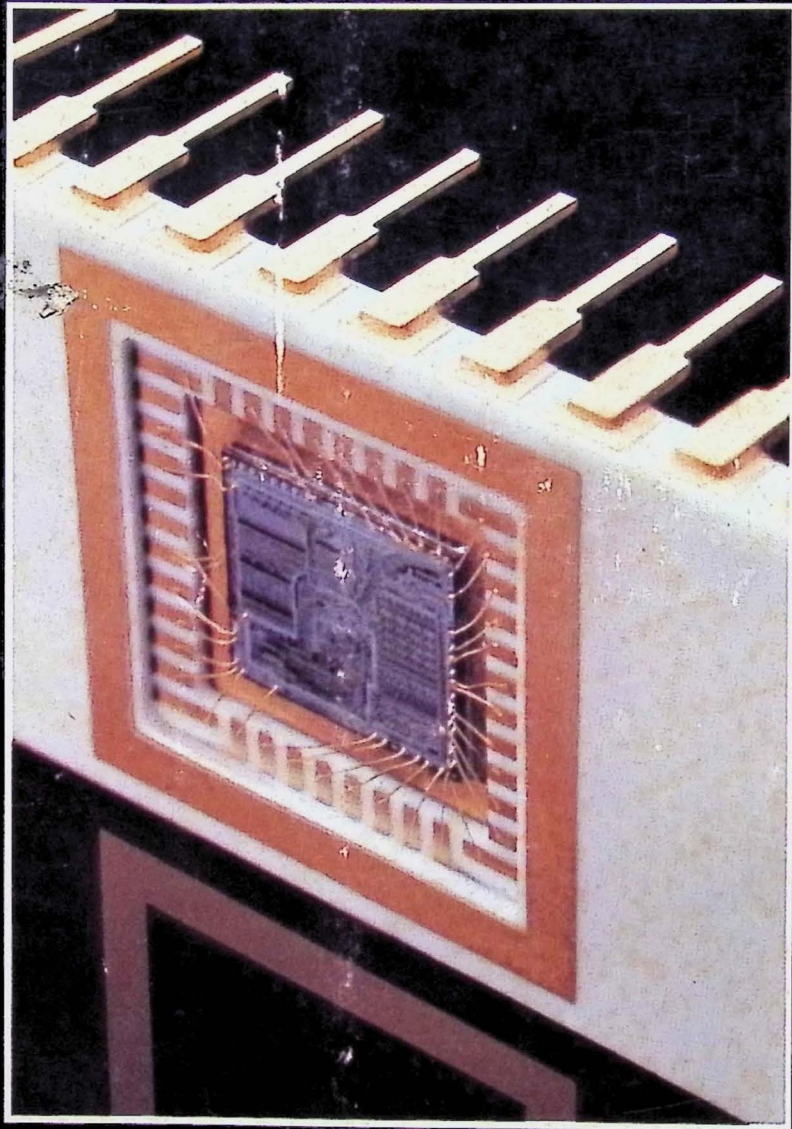


microNOVA  
INTEGRATED CIRCUITS  
DATA MANUAL



 Data General

Data General Corporation, Westboro, Massachusetts 01581

# **microNOVA INTEGRATED CIRCUITS DATA MANUAL**

**FOR  
DESIGN ENGINEERS**

**SECOND EDITION**

## **NOTICE**

This manual has been prepared by Data General Corporation (DGC) to serve as a convenient compilation of integrated circuits made by DGC. DGC reserves the right to make changes at any time in order to improve circuit design and method of presentation herein. DGC does not assume any responsibility for its circuits or applications which may be made thereof, and does not guarantee that these circuits or applications thereof are free from patent infringement. DGC shall not be responsible for any damages (including consequential) caused by use of or reliance upon materials presented herein.

**NOVA** and **ECLIPSE** are registered trademarks of Data General Corporation, Westboro, Massachusetts. **DASHER** is a trademark of Data General Corporation, Westboro, Massachusetts.

Ordering No. 014-000074  
© Data General Corporation, 1977  
All Rights Reserved  
Printed in the United States of America  
Rev. 03, November 1977

# DATA GENERAL DOMESTIC OFFICES

## SALES AND SERVICE

### ALABAMA

90 Bagby Drive  
Birmingham, Alabama 35209  
(205) 942-3730

### ARIZONA

1950 E. Watkins  
Phoenix, Arizona 85034  
(602) 254-6412

4633 E. Broadway  
Suite 100  
Tucson, Arizona 85100

### CALIFORNIA

2221 Rosecrans Avenue  
Suite 204  
El Segundo, California 90245  
(213) 973-0401

2445 Faber Place  
Palo Alto, California 94303  
(415) 321-8010

15114 Downey Avenue  
Paramount, California 90723  
(213) 633-0805

4031 No. Freeway Blvd.  
Sacramento, California 95834  
(916) 920-3434

4305 Gesner Street  
San Diego, California 92117  
(714) 276-8450

120 Howard Street  
San Francisco, California 94105  
(415) 543-6730

1220 Village Way  
Suite A  
Santa Ana, California 92705  
(714) 558-9421

5276 Hollister Avenue  
Suite E  
Santa Barbara, California 93111  
(805) 964-8695

7011 Hayvenhurst Avenue  
Suite A  
Van Nuys, California 91406  
(213) 787-8870

### COLORADO

Denver Technological Center  
Building No. 29  
8455 E. Prentice Avenue  
Englewood, Colorado 80110  
(303) 773-2620

### CONNECTICUT

50 Shaw Road  
North Branford, Connecticut 06471  
(203) 484-2771

### FLORIDA

1001 Northwest 62nd Street  
Suite 404  
Ft. Lauderdale, Florida 33309  
(305) 771-0784

6880 Lake Ellenor Drive  
Suite 115  
Orlando, Florida 32809  
(305) 851-8230

5444 Bay Center Drive  
Suite 707  
Tampa, Florida 33605  
(813) 879-2515

### GEORGIA

1639 Tullie Circle  
Suite 128  
Atlanta, Georgia 30329  
(404) 325-3181

### IDAHO

130 South Cole Road  
Boise, Idaho 83075  
(208) 375-6510

### ILLINOIS

2524 W. Farrelly Avenue  
S.W. of Route 150 and Glen  
Peoria, Illinois 61614  
(309) 688-2224

Chicago Office  
1111 Plaza Drive  
Schmumburg, Illinois 60195  
(312) 885-0505

### INDIANA

2421 Production Drive  
Suite 202  
Indianapolis, Indiana 46241  
(317) 248-8306

### KENTUCKY

4299 Bardstown Rd.  
Louisville, Kentucky 40218  
(502) 491-6595

### LOUISIANA

717 S. Foster Drive  
Suite 140  
Baton Rouge, Louisiana 70821  
(504) 923-0660

### MARYLAND

Washington D.C. Office  
1900 Sulphur Spring Road  
Suite 310  
Baltimore, Maryland 21227  
(301) 247-8900

### MASSACHUSETTS

810 Memorial Drive  
Cambridge, Massachusetts 02138  
(617) 661-7710

1500 Main Street  
Bay State West  
Suite 1820  
Springfield, Massachusetts 01103  
(413) 739-9880

888 Worcester Road  
Wellesley, Massachusetts 02161  
(617) 235-8171

Mechanics Bank Tower  
Suite 640  
Worcester, Massachusetts 01608  
(617) 754-3096

### MICHIGAN

Detroit Office  
1 Northland Drive Building  
Suite 386  
16000 West Eight Mile Road  
Southfield, Michigan 48075  
(313) 569-0200

### MINNESOTA

2850 Metro Drive  
Minneapolis, Minnesota 55420  
(612) 854-7727

### MISSOURI

4520 Madison Avenue  
Suite 210  
Kansas City, Missouri 64111  
(816) 531-1326  
St. Louis Office  
11854 Lakland Rd.  
Creve Coeur, Missouri 63141  
(314) 726-0811

### NEW HAMPSHIRE

491 Amherst Street  
Nashua, New Hampshire 03060  
(603) 888-8524

### NEW JERSEY

1013 Annapolis Drive  
Cherry Hill, New Jersey 08003  
30 Gates Drive  
Wayne, New Jersey 07470  
(201) 785-3910

### NEVADA

Las Vegas Office  
3355 W. Spring Mountain Rd. No. 54  
Las Vegas, Nevada 89103

### NEW MEXICO

2601 Wyoming, N.E.  
Suite A  
Albuquerque, New Mexico 87112  
(505) 294-1416

### NEW YORK

3527 Harlem Road  
Buffalo, New York 14200  
(716) 835-3710

12 Avis Drive  
Gupilli Industrial Park Office Bldg.  
Latham, New York 12110  
(518) 783-3101

150 Broad Hollow Road  
Third Floor  
Melville, New York 11746  
(516) 427-0311

L141 Meadowbrook Court  
Newfield, New York 14867  
(607) 272-9440

144 East 44th Street  
New York, New York 10017  
(212) 557-1122

625 Panorama Trail  
Rochester, New York 14825  
(716) 385-2000

104 Pickard Drive  
Syracuse, New York 13211  
(315) 455-1525

### NORTH CAROLINA

2 Fairview Plaza  
Suite 106  
5950 Fairview Road  
Charlotte, North Carolina 28210  
(810) 621-0570

405 Parkway Drive  
Greensboro, North Carolina 27401  
(919) 378-9401

### OHIO

Columbus Office  
P.O. Box 29292  
Columbus, Ohio 43229  
(614) 882-1039

1161 Lyons Road  
Building E  
Dayton, Ohio 45459  
(573) 435-1932

Cleveland Office  
26250 Euclid Avenue  
Suite 925  
Euclid, Ohio 41132  
(216) 289-2660

6161 North May Avenue  
Suite 39W  
Oklahoma City, Oklahoma 73112  
(405) 840-3891

9726 E. 42nd South  
Suite 200  
Tulsa, Oklahoma 74145  
(918) 664-8530

### OKLAHOMA

6161 North May Avenue  
Suite 39W  
Oklahoma City, Oklahoma 73112  
(405) 840-3891

9726 E. 42nd South  
Suite 200  
Tulsa, Oklahoma 74145  
(918) 664-8530

### OREGON

4614 S.W. Kelly  
Portland, Oregon 97201  
(503) 223-7150

### PENNSYLVANIA

Philadelphia Office  
Dublin Hall, 4th Floor  
1777 Wallon Road  
Blue Bell, Pennsylvania 19422  
(215) 643-5515

Carnegie Office Park  
600 Bell Avenue  
Building 100  
Carnegie, Pennsylvania 15106  
(412) 279-3500

### RHODE ISLAND

225 Newman Avenue  
Rumford, Rhode Island 02916  
(401) 438-0001

### TENNESSEE

1515 East Magnolia Avenue  
Hamilton East Building  
Suite 130  
Knoxville Tennessee 37917  
(615) 524-1841

1355 Lynnfield Road  
Suite 221 Bldg. B  
Memphis, Tennessee 38138  
(901) 761-3366

### TEXAS

111 West Anderson Lane  
Suite 301  
Austin, Texas 78752  
(512) 458-4201

4450 Sigma Road  
Suite 130  
Dallas Texas 75240  
(214) 233-4496

3101 Fierro  
El Paso, Texas 79935  
4949 W. 34th Street  
Suite 123  
Houston, Texas 77092  
(713) 688-8841

3644 West 2100 South  
Salt Lake City, Utah 84120  
(801) 972-1338

UTAH

3644 West 2100 South  
Salt Lake City, Utah 84120  
(801) 972-1338

### VIRGINIA

1757 Old Meadow Road  
McLean, Virginia 22101  
(703) 893-0910

5 Koger Executive Center  
Suite 202  
Norfolk, Virginia 23502  
(804) 481-0933

5001 W. Broad Street  
Suite 209  
Richmond, Virginia 23230  
(804) 288-1675

417 Apersen Road  
Salem, Virginia 24153  
(703) 389-8155

### WASHINGTON

Seattle Office  
Suite 107  
10604 N.E. 38th Place  
Quad 1 North  
Kirkland, Washington 98033  
(206) 827-0503

### WISCONSIN

N81 W12920 Leon Road  
Suite 105  
Menomonee Falls, Wisconsin 53051  
(414) 255-3230



Data General Corporation, Westboro, Massachusetts 01581



# DATA GENERAL INTERNATIONAL OFFICES

## INTERNATIONAL OFFICES

### CANADA

415 Homer Road  
Toronto, Ontario M8W 4W3  
(416) 259-4271  
810-492-2553 TWX

438 Isabay  
St. Laurent, Montreal  
(514) 342-5730  
810-421-4758 TWX

210 Gladstone Avenue  
Suite 2003  
Ottawa, Ontario K29 0Y8  
(613) 563-1121  
810-562-8536 TWX

1020G Sheldridge Way  
Richmond V6X 2W7, B.C.  
(604) 273-2711  
810-923-5080 TWX

