
MECHANICAL DATA 28 Pin

PLASTIC DUAL IN LINE (DIP)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (PLCC)



All dimensions are inches.

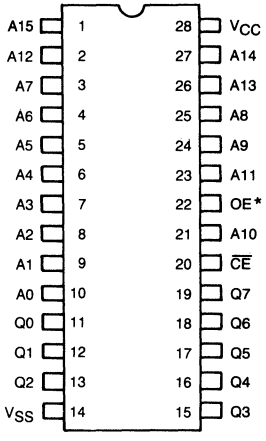
512K (64K X 8) CMOS ROM

23C512

- **Fully Static Operation**
- **Silicon Gate CMOS Technology**
- **Maximum Access Time**
  23C512-15 ..... 150 ns
  23C512-20 ..... 200 ns
  23C512-25 ..... 250 ns
- **Fully TTL Compatible**
- **5 Volt Only Operation**
- **Byte-Wide Industry Standard JEDEC Pin-Out**
- **Available in Two Temperature Ranges**
  23C512 (Commercial) ..... 0 °C to 70 °C
  23C512I (Industrial) ..... −40 °C to 85 °C

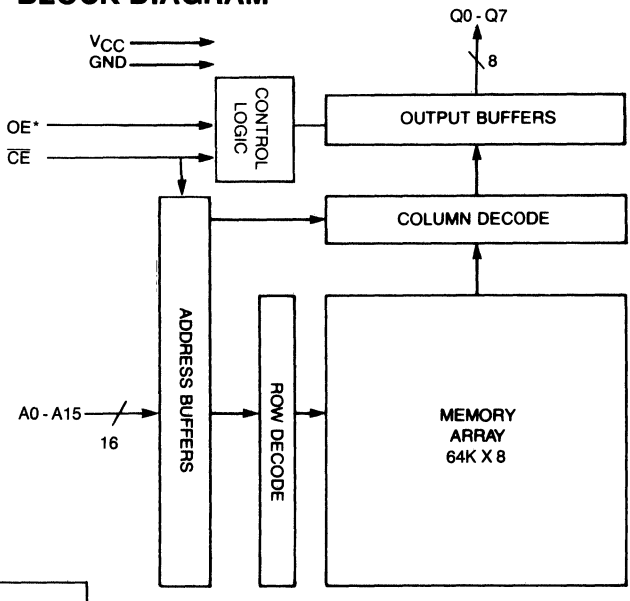
The NCR 23C512 is a mask programmable read-only-memory with a 64K word by 8 bit organization. Designed for ease of use, this device requires only a 5 volt power supply, is TTL compatible, and because of its totally static (asynchronous) operation, requires no clock. This CMOS ROM offers very low power dissipation in the operational mode and has an automatic power down feature that significantly reduces power consumption in the standby mode. The active level of output enable OE is programmable, thereby providing easier system implementation. The NCR 23C512 is packaged in a 28 pin DIP with an industry standard byte-wide JEDEC pin-out. Optionally, this device is available in a space saving 32 pin surface mounted plastic leaded chip carrier.

PIN CONFIGURATION



*Programmable Output Enable High, Low or Don't Care

FUNCTIONAL BLOCK DIAGRAM



PIN NAMES

A0-A15	Address Inputs
Q0-Q7	Data Outputs
OE	Output Enable
$\overline{CE}$	Chip Enable
VCC	5V ± 10% Supply Voltage
VSS	Ground

MEMORY PRODUCTS

ABSOLUTE MAXIMUM RATINGS

Voltage on any pin with respect to ground ..... −0.5 to +7V
Storage temperature ..... −65 °C to 150 °C

Stresses above "absolute maximum ratings" may result in damage to the device. Functional operation of devices at the "absolute maximum ratings" or above the recommended operating conditions stipulated elsewhere in this specification is not implied.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	23C512			23C512I			Units
		Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply voltage*	4.5	5.0	5.5	4.5	5.0	5.5	Volts
V_{IH}	Input high level voltage	2.0		V_{CC}	2.0		V_{CC}	Volts
V_{IL}	Input low level voltage	−0.3		0.8	−0.3		0.8	Volts
T_A	Operating ambient temperature	0		70	−40		85	°C

*V_{CC} must be applied at least 100 μs before proper device operation is achieved.