5923 3rd Street S.E.  
Calgary, Alberta  
(403) 253-8427  
810-821-2473 TWX

810 10339 - 124 Street  
Edmonton, Alberta  
(403) 482-3443  
810-831-1240 TWX

### AUSTRALIA

Melbourne Office  
96 Camberwell Road  
P.O. Box 447  
East Hawthorn, Victoria 3123  
(03) 821361  
790-33041 TLX

Perth Office  
43 Ventnor Avenue  
West Perth, W. Australia 8000  
092-215981  
790-92093 TLX

Sydney Office  
P.O. Box 117  
40 Yeo Street  
Neutral Bay Junction  
N.S.W. 2089  
(02) 9081368  
790-25046 TWX  
Brisbane Office  
87 St. Paul's Terrace  
Springhill, Queensland 4000  
(07) 229-5942  
790-40159 TWX

### NEW ZEALAND

Suite B, Level 13  
William City Center  
Plimmer Steps  
Wellington, New Zealand  
723-095  
791-31002 TLX

## EUROPEAN SUBSIDIARIES

### AUSTRIA

Gesellschaft MBH  
Gottfried Keller Gasse 2  
1030 Vienna, Austria  
(222) 73-4566/69  
847-74559 TLX

### BELGIUM

Data General, SA  
Vorstaan 191-197  
Bus 111  
Boulevard du Souverain  
Brussels 1160  
Brussels, Belgium  
(02) 660-49-44  
81206 TLX

### FRANCE

15, Rue le Sueur  
75116-Paris, France  
1-500-06-06-500-98-18  
810-289 TLX

La Boursidiere, RN 186  
Immeuble M BP 78  
Route Nationale 186  
92350-Le Plessis Robinson, France  
(1) 630-21-05  
200143 TLX

105-107 Rue De Crequi  
69006 Lyon, France  
(78) 5284-21  
842-38-130 TLX

### GERMANY

Data General, GMBH  
Frankfurter Allee 27  
8236 Eschborn  
W. Germany  
8196-48688  
841-417434 TLX  
Niederlassung West  
403 Ratingen, Germany  
Borsigstrasse 12  
(2012) 46011/12  
8-585-271 TLX

Niederlassung Hamburg  
Kleine Bahnstrasse 10  
2 Hamburg 54, West Germany  
(40) 850-50-26  
212448 TLX

Prinzregentenstrasse 54/IV  
8000-Munich, West Germany  
089-223833  
841-524079 TLX

Talstrasse 172  
D-7024 Filderstadt 1  
711-702061

### ITALY

Data General S.P.A.  
Via Morigi, 3/A  
20123 Milano, Italy  
866-048  
843-33243 TLX

Data General S.P.A.  
Via Vincenzo Usani 90  
1-00151 Roma  
(06) 5236691

### NETHERLANDS

Data General Holland, BV  
Van Gijnstraat 17  
Rijswijk Z.H. The Netherlands  
(70) 907-694  
844-33545 TLX

### SWEDEN

Data General, Sweden AB  
Industrivagen 20  
S-17148 Solna, Stockholm  
8-83-4400  
854-10089 TLX

Doktor Forselius Gata 1  
S-42326 Gothenburg  
31-81-31-90

Carlsgatan 54  
S-211 20 Malmoe  
30-10-35-85

### SWITZERLAND

Data General A.G.  
Badenerstrasse 587  
8048-Zurich, Switzerland  
1-54-17-82  
845-58437 TLX

Avenue de Valmont 12  
CH 1010 Lausanne, Switzerland  
021-33-33-34 or 35  
26105 TLX

### UNITED KINGDOM

London Office  
Westway House  
320 Ruislip Road East  
Greenford  
Middlesex, England  
1-578-9231  
851-935364 TLX

Manchester Office  
Nelson House, Park Road  
Timperley, NR Altrincham  
Cheshire, England  
(81) 869-3935/9  
667903 TLX

Swan Office Centre  
1506/8 Coventry Road  
Yardley  
Birmingham, England B258AD  
021-707-3433  
851-339392 TLX

### SCOTLAND

Hellenic House, 2nd Fl.  
87-97 Bath Street  
Glasgow G2 2EE  
Scotland, UK  
41-332-3205  
851-779258 TLX

### SOUTH AMERICA

Caixa Postal 673  
Sao Paulo 04559, Sp. Brazil  
543-0138  
1122336 EVAM BR TLX

### MIDDLE EAST

Data General Greece  
280 Syngrou Avenue  
Athens, Greece  
958-0845  
21/8708 TLX

### ARGENTINA

Compucorp Calculadores Y Sistemas, S.A.  
Bartolome Mitre 2480, Piso 1  
Buenos Aires Republica Argentina  
47-0065, 47-0068, 47-2224  
CGSSA-BA 2029 AR TLX

### COLOMBIA

MBB Compania Ltda.  
Carrera 7 N 81-26  
APDO Aereo 90870  
Bogota, Colombia  
389-585

### COSTA RICA

Sistemas Analticos, SA  
Apartado 6244  
San Jose, Costa Rica  
22-81-58  
2481 TLX FEGOR

### ECUADOR

Sistemas Andinos S.A.  
P.O. Box 4560-A  
Quito Ecuador  
521-875, 230-593  
2393 Bazoas ED TLX

### EGYPT

T.E.A. Computers  
2 Maaruf Street  
P.O. Box 1901  
Cairo, Egypt  
56655, 40077 Cable  
Teamincomp  
2478 Teacom Un TLX

### FINLAND & U.S.S.R.

Oy Stromberg AB  
P.O. Box 118  
SF 00101 Helsinki 10, Finland  
55-00-45  
122960 TLX

### GREECE

Corals S.A.  
10 Stadiou Street  
Athens 133, Greece  
32-33714  
55-28 TLX

### HONG KONG

Dataprep (HK), Ltd.  
14th Floor, Block B  
Watson's Estate  
Hong Kong, B.C.C.  
5-717-231  
780-73184 TLX

### IRAN

Datiran Corporation  
No 9 Eleventh Street  
Miremad Avenue  
Takhle Tavous Avenue  
Tehran 15 Iran  
215-308 MOOXIR

### ISRAEL

TEAM Computers Ltd.  
14 Hah Iyar Street  
Kikar Hamedina  
Tel Aviv, Israel  
(03) 256-819, 256-866, 262-580  
341491 TLX

### JAPAN

Nippon Mini-Computer Corporation  
Trade Dept.  
Tohyo Bldg. 6-12-20  
Jingumae  
Shibuya, Tokyo  
(03) 406-8451  
781-242-5071 TLX

### KOREA

M-C International (Korea) Ltd.  
CPO Box 1355  
Seoul, Korea  
23-410/5  
K24228 Emcee Korea Seoul  
Room 1407  
Center Bldg.  
SO-Kong Dony  
Seoul, Korea  
717 Market St.  
San Francisco, CA 94103  
(415) 397-1455

### LYBIA

T.E.A. Computers  
Ben Hour Street  
P.O. Box 4494  
Tripoly, Lybia  
37283  
Handiskawi Tripoly Lybia (Cable)

### MALAYSIA

Dataprep (Malaysia) Sendirian Berhad  
6th Floor, Wisma Keng Soon  
120 Jalan Pudu  
Box 2491  
Kuala Lumpur 05-03  
Malaysia  
3-88795  
784-30420 TLX

### PERU

Digita S.A.  
Los Negocros 240  
Surquillo Lima 34  
Peru  
45-5773, 47-1111  
21083 TLX

### PHILIPPINES

Dataprep (Philippines), Inc.  
G & S Building  
250 Beundia Avenue  
Makati, Metro, Manila, Philippines  
86-49-51  
722-7496 TLX  
3968-DPP-PN TLX



Data General Corporation, Westboro, Massachusetts 01581

# DATA GENERAL INTERNATIONAL OFFICES

## EUROPEAN SUBSIDIARIES

### PUERTO RICO

Pan American Computer Corporation  
1810 Banco Popular Center  
Hato Rey  
Puerto Rico 00918  
(809) 767-6524

### SAUDI ARABIA

Rola Trading Corporation  
P.O. Box 3172  
Riyadh  
Saudi Arabia

### SINGAPORE

Dalaprep (S) PTE Ltd.  
2nd Floor, Marina House, Shenton Way  
Singapore 1  
211311 DPREP  
786-21249 TLX

### SOUTH AFRICA

Perseus Computing & Automation, Ltd.  
Dejong Centre, 2nd Floor  
457 Rodericks Road Lynnwood  
Pretoria, Republic of South Africa  
47-27-82  
960-30643 TLX

Perseus Computing & Automation, Ltd.  
P.O. Box 4898  
Johannesburg  
Republic of South Africa  
23-6158

### SPAIN

Madrid Hispano Electronica Information  
Comandante Zorita 8  
Madrid 20, Spain  
2346304 & 2344316  
831-22404 TLX

Figols 27-29  
Barcelona - 14, Spain  
3218595 & 3219085

Zabalbide, 42  
Bilbao-6  
Spain  
4238309 & 4238082

Navarro Reverter, 2  
Valencia-4  
Spain 3731297

Sanchoel Sabio, 28  
San Sebastian, Spain  
462554

### URUGUAY

Lumiera S.A.  
Rincon 531-ESC. 401  
Montevideo, Uruguay  
90-4774  
546-A/B Quesmar UY TLX

### U.S.S.R.

Codeornitec  
6263 Varial Drive  
Woodland Hills, CA 91364  
213-348-1719  
698474 TLX

## OEM's

### MEXICO

Maquinas de Informacion S.A.  
Insegentes SUV No 1722-202  
Mexico 20 D.F.  
524-9195

### PORTUGAL

Equipamentos de Laboratorio, Ltd.  
P.O. Box 1100  
Lisbon, Portugal  
976551  
632-12702 TLX

### SRI LANKA (CEYLON)

Malliaraja Organization  
7 Braybrooke Place  
Colombo 2  
Sri Lanka  
Cable 1195 Colombo Rendezvous

### TURKEY

Sisag Company Ltd.  
CINNAH-CD, No. 5  
Ankara, Turkey

## MANUFACTURING SUBSIDIARIES

### HONG KONG

Data General Hong Kong, Ltd.  
Kaiser Estates, 5th Floor  
Man Yue Street  
Hung Hom  
Kowloon  
Hong Kong  
K-834381-4

### THAILAND

Data General Thailand, Ltd.  
2189 New Petchburi Road  
Bangkok  
Thailand  
916357



Data General Corporation, Westboro, Massachusetts 01581

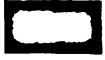
# **microNOVA INTEGRATED CIRCUITS DATA MANUAL INDEX**



Data General Corporation, Westboro, Massachusetts 01581

# INDEX

	<b>Overview of microNOVA Line Integrated Circuits</b>	<b>1</b>
<b>mN601</b>	<b>16-Bit Microprocessor</b>	<b>2</b>
<b>mN603</b>	<b>I/O Controller</b>	<b>3</b>
<b>mN606</b>	<b>4K Dynamic Random Access Memory</b>	<b>4</b>
<b>mN506</b>	<b>Quad Sense Amp/Bus Driver</b>	<b>5</b>
<b>mN629</b>	<b>CPU I/O Transceiver</b>	<b>6</b>
<b>mN634</b>	<b>Memory Transceiver</b>	<b>7</b>
<b>mN636</b>	<b>IOC I/O Transceiver</b>	<b>8</b>
<b>mN638</b>	<b>Memory Clock Driver</b>	<b>9</b>
<b>mN640</b>	<b>CPU/IOC Clock Driver</b>	<b>10</b>
	<b>Design Notes</b>	<b>11</b>
	<b>Appendices</b>	<b>12</b>



# OVERVIEW OF microNOVA LINE INTEGRATED CIRCUITS



Data General Corporation, Westboro, Massachusetts 01581

# OVERVIEW

## INTRODUCTION

This data manual describes Data General's full microcomputer chip set. The microcomputer consists of three principal integrated circuits: a 16-bit microprocessor, an I/O controller, and a 4Kx1 dynamic RAM. As part of the set, each integrated circuit is supported by the necessary system buffer elements. The principal chips are fabricated with N-channel, silicon gate, metal oxide semiconductor (NMOS) technology. The system buffer chips are small and medium scale integration, bi-polar circuits.

The chip set is designed to implement high-performance microcomputer or controller systems. It incorporates Data General's powerful NOVA line minicomputer instruction set and 16-bit central processing unit (CPU) architecture.

microNOVA line integrated circuits are available on an individual basis as well as in full microcomputer chip sets. In addition, the microNOVA family includes the following supporting hardware.

- CPU boards
- RAM memory boards
- programmable read-only memory boards
- asynchronous interface boards
- general purpose I/O boards
- power supply
- hand-held console
- card cages
- a variety of peripheral devices; such as:
  - video displays
  - hard copy data terminals
  - single and dual diskette sub-systems
  - line printers
  - paper tape reader

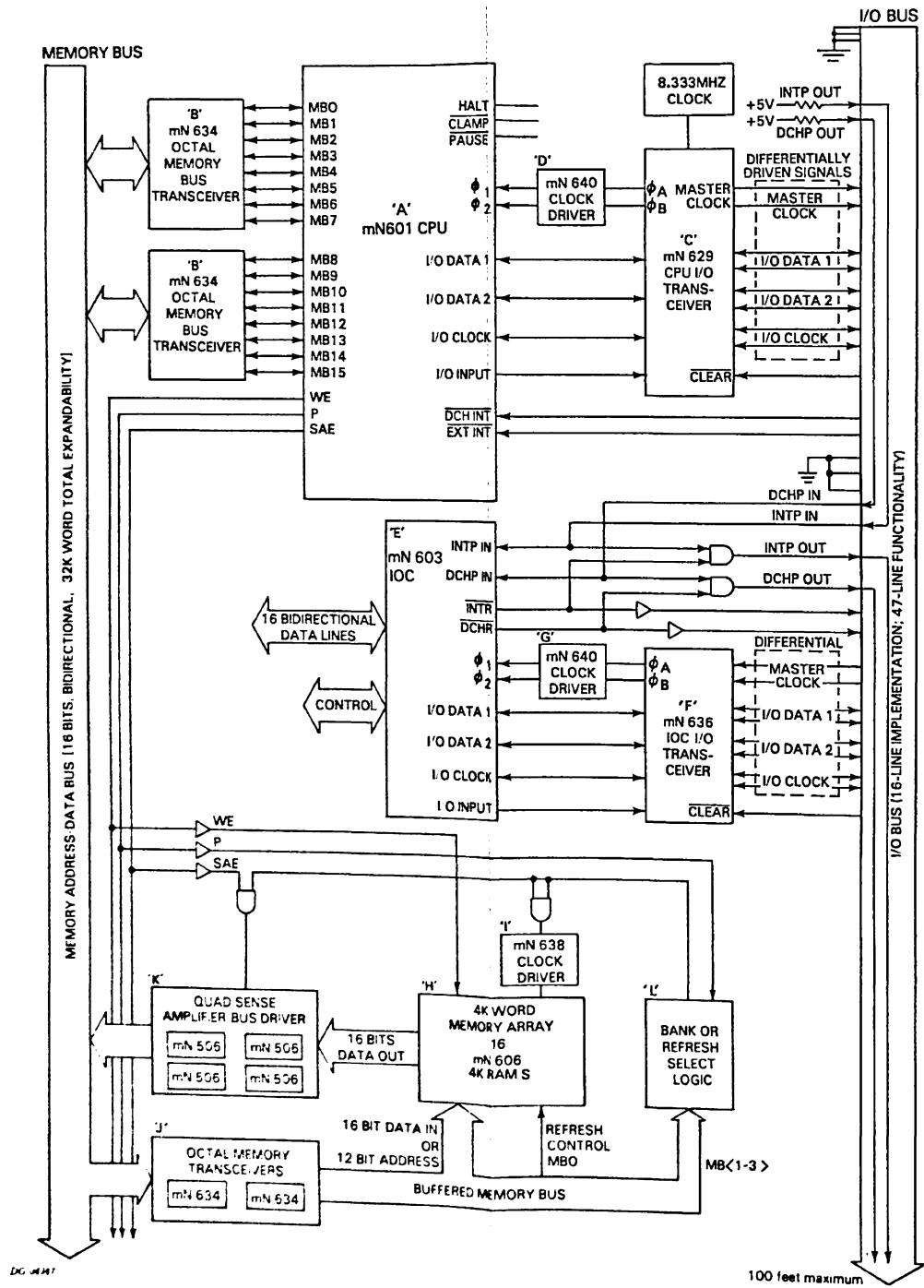
The microNOVA family comes with proven Data General development and runtime software that includes two operating systems, a diskette-based Disc Operating System (DOS) and the Real-Time Operating System (RTOS). Software developed for Data General's NOVA line minicomputers can be easily reassembled for use with microNOVA systems. The operating systems support two high level language compilers, FORTRAN IV and extended BASIC, as well as various utilities, including: a Command Interpreter, text editor, macro assembler, relocatable loader, a library file editor, and a symbolic debugger.

microNOVA development software is based on the Disc Operating System. The diskette-based DOS provides a smooth flow through varying development phases to completed production software. OEMs who need a larger development system can utilize the NOVA 3 line of minicomputers. NOVA 3 is software-compatible with microNOVA and supports up to 128K words of memory, a peripheral range, and the Real-time Disc Operating System (RDOS).

Data General's Real-Time Operating System is a DOS-compatible runtime executive. It is compact and memory-resident, and has a real-time multitask capability for controlling real-time applications. RTOS provides standard interrupt servicing, device polling and execution scheduling functions.

Most of these features make the microNOVA family ideal for applications like complex instrumentation, industrial automation, communications, and data acquisition, where 16-bit NOVA power and precision are required.

# microNOVA FUNCTIONAL BLOCK DIAGRAM



# OVERVIEW

## THE CHIP SET

Data General's microcomputer chip set consists of the following dual-in-line packaged (DIP), integrated circuits:

Integrated Circuit	Packaging
mN601 16-bit microprocessor	40-pin ceramic
mN603 I/O controller (IOC)	40-pin ceramic
mN606 4Kx1 dynamic RAM	20-pin plastic
mN640 CPU and I/O clock driver	16-pin cerdip
mN638 memory clock driver	16-pin cerdip
mN629 CPU I/O transceiver	20-pin ceramic
mN636 IOC I/O transceiver	20-pin ceramic
mN634 octal memory transceiver	20-pin ceramic
mN506 quad sense amp/memory bus driver	14-pin cerdip

## THE MICROCOMPUTER

The block diagram on the opposite page shows a recommended microcomputer chip set configuration. It is organized around two data buses: memory and I/O (see diagram).

### microNOVA MEMORY BUS

Memory addresses and data are passed between the mN601 microprocessor and an array of mN606 RAMs via the microNOVA memory bus. This is a 16-bit parallel, bidirectional, high-speed bus.

The microprocessor is interfaced to the bus via two mN634 octal memory transceivers. These transceivers drive data between the CPU's memory address/data lines and the memory bus.

A RAM grouping is interfaced to the bus via two mN634 octal memory transceivers and four mN506 quad sense amp/bus drivers. The memory transceivers drive both addresses and data sent to memory via the bus onto the RAM input lines while the sense amp/bus drivers drive data from the RAM output lines onto the bus.

Three memory control lines, P, SAE and WE, allow the microprocessor to perform four kinds of memory operations: read, write, read-modify-write and refresh.

### microNOVA I/O BUS

Information is passed between the mN601 microprocessor and mN603 I/O controllers via the bidirectional, microNOVA I/O bus. This bus combines the advantages of high speed with high noise immunity (due to its differential nature, described below), while permitting the use of low cost ribbon cable. The I/O bus can be extended to a maximum distance of 100 feet from the microprocessor.

The microNOVA I/O bus contains 16 lines. The most critical lines (two serial data lines and two clock lines) are differentially driven and received by an mN629 CPU I/O transceiver and one mN636 IOC transceiver per mN603 I/O controller.

At the microprocessor end of the bus, the CPU I/O transceiver receives serial data from the CPU and differentially drives it to the IOC I/O transceiver(s). It also receives the differentially driven data signals from the IOC I/O transceiver(s) and passes the data to the CPU. At the I/O controller end of the bus, each IOC I/O transceiver performs a similar function for its I/O controller. Up to twenty IOC I/O transceivers can be interfaced to the microNOVA I/O bus.



# OVERVIEW

The differentially driven I/O CLOCK (see diagram) is transmitted with the data; it strobes the data into the receiving circuit. The CPU I/O transceiver differentially drives the MASTER CLOCK to all IOC I/O transceivers. Each IOC transceiver passes the MASTER CLOCK to its IOC as two non-overlapping clocks via an mN640 clock driver. The MASTER CLOCK synchronizes the operation of the I/O controllers with the CPU.

A system reset line ( $\overline{\text{CLEAR}}$ ), two device request lines, two types of device priority lines, and three ground lines comprise the remainder of the bus.

The  $\overline{\text{CLEAR}}$  line allows the system to be reset by external hardware. When this line is pulled to a low logic level for a period of 10  $\mu$ secs, the microcomputer is reset. This time period ensures that the CPU and IOC I/O transceivers are in receive mode; refer to the mN601, mN603, mN629 and mN636 circuit descriptions.

The device request lines ( $\overline{\text{EXTINT}}$  and  $\overline{\text{DCHINT}}$ ) allow I/O devices to request microprocessor service for the purpose of transferring data in either of two modes: programmed I/O or data channel break. These are explained later in the section entitled Data Transfers.

## Hardware Device Priority Network

The priority lines (INTP OUT, DCHP OUT, INTP IN and DCHP IN) comprise an interdevice daisy chain. When more than one I/O device is requesting service, this network gives priority to the device requesting service that is physically closest on the I/O bus to the microprocessor. The INTP lines control priority between devices requesting interrupt service. Similarly, the DCHP lines control priority between devices requesting data channel service.

The manner in which the priority system is implemented is the same for both types of requests. When a device requests service, that device's INTP/DCHP OUT line goes to the low logic level. This removes priority from devices further away from the microprocessor, by pulling their INTP/DCHP IN lines to a low logic level. When the microprocessor acknowledges a request for service, a device can respond only if it is requesting service and its INTP/DCHP IN line is at a high logic level. If it is at a low logic level, it indicates that a higher priority device in the chain is requesting service. The priority lines previously disabled by the service request are reenabled when the responding device clears its request line.

## mN601 16-BIT MICROPROCESSOR

This is a 16-bit central processing unit (CPU) that incorporates the NOVA architecture. Its internal structure and operations are described in the mN601 section of this manual.

The mN601 is designed to operate with the following system buffer elements: two mN634 octal memory transceivers, an mN629 CPU I/O transceiver and an mN640 clock driver. The primary functions performed by these circuits for the CPU are described below. Detailed information is found in the individual circuit descriptions.

The CPU (designated 'A' in the diagram) has two ports: memory and I/O. Two mN634 octal memory transceivers (designated 'B') drive data between the CPU's memory address/data lines MB (<0-15>) and the microNOVA memory bus. The memory control signals, P, SAE and WE, allow the CPU to perform four kinds of memory operations: read, write, read-modify-write and refresh.

The mN629 CPU I/O transceiver (designated 'C') performs two functions for the CPU. First, it interfaces the CPU's serial data (I/O DATA <1,2>) and data clock (I/O CLOCK) lines with the microNOVA I/O bus. The I/O INPUT signal is used by the CPU to control the transceiver's operating mode, either transmit or receive. Second, the CPU I/O transceiver generates the two non-overlapping system clocks ( $\phi A$  and  $\phi B$ ) from a single 8.333MHz clock input.

The  $\phi A$  and  $\phi B$  clocks are shaped by external circuitry (see Design Notes) and the outputs are supplied to the mN640 clock driver (designated 'D'). The clock driver receives these signals and provides signal characteristics that are compatible with the MOS clock input circuits of the CPU,  $\phi 1$  and  $\phi 2$ .

The remaining mN601 pins perform the following functions. The HALT pin reflects the state of the CPU, halted (idling) or performing tasks. The PAUSE pin indicates when the CPU is performing a memory operation. By monitoring the output of this pin, multiporting of memory can be implemented. The  $\overline{\text{CLAMP}}$  pin is an input pin used during the power-up and initialization sequence.

# OVERVIEW

## mN603 I/O CONTROLLER (IOC)

The IOC is the interface between a peripheral device and the microNOVA I/O bus. Details of its internal architecture and operations are contained in the mN603 section of this manual.

It is designed to operate with the following system buffer elements: an mN636 IOC I/O transceiver and an mN640 clock driver. The primary functions performed by these circuits for the IOC are described below. Detailed information is found in the individual circuit descriptions.

The mN603 (designated 'E' in the diagram) has two ports: I/O bus and peripheral.

The mN636 IOC I/O transceiver (designated 'F') performs two functions for the IOC. First, it interfaces the IOC's serial data (I/O DATA<1,2>) and data clock (I/O CLOCK) lines with the microNOVA I/O bus. The I/O INPUT signal is used by the IOC to control the transceiver's operating mode, transmit or receive. Second, the mN636 transceiver receives the MASTER CLOCK from the I/O bus and generates two non-overlapping clock signals,  $\phi A$  and  $\phi B$ .

The  $\phi A$  and  $\phi B$  non-overlapping clocks are shaped by external circuitry (see Design Notes) and the outputs are supplied to the mN640 clock driver (designated 'G'). The clock driver receives these signals and provides signal characteristics that are compatible with the MOS clock input circuits of the IOC,  $\phi 1$  and  $\phi 2$ .

The IOC communicates with the device via a 16-bit parallel data bus and a group of control lines. The data and control lines together create the functional equivalent of the 47-line NOVA I/O bus.