STATIC ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1] (UNLESS OTHERWISE NOTED)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_N	Input leakage current	V_{IN} = 0V to V_{CC} max			10	μA
I_O	Output leakage current	V_O = 0.2 to V_{CC} max Chip Deselected			±10	μA
V_{OH}	Output high voltage	I_{OH} = −200μA	2.4			Volts
V_{OL}	Output low voltage	I_{OL} = 3.2mA			0.4	Volts
I_{CC}[2]	Supply current—active	Outputs Open			40	mA
I_{SB}[3]	Supply current—standby	Chip Deselected			40	μA

CAPACITANCE[1], T_A = 25 °C, f = 1 MHZ

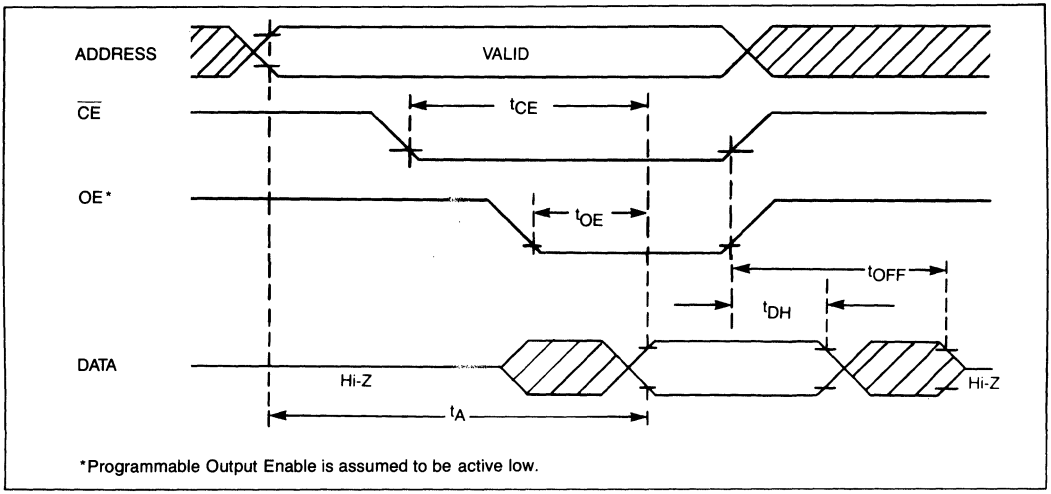
Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input capacitance	All pins except pin under test			10.0	pF
C_O	Output capacitance	are tied to ground			12.5	pF

NOTES: 1. Characteristics are the same for both Commercial and Industrial Operating Temperature Ranges.
2. Current is proportional to cycle rate. I_{CC} is measured at the specified minimum cycle time. Data outputs open.
$V_{IL} = V_{SS}$, $V_{IH} = V_{CC}$
3. $\overline{CE} \geq V_{CC} - 0.3V$ or $CE \leq V_{SS} + 0.3V$

AC CHARACTERISTICS OVER RECOMMENDED OPERATING CONDITIONS[1]

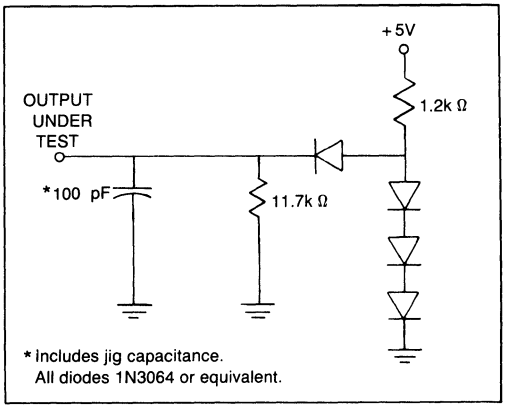
Symbol	Parameter	23C512-15		23C512-20		23C512-25		Units
		Min	Max	Min	Max	Min	Max	
t_A	Address Access Time		150		200		250	ns
t_{CE}	Chip Enable Access Time		150		200		250	ns
t_{OE}	Output Enable Access Time		75		100		125	ns
t_{DH}	Data Hold Time	0		0		0		ns
t_{OFF}	CS Active to Date High Impedance		80		80		80	ns

AC WAVEFORMS



*Programmable Output Enable is assumed to be active low.