## mN606 4Kx1 DYNAMIC RAM

This is a 4096 by one bit memory. Details of its internal architecture and operations are contained in the mN606 section of this manual.

Sixteen mN606 RAMs comprise a 4Kx16-bit memory array. This array is designed to operate with the following system buffer elements: one mN638 clock driver, two mN634 octal memory transceivers and four mN506 quad sense amp/bus drivers. For an 8Kx16-bit array, it is necessary to add only one mN638 clock driver, since the two mN634 transceivers and four mN506 sense amp/bus drivers support 8K of memory. The primary functions performed by these system buffer elements for the mN606 RAMs are described below. Detailed information is found in the individual circuit descriptions.

The 4Kx16-bit dynamic RAM array (designated 'H' in the diagram) has two ports: input (address and data) and output.

The two mN634 octal memory transceivers (designated 'J') drive addresses and data from the microNOVA memory bus onto the RAM address and data input lines. The four mN506 quad sense amp/bus drivers (designated 'K') drive data from the RAM output lines onto the microNOVA memory bus.

The microprocessor addresses up to 32,768 memory locations; thus, eight 4Kx16-bit memory arrays can be interfaced to the microNOVA memory bus. Each 4Kx16-bit array has a 12-bit memory address (MB <4-15>). The external bank select logic (designated 'L') selects the appropriate array from the address appearing on the MB <1-3> lines.

When both the memory control signal P is asserted and the bank select logic is enabled, the mN638 clock driver (designated 'I') provides the signal characteristics required for the MOS clock input circuit of the RAM.

At the beginning of a memory operation, the RAM samples the address and tests the logic level, high or low, of MB0. If MB0 is at a high level, the RAM initiates a refresh operation; if it is at a low level, the RAM initiates either a read, write, or read-modify-write operation.

When mN606 dynamic RAMs are used in a microNOVA-based system, the mN601 microprocessor automatically refreshes them without external circuitry.

# OVERVIEW

## DATA TRANSFERS

Addresses and data are transferred between the CPU and memory via the microNOVA memory bus. Control information and data words are transferred between the CPU and I/O devices via the microNOVA I/O bus.

### MEMORY

The mN601 performs four kinds of memory operations: read, write, read-modify-write and refresh. All memory operations are initiated on the rising edge of P. As described above, the address on the bus is sampled by the memories and the state of MB0 is tested. If this signal is high, a refresh operation is performed; if it is low, either a read, write, or read-modify-write operation is performed. Detailed information is found in the Memory Operations section of the mN601 circuit description.

When non-volatile memory is required, programmable read-only memory (PROM) can be used. Refer to the microNOVA Computer Systems Technical Reference (DGC ordering number 014-000073). At the board level, PROM can overlay RAM, and PROM priority control lines resolve any memory conflicts in favor of the PROM.

### INPUT/OUTPUT

Information is passed between the CPU and I/O controllers via the two serial data lines of the microNOVA I/O bus. The bus connections are made through the appropriate I/O transceivers. The data consists of both control information and 16-bit data words.

The CPU performs I/O data transfers under two operating modes: programmed I/O and data channel break. Under programmed I/O, the CPU moves data between one of its internal registers (namely, accumulators; see mN601 section) and an I/O device. Under data channel break, the CPU moves data between memory and an I/O device. Data channel breaks allow blocks of information to be transferred without altering program flow. In addition, data channel breaks are transparent to the executing program.

An I/O device requests microprocessor service for the purpose of transferring data by asserting either the program interrupt ( $\overline{\text{EXTINT}}$ ) or data channel break ( $\overline{\text{DCHINT}}$ ) device request lines. These requests are synchronized by a CPU-generated signal called Request Enable. It ensures that the device request and priority lines are at a steady state when they are examined by the CPU. Request Enable is sent to all I/O controllers connected to the I/O bus as a two-bit code (11), one bit carried on each serial data line.

Detailed information concerning I/O data transfers is found in the I/O Operations section of the mN601 circuit description and in the mN603, mN629 and mN636 circuit descriptions. Timing diagrams of I/O bus operations can be seen in Appendix A. These diagrams depict the transmission and reception of control information and data words as they pass between the CPU, the CPU transceiver, the IOC transceiver and the I/O controller, under both programmed I/O and data channel. (The I/O transactions appearing in the diagrams apply when the system is configured as shown in the block diagram at the beginning of this section.)

#### Programmed I/O

Under programmed I/O, the CPU sends the I/O instruction, exactly as fetched from memory, to all controllers connected to the microNOVA I/O bus. The instruction includes information that identifies the controller to which it is directed. Following the instruction, the CPU performs either a data out operation or a data in operation, as required by the instruction.

When either data out or data in instructions are performed, both the instruction and the 16-bit data word are transmitted as 18-bits, 9 bits on each line. The first bit on each line comprises a two-bit code that identifies the type of transfer; 00 indicates an I/O command and 01 indicates a data word.

# OVERVIEW

## Data Channel Break

When the CPU performs a data channel break, it sends an address request to the microNOVA I/O bus. This request consists of a two-bit code (10). The highest priority device requesting data channel service responds to this code. The device responds first by sending a 16-bit word (plus the two-bit data word prefix code, 01, described above) containing a 15-bit memory address and a direction of transfer bit (data out or data in) to the CPU. Following the receipt of this information, the CPU initiates a memory operation at the address specified; then, it examines the direction of transfer bit.

If it is set for a data out, the CPU transmits the 16-bit word received during the read portion of the memory operation together with the two-bit data word prefix code. The memory operation is terminated without altering the contents of the addressed location.

If it is set for a data in, the device transmits the 16-bit word to be entered in memory together with the two-bit data word prefix code. After the data word is received by the CPU, it is written into memory during the write portion of the memory operation in progress.

## RELATED DOCUMENTS

The following supporting documents are available from Data General Corporation.

	Ordering No.
microNOVA Programmer's Reference	015-000050
microNOVA DTOS	015-000059
microNOVA Computer Systems	014-000073

**mN601**  
**16-BIT MICROPROCESSOR**  
**CONTENTS**



# mN601

## CONTENTS

<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>4</b>	<b>ARCHITECTURE</b>
<b>6</b>	<b>INSTRUCTION SET</b>
<b>7</b>	<b>MEMORY REFERENCE INSTRUCTIONS</b>
<b>7</b>	<b>Addressing Modes</b>
<b>8</b>	<b>INDIRECT ADDRESSING</b>
<b>9</b>	<b>AUTO-INCREMENTING/DECREMENTING</b>
<b>11</b>	<b>No-Accumulator-Effective Address Instructions</b>
<b>11</b>	<b>One Accumulator-Effective Address Instructions</b>
<b>12</b>	<b>TRAP INSTRUCTION</b>
<b>13</b>	<b>ARITHMETIC/LOGICAL INSTRUCTIONS</b>
<b>15</b>	<b>Two Accumulator-Multiple Operation Instructions</b>
<b>17</b>	<b>Multiply/Divide Instructions</b>
<b>18</b>	<b>STACK INSTRUCTIONS</b>
<b>23</b>	<b>INPUT/OUTPUT INSTRUCTIONS</b>
<b>23</b>	<b>Busy and Done Flags</b>
<b>23</b>	<b>Addressing Non-Existent Device Codes</b>
<b>26</b>	<b>PROGRAM INTERRUPTS</b>
<b>26</b>	<b>16-Level Priority Interrupts</b>
<b>27</b>	<b>CPU I/O INSTRUCTIONS</b>
<b>32</b>	<b>OPERATIONS PROTOCOL</b>
<b>32</b>	<b>MEMORY OPERATIONS</b>
<b>32</b>	<b>Address/Data Pins</b>
<b>32</b>	<b>Control</b>
<b>33</b>	<b>READ</b>
<b>33</b>	<b>WRITE</b>
<b>34</b>	<b>READ-MODIFY-WRITE</b>
<b>34</b>	<b>REFRESH</b>

# mN601

## CONTENTS

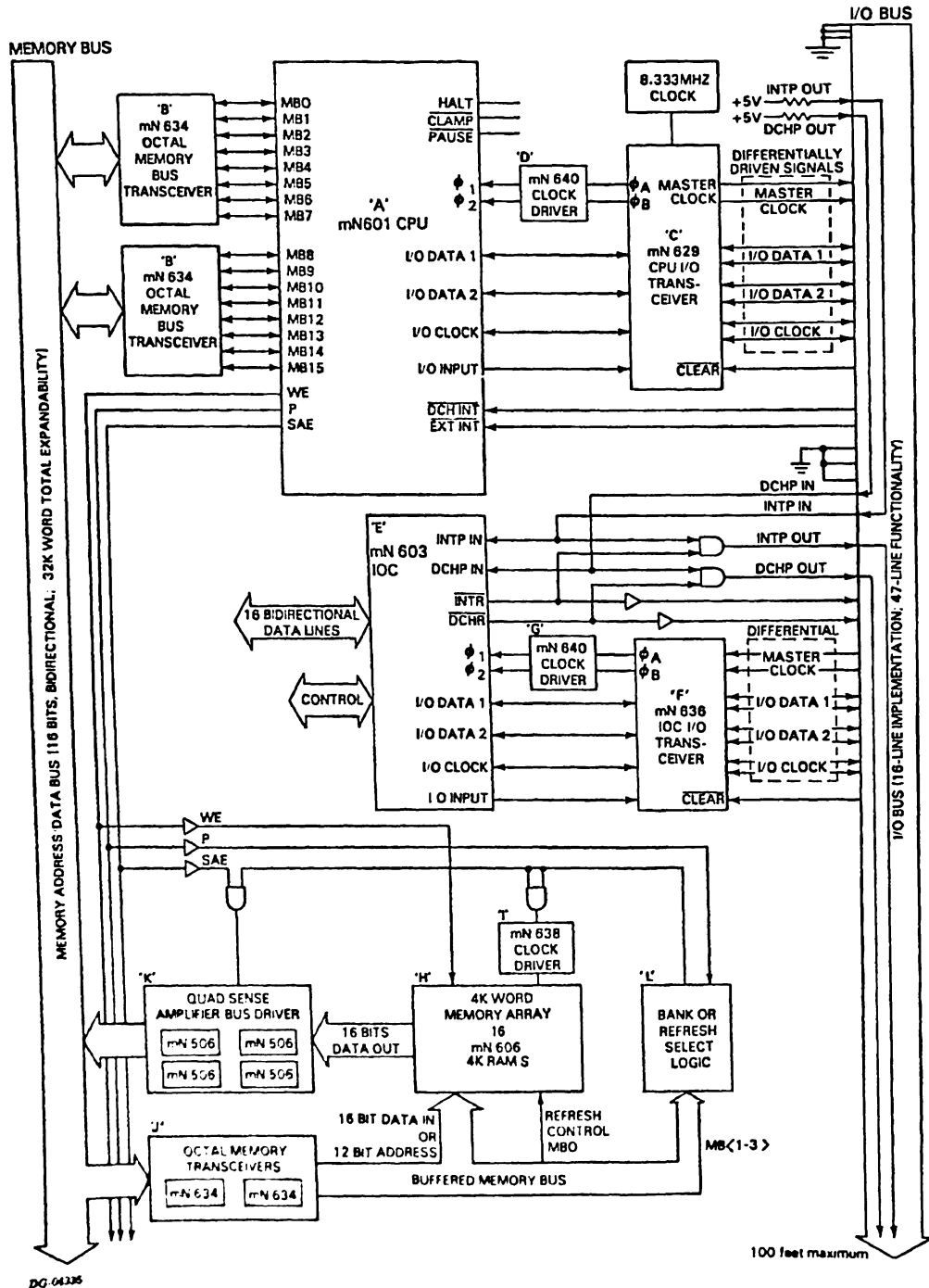
<b>35</b>	<b>I/O OPERATIONS</b>
<b>35</b>	<b>Programmed I/O</b>
<b>35</b>	<b>Data Channel Break</b>
<b>35</b>	<b>I/O Bus Transfers</b>
<b>36</b>	<b>Request Enable</b>
<b>37</b>	<b>Data Channel Address Request</b>
<b>37</b>	<b>Data</b>
<b>38</b>	<b>I/O Command</b>
<b>39</b>	<b>I/O Transfer Protocol</b>
<b>39</b>	PROGRAMMED I/O DATA OUT
<b>39</b>	PROGRAMMED I/O DATA IN
<b>40</b>	DATA CHANNEL DATA OUT
<b>40</b>	DATA CHANNEL DATA IN
<b>41</b>	<b>Timing</b>
<b>41</b>	PROGRAMMED I/O LATENCY
<b>42</b>	DATA CHANNEL LATENCY
<b>43</b>	<b>STATUS</b>
<b>43</b>	Initialization
<b>43</b>	Halt
<b>43</b>	Reset
<b>44</b>	<b>INSTRUCTION EXECUTION TIMES (IN CPU CYCLES)</b>
<b>45</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>45</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>45</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>45</b>	<b>DC CHARACTERISTICS</b>
<b>47</b>	<b>AC CHARACTERISTICS</b>
<b>47</b>	Switching Diagrams
<b>48</b>	Transition Timing
<b>49</b>	<b>TIMING TABLES</b>
<b>49</b>	<b>MEMORY READ/WRITE</b>
<b>49</b>	<b>I/O BUS</b>
<b>49</b>	<b>HALT</b>
<b>50</b>	<b>PACKAGE SPECIFICATIONS</b>



**This page intentionally left blank.**



# mN601 16-BIT MICROPROCESSOR



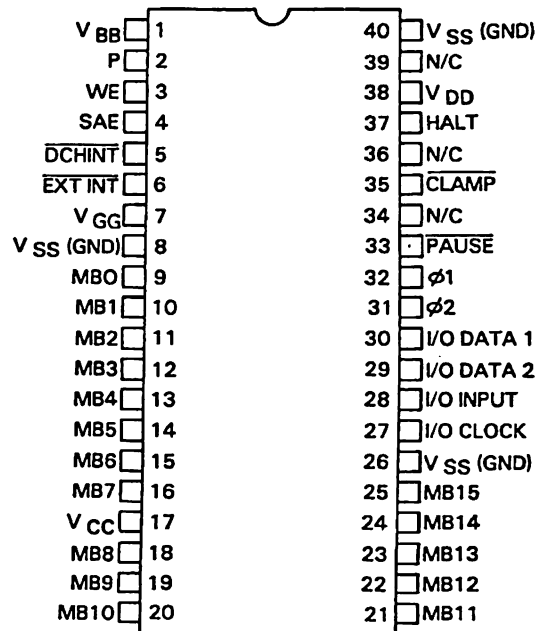
# mN601

## 16-BIT MICROPROCESSOR (CPU) NMOS INTEGRATED CIRCUIT

### FEATURES

- FULL NOVA 16-BIT ARCHITECTURE AND INSTRUCTION SET IN A SINGLE 40-PIN NMOS SILICON-GATE CHIP
- INTEGRAL DATA CHANNEL, REAL-TIME CLOCK, AND MULTIPLY/DIVIDE
- INTEGRAL HIDDEN REFRESH AND CONTROL LOGIC FOR DYNAMIC RAMs
- COMPREHENSIVE STACK ARCHITECTURE WITH HARDWARE STACK AND FRAME POINTER REGISTERS
- POWERFUL SAVE AND RETURN INSTRUCTIONS FOR SUBROUTINE CALLS
- MULTIPLE ACCUMULATORS
- SYSTEM MEMORY SUPPORT UP TO 32K WORDS OF RAM/PROM
- SINGLE WORD INSTRUCTION FORMAT IN A VARIETY OF ADDRESSING MODES - ABSOLUTE, RELATIVE, INDEXED, DEFERRED AND AUTO-INCREMENT/DECREMENT
- SEPARATE MEMORY AND I/O BUSES
- I/O BUS PROVIDES FUNCTIONAL EQUIVALENT OF 47-LINE NOVA I/O BUS
- 16-LEVEL PROGRAMMED PRIORITY INTERRUPT

### PACKAGE



DG-04026

### GENERAL DESCRIPTION

The mN601 is a full 16-bit NMOS microprocessor that provides the central processing function for Data General's microNOVA family. The mN601 features NOVA 16-bit architecture, including hardware multiply/divide; multiple addressing modes, including absolute, relative, indexed, deferred, and auto-increment/decrement; multiple accumulators, including two that can be used as index registers; hardware stack and frame pointers with stack overflow protection; programmed priority interrupt to 16 levels; and separate memory and I/O buses.

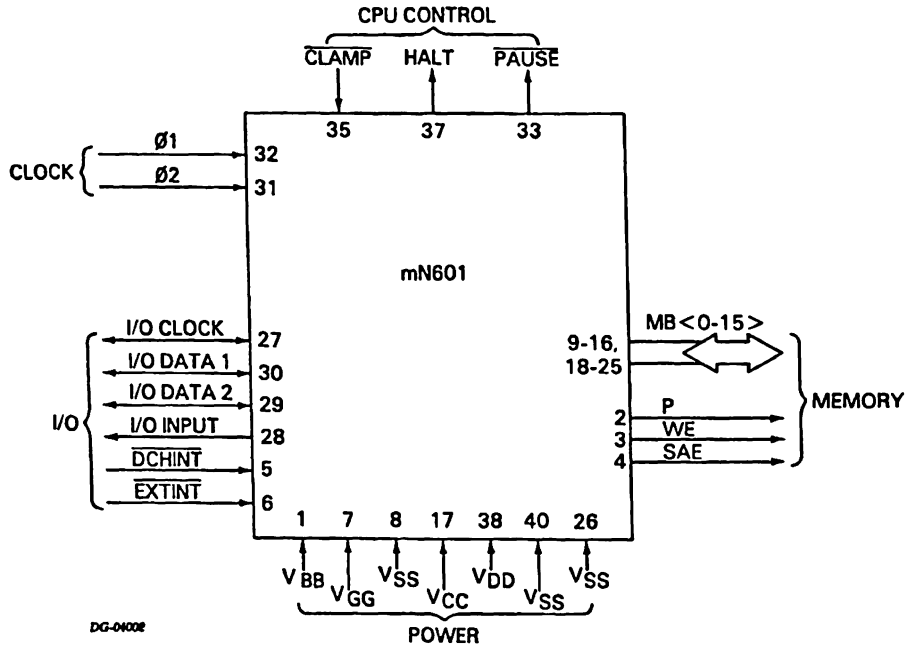
Data General's microNOVA family of integrated circuits also includes memories, peripheral controllers, and supporting circuits which can be used to develop a complete, high-performance, microcomputer system.

The mN601 is supported by the mN606 4K RAM, the mN603 I/O Controller (IOC), and system buffer elements. The 4096-bit mN606 uses cost-effective dynamic NMOS RAM technology to support microNOVA's large memory orientation. The mN603 IOC delivers the full functional capability of the 47-line NOVA I/O bus, including a parallel 16-bit I/O port. System buffer elements allow microNOVA support up to 32K words of memory and a full complement of peripherals.

# mN601

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



The following table describes the function of each pin shown in the pin connection diagram.

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	I/O	FUNCTION
$\phi 1$ $\phi 2$	32 31	IN IN	<p style="text-align: center;"><b>CLOCK</b></p> <p>Two-phase system clock. Operates between 0 and 14V amplitude. Generates 4-phase internal clock, providing internal control timing.</p>
MB0-MB15	9-16 18-25	I/O	<p style="text-align: center;"><b>MEMORY BUS</b></p> <p>Bi-directional, data and address bus. When used as a data bus, MB0 contains the MSB (most significant bit) and MB15, the LSB (least significant bit). As an address bus, MB&lt;1-15&gt; contains the address and MB0 defines the memory operation as either refresh or read/write.</p> <p>This bus is precharged internally by the CPU. It is held high by individual internal pull-up resistors tied to each line. During address or data transfers, each line transmitting a 0 is discharged; each line transmitting a 1 is undischarged. See the Memory Operations Section for detailed information.</p>
P	2	OUT	<p style="text-align: center;"><b>MEMORY CONTROL</b></p> <p>Active high. Initiates all memory cycles. When P goes high, indicates a valid address is on the memory bus.</p>
SAE	4	OUT	Active high. Enables sense amplifier outputs on the memory bus during a read operation.
WE	3	OUT	Active high. Enables write operation. On the falling edge of WE, information on the memory bus is written into memory.

# mN601

## PIN DESCRIPTIONS

### PIN DESCRIPTIONS (Cont.)

MNEMONIC	PIN NO.	I/O	FUNCTION
<b>I/O BUS</b>			
I/O DATA 1	30	I/O	Two-line, bi-directional bus. All data and I/O command information is transmitted serially between the mN601 and I/O devices at the MASTER CLOCK rate. During a 16-bit transfer, I/O DATA1 carries the most significant 8 bits (0-7); I/O DATA2, the least significant bits (8-15).
I/O DATA 2	29	I/O	
I/O INPUT	28	OUT	Indicates direction of I/O transfer. When low, indicates a transfer from the mN601; when high, a transfer to the mN601.
I/O CLOCK	27	I/O	Synchronizes all I/O transfers to and from the mN601. Holding I/O CLOCK low for 10 $\mu$ secs resets the microcomputer. I/O CLOCK should not be pulled low when I/O INPUT is in the low state.
$\overline{\text{EXTINT}}$	6	IN	Active low. When asserted low by a device, initiates a program interrupt, providing CPU interrupts are enabled.  When the mN601 is in the HALT state (see below), it is started by generating a program interrupt.
$\overline{\text{DCHINT}}$	5	IN	Active low. When asserted low by a device, the CPU will execute a data channel break. The mN601 responds to data channel break requests while it is in the HALT state as well as during operations.
<b>CPU CONTROL</b>			
HALT	37	OUT	Active high. When the CPU is halted (idling), as the result of either resetting the CPU by pulling I/O CLOCK low or the execution of a HALT instruction, HALT generates a sequence of .417MHz pulses.  In the HALT state, the mN601 performs the following functions: <ul style="list-style-type: none"> <li>● It generates Request Enable signals (see mN601 I/O Operations section) that synchronize program interrupt and data channel requests.</li> <li>● It responds to program interrupts and data channel requests. (The normal way to start the mN601 is to generate a program interrupt.)</li> <li>● It generates dynamic RAM refresh signals to prevent the loss of information stored in RAM.</li> </ul>
$\overline{\text{CLAMP}}$	35	IN	Active low. When the mN601 is first powered up, $\overline{\text{CLAMP}}$ should be held low for a minimum of 100 $\mu$ sec. Then, when it goes high, the CPU is properly initialized and enters the HALT state.
$\overline{\text{PAUSE}}$	33	OUT	Active low. $\overline{\text{PAUSE}}$ goes to the low level for a period of 240ns when the CPU is not using memory. During this period it is permissible to disable the $\phi 1$ and $\phi 2$ clocks, thus allowing multiporting of memory.
<b>POWER</b>			
V <sub>BB</sub>	1		-4.25V $\pm$ .5V
V <sub>CC</sub>	17		+5V $\pm$ .25V
V <sub>DD</sub>	38		+10V $\pm$ 1V
V <sub>GG</sub>	7		+14V $\pm$ 1V
V <sub>SS</sub>	8,26,40		Ground
	34		Reserved for future use
	36		Reserved for future use
	39		Reserved for future use

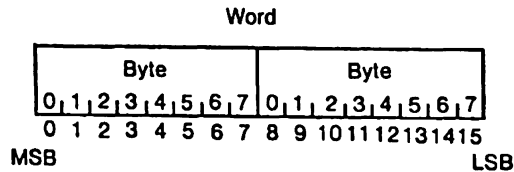
# mN601

## ARCHITECTURE

### ARCHITECTURE

The mN601 is a 16-bit parallel word central processing unit which executes the full NOVA line minicomputer instruction set, including an extended instruction set for stack and trap routines.

The instruction set addresses 32,768 words (65,536 bytes) of memory. Each memory address is 15-bits long. Each word is 16-bits long, consisting of two 8-bit bytes. Both words and bytes can be used as operands.



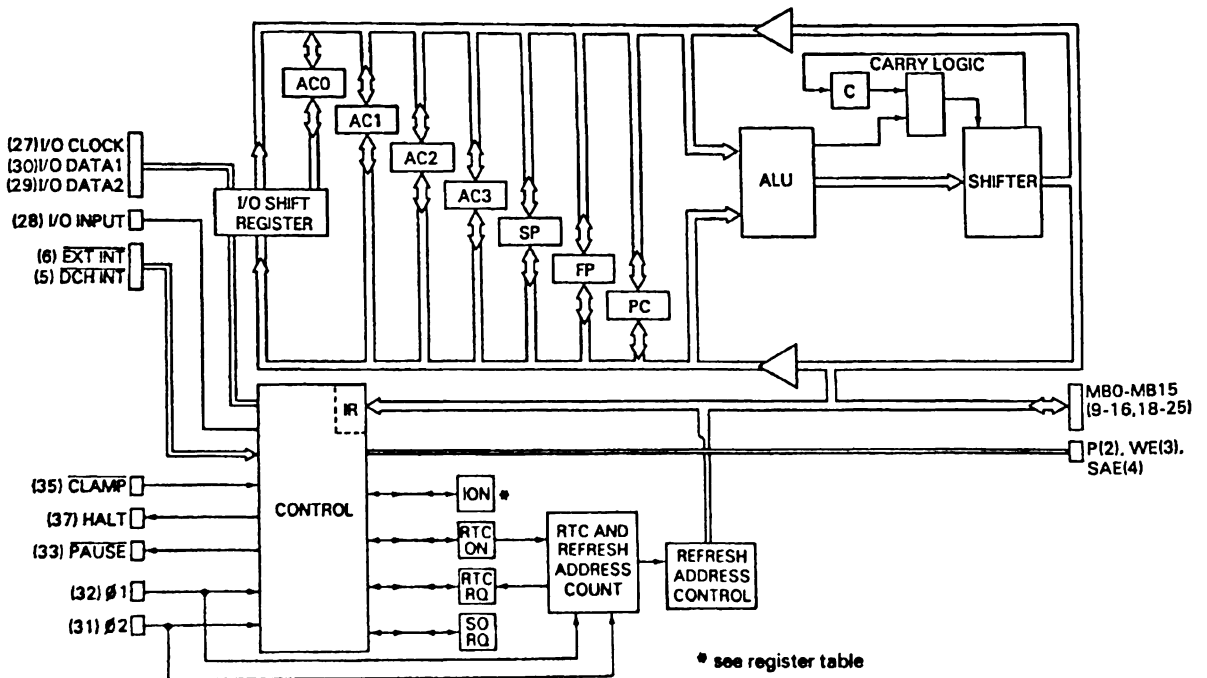
One word containing two bytes

The mN601 communicates with memory via a bi-directional, 16-bit parallel address and data bus. Three control lines direct the four memory operations: read, write, read-modify-write and refresh.