AC TEST LOAD CIRCUIT



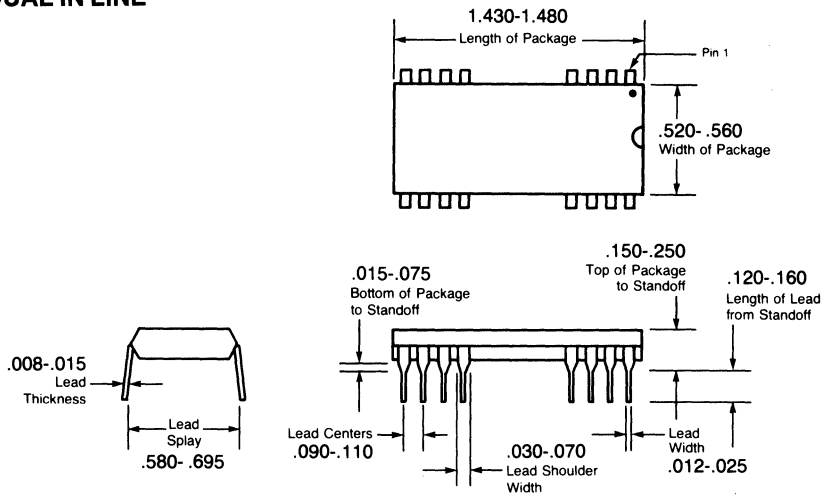
* Includes jig capacitance.
All diodes 1N3064 or equivalent.

AC CONDITIONS OF TEST

Input Pulse Levels . . . . . 0.8 volts to 2.0 volts
Inputs Rise & Fall Times . . . . . 10 ns
Output Timing Levels . . . . . 0.8 volts to 2.0 volts

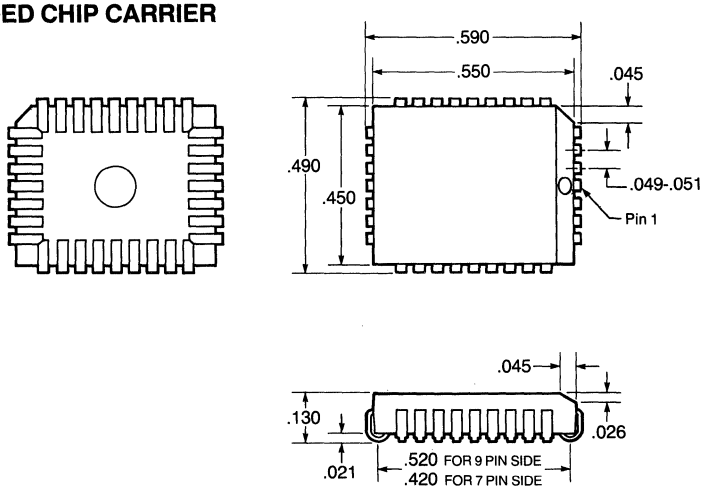
MECHANICAL DATA

PLASTIC DUAL IN LINE (28 Pin)



All dimensions are inches.

PLASTIC LEADED CHIP CARRIER (32 Pin)



All dimensions are in inches with ±.005 tolerance.

MILITARY PRODUCTS

MILITARY PRODUCTS REFERENCE

NCR's full line of memory products is available in military temperature ranges. Information regarding these products can be found in the Memory Products section in this data book.

In addition to NCR memory products, our military business unit plans to offer the 53C80, 53C81 and 53C90 SCSI products screened to military specifications. Commercial versions of these products are included in the SCSI Products section in this data book.

For further information regarding NCR military products, please call or write the following source.

NCR Military Products
1635 Aeroplaza Drive
Colorado Springs, Colorado 80916
1-800-543-5618