The mN601 communicates with I/O controller interfaces via two, bi-directional, serial lines that carry data and I/O commands. One control line indicates the direction of transfer, to or from the CPU, and a clock line synchronizes the transfer of information. Two interrupt lines, programmed I/O and data channel, allow I/O interfaces to request processor time.

The Expanded Block Diagram shows the internal architecture of the mN601, together with the memory bus, the I/O bus, and their respective control lines.

### EXPANDED BLOCK DIAGRAM



DG-06002

# mN601 ARCHITECTURE

Within the mN601, information is stored in the thirteen registers listed below.

REGISTERS	USER ACCESSIBLE	SIZE
accumulator 0	yes	16 bits
accumulator 1	yes	16 bits
accumulator 2	yes	16 bits
accumulator 3	yes	16 bits
program counter	yes	15 bits
stack pointer	yes	15 bits
frame pointer	yes	15 bits
carry bit	yes	1 bit
interrupt enable bit (ION)	yes	1 bit
real-time-clock enable bit (RTC ON)	yes	1 bit
real-time-clock request bit (RTC RQ)	no	1 bit
stack overflow request bit (SO RQ)	no	1 bit
refresh address counter	no	6 bits

- The four accumulators provide scratchpad memory, or work space, during program execution.
- The program counter contains a 15-bit memory address that is used by the processor when fetching the next instruction from memory during program execution. When the instruction is completed, the program counter is either automatically incremented by one or is modified by the program, before the next instruction is fetched.
- The stack pointer contains the memory address of the highest location in memory used by the stack. As information is pushed onto the stack, this address is appropriately incremented; as information is popped off the stack, the address is decremented.
- The frame pointer contains the memory address of a location within the stack. When a RETURN instruction is executed, this address is used as a reference for retrieving information from the stack.
- The carry bit reflects the results of certain arithmetic computations performed by the arithmetic logic unit (ALU), e.g., addition overflow.
- The interrupt enable bit can be controlled by the program. When enabled (set to 1), the processor responds to program interrupt requests.
- The real-time-clock enable bit is set by the program. When enabled (set to 1), the real-time clock contained in the processor generates an interrupt at 1.8432ms intervals.
- The real-time-clock request bit generates a program interrupt at fixed intervals of 1.8432ms when both the real-time clock and interrupts are enabled.
- The stack overflow request bit generates a program interrupt when the stack overflows a 256-word memory address boundary, e.g., 377<sub>8</sub> to 400<sub>8</sub>.
- The refresh address counter generates a 6-bit address specifying 64 numbers used to address 1/64 of all the available memory to be refreshed. All 32,768 RAM memory locations are refreshed at least once every 1.8432ms.

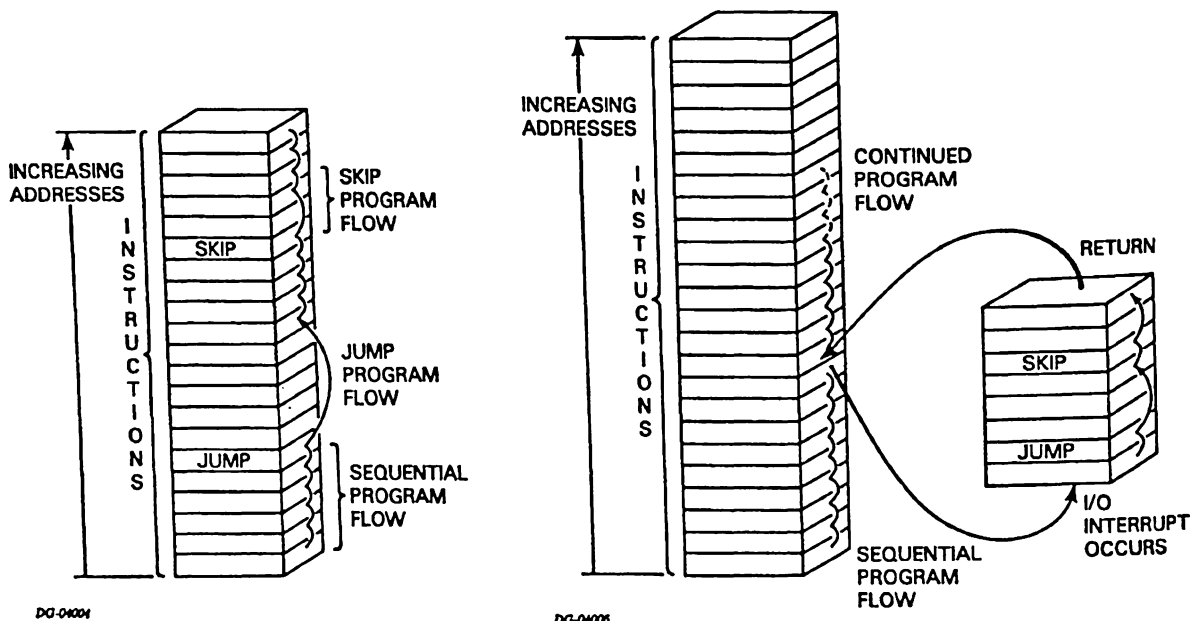
# mN601

## INSTRUCTION SET

### INSTRUCTION SET

Each mN601 (microNOVA) instruction is contained in one 16-bit word. Programs for the mN601 consist of sequences of instructions which are stored in external memory. The order in which these instructions are executed depends on the 15-bit memory address in the program counter register. During program execution, the processor sends this address to memory and fetches the contents of the specified memory location, via its memory address/data pins. During the execution of an instruction, information is moved between the processor's internal registers, memory or I/O data pins. When the instruction is completed, the program counter is updated. The CPU uses the contents of the program counter to address the next location.

Sequential operation can be altered by a jump or conditional skip instruction or by an interrupt. Jump instructions load a new address in the program counter; conditional skip instructions increment the program counter address by two if a specified test is true. In both cases, sequential operation is resumed from the updated address in the program counter. When the program is interrupted (e.g., I/O external interrupts, real-time clock interrupts, fault conditions of various kinds), the interrupt facility stores the next sequential program address in a specified location in memory so that the interrupt handler can return control to the interrupted program at the proper entry point. Then, the program counter is loaded with the memory address of the interrupt handler program and sequential operation resumes.







# mN601

## MEMORY REFERENCE INSTRUCTIONS

### EFFECTIVE ADDRESS CALCULATION

Absolute	Displacement is an unsigned number within the range 00000-00377 <sub>8</sub> that addresses one of the first 256 locations in memory, referred to as Page Zero. If the indirect bit is 0, the displacement is the effective address; if it is 1, the displacement is an indirect address (see the description of indirect addressing below).
PC relative	Displacement is a signed number that is added to the contents of the program counter. If the indirect bit is 0, the results of the addition are the effective address; if it is 1, the results are an indirect address.
AC2 indexed	Displacement is a signed number that is added to the contents of accumulator 2. If the indirect bit is 0, the results of the addition are the effective address; if it is 1, the results are an indirect address.
AC3 indexed	Displacement is a signed number that is added to the contents of accumulator 3. If the indirect bit is 0, the results of the addition are the effective address; if it is 1, the results are an indirect address.

**NOTE:** A memory address is always 15 bits long. When the results of an addition overflow 15 bits, the overflow is ignored.

Example: Program Counter = 77774<sub>8</sub>  
 Displacement = + 012<sub>8</sub>  
 Result = 000006<sub>8</sub> ; not 100006<sub>8</sub>

#### Indirect Addressing

When the indirect bit (bit 5) contains a 1, the calculated address is an intermediate (indirect) address that points to an effective address stored in memory. In this case, the processor fetches the contents of the calculated address and tests bit 0 (indirect sign bit when stored in memory). If it is 0, bits 1 through 15 of the contents of the indirect address are used as the effective address; if it is 1, bits 1 through 15 are used as an indirect address. In the latter case, the indirection chain continues until the processor either fetches a word in which bit 0 equals 0 (in this case, the processor has the effective address) or has executed eight levels of indirection (or five levels if the indirection chain accesses auto-incrementing/decrementing locations, explained below).

An internal counter is used to control the number of indirect addresses through which the processor searches for an effective address. This counter is initialized when the indirect bit is set to 1. Each time a word in the indirection chain is retrieved from memory the counter is incremented by two (or by three if an auto-increment or decrement location is indirectly addressed). If the counter becomes greater than 17 before an effective address is retrieved, the processor halts.

# mN601

## MEMORY REFERENCE INSTRUCTIONS

### Auto-Incrementing/Decrementing

Memory locations with addresses within the range  $020_8$  through  $027_8$  are called auto-incrementing locations. When one of these locations is indirectly addressed, the processor automatically increments bits 0 through 15 of the contents of the indirect address before using these bits as either an effective or indirect address. The updated contents are written back into the auto-incrementing location.

Memory locations with addresses within the range  $030_8$  through  $037_8$  are called auto-decrementing locations. When one of these locations is indirectly addressed, the processor automatically decrements bits 0 through 15 of the contents of the indirect address before using these bits as either an effective or indirect address. The updated contents are written back into the auto-decrementing location.

**NOTE 1:** When referencing auto-increment/decrement locations, the state of bit 0 before the increment or decrement is the condition upon which the continuation of the indirection chain is based. For example: if an auto-increment location contains  $17777_8$  and the location is referenced as part of an indirection chain, location 0 will be the next indirect address in the chain; not the effective address.

**NOTE 2:** When a non-existing memory address is specified in a memory reference instruction, the contents will appear as all ones.

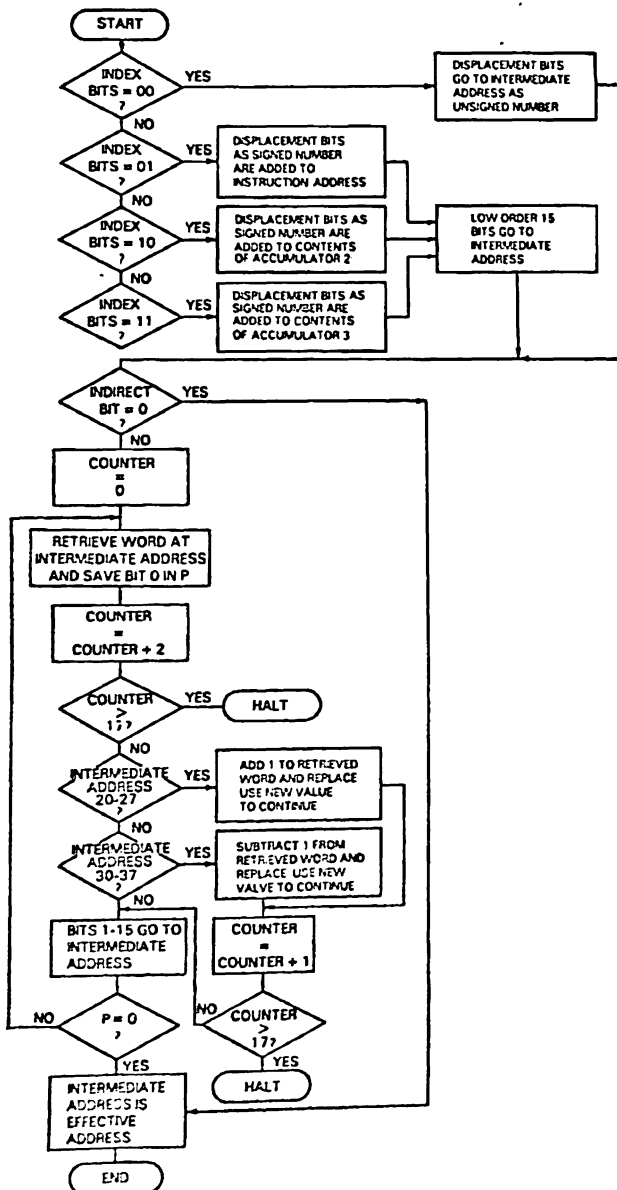
# mN601

## MEMORY REFERENCE INSTRUCTIONS

### MEMORY ADDRESSING

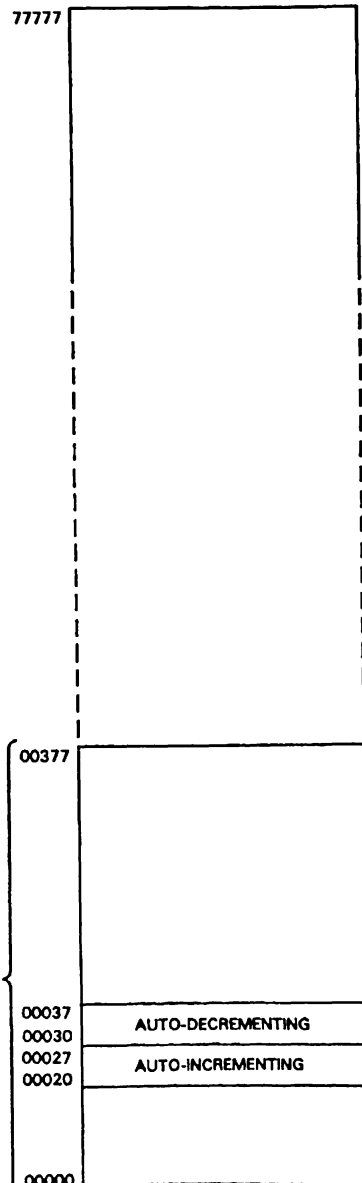
MODE	INDEX BITS	MEMORY ADDRESSING CAPABILITY (OCTAL)
Absolute	00	00000-00377
PC Relative	01	00000-77777
AC2 Indexed	10	00000-77777
AC3 Indexed	11	00000-77777

### ADDRESS CALCULATION



DG-06402

### MEMORY

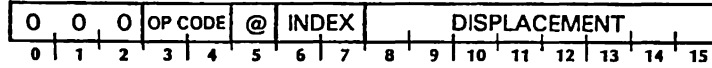


DG-04008

# mN601

## MEMORY REFERENCE INSTRUCTIONS

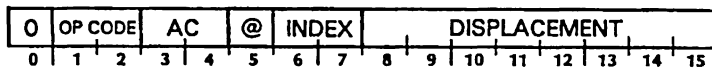
### No Accumulator-Effective Address Instructions



**NOTE:** The letter E in the function descriptions below denotes effective address; AC denotes accumulator.

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																																		
JMP	Jump	Compute (E) and load E into PC.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	0	0	0	0	0	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	@	INDEX	DISPLACEMENT																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						
JSR	Jump To Subroutine	Compute E. Store contents of PC+1 in bits 1-15 of AC3 and set bit 0 to 0; then, load E into PC.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">1</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	0	0	0	0	1	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	@	INDEX	DISPLACEMENT																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						
ISZ	Increment And Skip If Zero	Compute E. Increment contents of location E and write results back into location E. If results equal 000000, increment PC by two; otherwise increment PC by one.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">1</td><td style="width: 3%;">0</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	0	0	1	0	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	1	0	@	INDEX	DISPLACEMENT																														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						
DSZ	Decrement And Skip If Zero	Compute E. Decrement contents of location E and write results back into location E. If results equal 000000, increment PC by two; otherwise increment PC by one.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">1</td><td style="width: 3%;">1</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	0	0	1	1	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	1	1	@	INDEX	DISPLACEMENT																														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						

### One Accumulator-Effective Address Instructions



AC	Bits	
	3	4
AC0	0	0
AC1	0	1
AC2	1	0
AC3	1	1

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																																
LDA	Load Accumulator	Compute E. Load specified AC with contents of location E.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">0</td><td style="width: 3%;">1</td><td style="width: 3%;">AC</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	0	1	AC	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	AC	@	INDEX	DISPLACEMENT																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																				
STA	Store Accumulator	Compute E. Store contents of specified AC in location E.	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 3%;">0</td><td style="width: 3%;">1</td><td style="width: 3%;">0</td><td style="width: 3%;">AC</td><td style="width: 3%;">@</td><td style="width: 3%;">INDEX</td><td colspan="10" style="width: 50%;">DISPLACEMENT</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	0	AC	@	INDEX	DISPLACEMENT										0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	AC	@	INDEX	DISPLACEMENT																													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																				

# mN601

## TRAP INSTRUCTION

### TRAP INSTRUCTION

This instruction can be used as a single word call to subroutines containing extended instructions not implemented by the mN601. The format of the TRAP instruction is shown below.

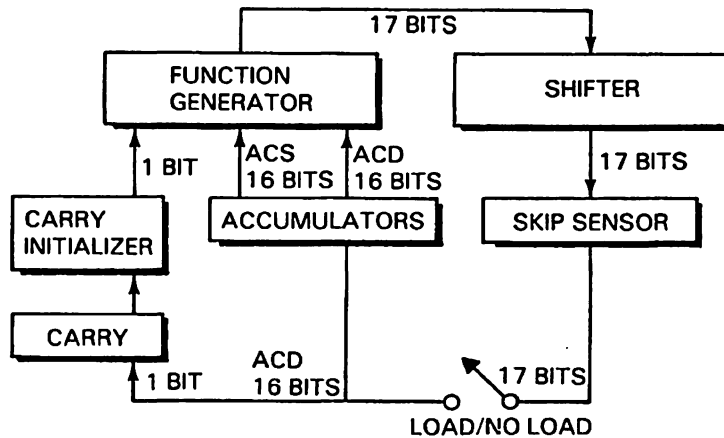
MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																								
TRAP	Trap	The address of this instruction is placed in memory location 46 <sub>g</sub> and bit 0 of that location is set to 0. Then, the processor performs a jump indirect through memory location 47 <sub>g</sub> . The contents of ACS and ACD do not affect this instruction.	<table border="1"><tr><td>1</td><td>ACS</td><td>ACD</td><td>TRAP NUMBER</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr></table>	1	ACS	ACD	TRAP NUMBER	1	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	ACS	ACD	TRAP NUMBER	1	0	0	0																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15												

## ARITHMETIC/LOGICAL INSTRUCTIONS

### ARITHMETIC/LOGICAL INSTRUCTIONS

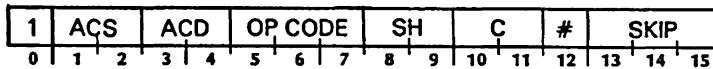
This instruction group performs arithmetic and logical operations on the contents of one or two accumulators. In some cases, it changes the value of the carry bit and, if specified, manipulates and/or tests the results of the operations.

The logical organization of the mN601's internal arithmetic logic unit (ALU) is shown below.



DC-00927

### TWO ACCUMULATOR-MULTIPLE OPERATION



AC = accumulator  
 ACS = source accumulator  
 ACD = destination accumulator

	ACS		ACD	
Bits	1	2	3	4
AC0	0	0	0	0
AC1	0	1	0	1
AC2	1	0	1	0
AC3	1	1	1	1

Each instruction specifies two accumulators, source and destination, that supply the function generator with operands. The function generator performs the operation specified by the operation code, e.g., add, subtract, and produces a carry bit whose value depends on three quantities: an initial value specified by the instruction, the inputs, and the function performed. The initial value can be derived either from the previous value of the carry bit or a new value specified by the instruction.

The 17-bit (16 bits plus the carry bit) result of the function generator goes to the shifter. Depending upon the shift operation specified, the results are either passed undisturbed through the shifter or they are manipulated in one of the following ways: the 17-bit result is rotated one bit position, either left or right; or, the two 8-bit halves are swapped without affecting the carry bit.

# mN601

## ARITHMETIC/LOGICAL INSTRUCTIONS

CODED CHARACTER	SHIFTER OPERATION
L	<p>Left rotate one place. Bit 0 is rotated into the carry position, the carry bit into bit 15.</p>
R	<p>Right rotate one place. Bit 15 is rotated into the carry position, the carry bit into bit 0.</p>
S	<p>Swap the halves of the 16-bit result. The carry bit is not affected.</p>

DC-04423

The 17-bit output of the shifter can be tested for a conditional skip. The skip sensor tests either, or both, the carry bit or the 16-bit result of the shifter, depending upon the instruction, to determine if either, or both, is equal or not equal to zero.

After the skip sensor has tested the output of the shifter, the 17-bit result of the shifter is either loaded or not loaded, depending upon the instruction, into the destination accumulator and the carry bit.

CLASS ABBREV.	CODED CHARACTER	RESULT BITS	OPERATION
C	(option omitted)	00	Do not initialize the carry bit
	Z	01	Initialize the carry bit to 0
	O	10	Initialize the carry bit to 1
	C	11	Initialize the carry bit to the complement of its present value
SH	(option omitted)	00	Leave the result of the arithmetic or logical operation unaffected
	L	01	Combine the carry and the 16-bit result into a 17-bit number and rotate it one bit left
	R	10	Combine the carry and the 16-bit result into a 17-bit number and rotate it one bit right
	S	11	Exchange the two 8-bit halves of the 16-bit result without affecting the carry
#	(option omitted)	0	Load the result of the shift operation into ACD
	#	1	Do not load the result of the shift operation into ACD

DC-04424

CLASS ABBREV.	CODED CHARACTER	RESULT BITS	OPERATION
SKIP	(option omitted)	000	Never skip
	SKP	001	Always skip
	SZC	010	Skip if carry = 0
	SNC	011	Skip if carry ≠ 0
	SZR	100	Skip if result = 0
	SNR	101	Skip if result ≠ 0
	SEZ	110	Skip if either carry or result = 0
	SBN	111	Skip if both carry and result ≠ 0

DC-04425

**NOTE:** Instructions in the Two Accumulator-Multiple Operation format must NOT have both the 'No Load' and the 'Never Skip' options specified at the same time. These bit combinations are used by other instructions in the instruction set.

## ARITHMETIC/LOGICAL INSTRUCTIONS

### Two Accumulator-Multiple Operation Instructions

In the function descriptions below, it is assumed that the carry bit is set to the specified value and the shift, conditional skip and load functions are performed as stipulated in the instruction field. When a conditional skip is specified in the instruction, the program counter is incremented an additional time if the condition is true.

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
COM	Complement	The one's complement of the contents of ACS is placed in the shifter. The output of the shifter is placed in ACD and the carry bit.	
NEG	Negate	The two's complement of the contents of ACS is placed in the shifter. If the operation produces a carry of 1, carry is complemented. The output of the shifter is placed in ACD and the carry bit.	
MOV	Move	Contents of ACS are placed in the shifter. The output of the shifter is placed in ACD and the carry bit.	
INC	Increment	Contents of ACS are incremented by one. If the operation produces a carry of 1, carry is complemented. The output of the shifter is placed in ACD and the carry bit.	
ADC	Add Complement	The one's complement of the contents of ACS is added to the contents of ACD. If the operation produces a carry of 1, carry is complemented. The output of the shifter is placed in ACD and the carry bit.	



# mN601

## ARITHMETIC/LOGICAL INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																										
SUB	Subtract	The two's complement of the contents of ACS is added to the contents of ACD. If the operation produces a carry of 1, carry is complemented. The output of the shifter is placed in ACD and the carry bit.	<table border="1"> <tr> <td>1</td> <td>ACS</td> <td>ACD</td> <td>1</td> <td>0</td> <td>1</td> <td>SH</td> <td>C</td> <td>#</td> <td>SKIP</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> </tr> </table>	1	ACS	ACD	1	0	1	SH	C	#	SKIP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	ACS	ACD	1	0	1	SH	C	#	SKIP																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15														
ADD	Add	The contents of ACS are added to the contents of ACD. If the operation produces a carry of 1, the carry is complemented. The output of the shifter is placed in ACD and the carry bit.	<table border="1"> <tr> <td>1</td> <td>ACS</td> <td>ACD</td> <td>1</td> <td>1</td> <td>0</td> <td>SH</td> <td>C</td> <td>#</td> <td>SKIP</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> </tr> </table>	1	ACS	ACD	1	1	0	SH	C	#	SKIP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	ACS	ACD	1	1	0	SH	C	#	SKIP																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15														
AND	And	The contents of ACS and ACD are logically AND'd. The output of the shifter is placed in ACD and the carry bit.	<table border="1"> <tr> <td>1</td> <td>ACS</td> <td>ACD</td> <td>1</td> <td>1</td> <td>1</td> <td>SH</td> <td>C</td> <td>#</td> <td>SKIP</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> </tr> </table>	1	ACS	ACD	1	1	1	SH	C	#	SKIP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	ACS	ACD	1	1	1	SH	C	#	SKIP																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15														

### Multiply/Divide Instructions

The following instructions utilize an I/O instruction format and bits 10 to 15 specify device code 1. For this reason, device code 1 is an illegal device code and should not be assigned to any I/O interface device.

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
MUL	Multiply	<p>The contents of AC1 are multiplied by the contents of AC2, yielding a 32-bit intermediate result. The intermediate result is added to the contents of AC0, yielding a final 32-bit result. The high order bits (0-15) of the result are loaded into AC0 and the low order bits are loaded into AC1. Bit 0 of AC0 contains the high-order bit of the result; bit 15 of AC1 contains the low-order bit. The contents of AC2 are unchanged.</p>	<pre> 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15                     </pre>
DIV	Divide	<p>The 32-bit contents of AC0 and AC1 are divided by the contents of AC2. Bit 0 of AC0 is the high-order bit of the dividend; bit 15 of AC1 is the low-order bit. The resulting quotient is placed in AC1 and the remainder in AC0.</p> <p>NOTE: Before the divide operation is performed, the contents of AC0 are compared to the contents of AC2. If the contents of AC0 are greater than AC2, an overflow condition is indicated. In this case, the carry bit is set to 1 and the operation is terminated; otherwise, the carry bit is set to 0.</p>	<pre> 0 1 1 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15                     </pre>



# mN601

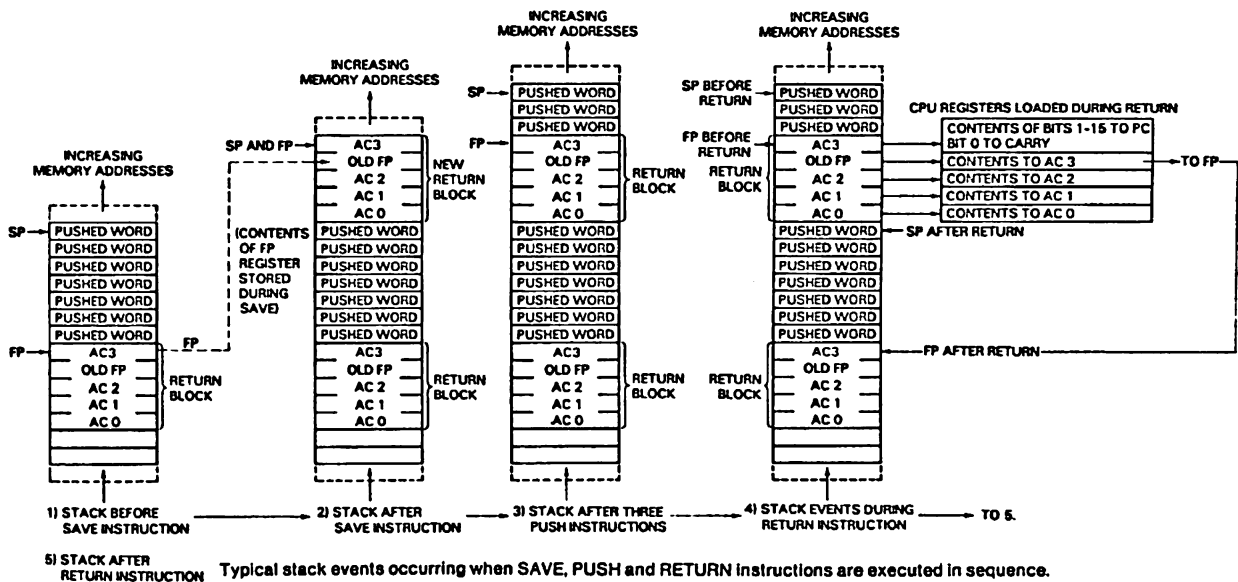
## STACK INSTRUCTIONS

Two instructions, SAVE and RETURN, move information between a return block in the stack and the CPU registers described below. When a SAVE instruction is executed, the contents of the registers are stored in the return block addressed by the SP (see below). When a RETURN instruction is executed, the contents of the return block addressed by the FP are loaded into the appropriate registers (see below).

Return blocks are intended to be used when entering into and exiting from a subroutine. To implement this subroutine technique, the SAVE instruction is executed as the first instruction of a subroutine called by a JSR (JUMP TO SUBROUTINE) instruction (see Memory Reference Instructions). A JSR instruction stores the return address (contents of program counter plus one) in AC3. Then, the SAVE instruction stores the contents of AC3 plus the carry bit in the last word pushed on the stack. Also, SAVE stores the contents of AC0, AC1, AC2, and FP in the stack. After this information is pushed on the stack, the SP and the FP are updated to their new values, and AC3 contains the new FP.

The RETURN instruction is the last instruction of the called subroutine. When RETURN is executed, the return address is automatically loaded into the program counter (PC); the carry bit, AC0, AC1, AC2, FP and SP are restored to their original values before the execution of the SAVE instruction; and the restored value of FP is left in AC3. See the illustration below.

### SAVE/RETURN



Typical stack events occurring when SAVE, PUSH and RETURN instructions are executed in sequence.

**NOTE:** The SP register contains the address of the top memory location in the stack. Thus, the three single variables pushed on the stack after the last return block (see Figure 4) are effectively pushed off the stack when the RETURN instruction is performed. These variables could be retained elsewhere in memory by executing a series of three POP instructions, each followed by a STA (STORE ACCUMULATOR, see Memory Reference Instructions) instruction prior to executing the RETURN instruction. Then, if desired, the variables can be pushed back on the stack after the execution of the RETURN instruction.

See Figures 4 and 5. After the RETURN instruction was executed, the contents of AC0, AC1, AC2, PC, carry and the SP and FP registers were restored to their original values before the SAVE instruction was executed. The new value of the FP register is left in AC3 after the RETURN instruction.

For ease of programming, there are four additional instructions which affect the contents of the stack and frame pointers: MTSP, MTFP, MFSP, and MFFP. When an MTSP (MOVE TO STACK POINTER) instruction is executed, the contents of the accumulator specified in the instruction are loaded into the SP; and the contents of the accumulator are unchanged. When an MFSP (MOVE FROM STACK POINTER) is executed, the contents of the SP are loaded into the accumulator specified in the instruction; and, the contents of the SP are unchanged. The MTFP (MOVE TO FRAME POINTER) and MFFP (MOVE FROM FRAME POINTER) instructions perform a similar function, moving information between the FP and the accumulator specified in the instruction.

During every instruction that pushes data onto the stack, a check is made for a stack overflow. A stack overflow exists when the stack pointer is incremented across a 256-word address boundary; e.g., xx377<sub>h</sub> to xx400<sub>h</sub>. When this occurs, the instruction is completed and a stack overflow interrupt is generated, providing CPU interrupts are enabled. CPU interrupts are explained in the CPU I/O instructions.

Stack Instructions utilize an I/O instruction format and bits 10 to 15 specify device code 1. For this reason, device code 1 is an illegal device code and should not be assigned to any I/O interface device (see I/O Operations section).

# mN601

## STACK INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
PSHA	Push	The SP is incremented by 1; then, the contents of the specified AC are loaded into the memory location addressed by the contents of SP. The contents of the specified AC are unchanged. If the contents of bits 8-15 of SP are 000 (a 256-word boundary has been crossed), a stack overflow interrupt request is generated.	
POPA	Pop	The specified AC is loaded with the contents of the memory location addressed by SP; then, SP is decremented by 1.	
MTSP	Move To Stack Pointer	The contents of bits 1-15 of the specified AC are placed in SP. The contents of the specified AC are unchanged.	
MTFP	Move To Frame Pointer	The contents of bits 1-15 of the specified AC are placed in FP. The contents of the specified AC are unchanged.	
MFSP	Move From Stack Pointer	The contents of SP are placed in bits 1-15 of the specified AC and bit 0 of the specified AC is set to 0. The contents of the SP are unchanged.	
MFFP	Move From Frame Pointer	The contents of FP are placed in bits 1-15 of the specified AC and bit 0 of the specified AC is set to 0. The contents of FP are unchanged.	

# mN601

## STACK INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																																
SAV	Save	<p>Information is pushed on stack as follows:</p> <ol style="list-style-type: none"> <li>1) the SP is incremented by 1; then, the contents of AC0 are stored in the memory location addressed by the contents of SP;</li> <li>2) the SP is incremented by 1; then, the contents of AC1 are stored in the memory location addressed by the contents of SP;</li> <li>3) the SP is incremented by 1; then, the contents of AC2 are stored in the memory location addressed by the contents of SP;</li> <li>4) the SP is incremented by 1; then, the contents of FP (before the SAVE instruction) are stored in bits 1-15 of the memory location addressed by the contents of SP and 0 is written into bit 0.</li> <li>5) the SP is incremented by 1; then, the content of the carry bit is stored in bit 0 and the contents of bits 1-15 of AC3 are stored in bits 1-15 of the memory location addressed by the contents of SP.</li> </ol> <p>If the contents of bits 8-15 of SP are 000 (a 256-word boundary was crossed) at any time during execution of this instruction, a stack overflow interrupt request is generated.</p> <p>At the end of the instruction, the new contents of SP are loaded in both the FP and bits 1-15 of AC3. Bit 0 of AC3 is set to 0.</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td> </tr> <tr> <td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">2</td><td style="padding: 2px;">3</td><td style="padding: 2px;">4</td><td style="padding: 2px;">5</td><td style="padding: 2px;">6</td><td style="padding: 2px;">7</td><td style="padding: 2px;">8</td><td style="padding: 2px;">9</td><td style="padding: 2px;">10</td><td style="padding: 2px;">11</td><td style="padding: 2px;">12</td><td style="padding: 2px;">13</td><td style="padding: 2px;">14</td><td style="padding: 2px;">15</td> </tr> </table> </div>	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																				



# mN601

## STACK INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																																
RET	Return	<p>Information is popped off the stack as follows:</p> <ol style="list-style-type: none"> <li>1) the contents of the FP are loaded into the SP;</li> <li>2) bit 0 of the memory location addressed by the contents of SP is loaded into the carry bit and bits 1-15 are loaded into the PC;</li> <li>3) the SP is decremented by 1; then, the contents of the location addressed by the contents of SP are loaded into AC3;</li> <li>4) the SP is decremented by 1; then, the contents of the location addressed by the contents of SP are loaded into AC2;</li> <li>5) the SP is decremented by 1; then, the contents of the location addressed by the contents of SP are loaded into AC1;</li> <li>6) the SP is decremented by 1; then, the contents of the location addressed by the contents of SP are loaded into AC0.</li> <li>7) the SP is decremented by 1.</li> </ol> <p>At the end of the instruction, the new contents of AC3 (bits 1-15) are loaded into FP.</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table> </div>	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	1																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																				

1

# mN601

## INPUT/OUTPUT INSTRUCTIONS

### INPUT/OUTPUT INSTRUCTIONS

This instruction group is used by the mN601 to communicate with I/O devices. It passes information between one of the mN601's four accumulators and the I/O bus and specifies one of three input and one of three output buffers (designated as A, B, and C) in an I/O controller connected to the bus.

I/O instructions accommodate a 6-bit device selection network which can address 64 separate device codes. When devices are interfaced to the mN601's I/O bus via an mN603 I/O Controller, each device can be connected to the bus in such a way that it will only respond to commands containing its own device code.

### Busy and Done Flags

It is assumed that each device has a set of flags, Busy and Done, that control its operation (see mN603). In general, these flags are used in the following manner. When they are both zero, the device is idle. To start a device, the program sets Busy to one and Done to zero. When an operation is finished, the device sets Done to one and Busy to zero. At this time, the device can issue a program interrupt request (see I/O Operations section.)

### Addressing Non-Existent Device Codes

If an attempt is made to issue an I/O instruction to a non-existent device, the mN601 functions as though the device exists and no indication is given that it does not exist. If an attempt is made to test the status of the Busy and Done flags (see below) of a non-existent device, the mN601 behaves as though the Busy and Done flag bits are 0. If a data in instruction is directed to a non-existent device, the specified accumulator will contain all ones. See functional description of I/O instructions.

The tables below illustrate the manner in which an I/O instruction is coded to set flags and how an I/O SKIP instruction is coded.

### FLAG SETTING

CLASS ABBREV.	CODED CHARACTER	RESULT BITS	OPERATION
f	(option omitted)	00	Does not affect the Busy and Done flags.
	S	01	Start the device by setting Busy to 1 and Done to 0.
	C	10	Idle the device by setting both Busy and Done to 0.
	P	11	The effect, if any, depends on the device.

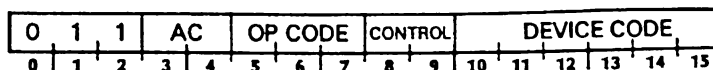
DG-04421

### FLAG TESTING

CLASS ABBREV.	CODED CHARACTER	RESULT BITS	OPERATION
t	BN	00	Tests for Busy = 1
	BZ	01	Tests for Busy = 0
	DN	10	Tests for Done = 1
	DZ	11	Tests for Done = 0

DG-04422

The format of an I/O instruction is shown below. The number of data bits moved depends upon the size of the buffer and the operating mode of the device.



**NOTE:** The control bits, 8 and 9, can be f or t depending upon the type of instruction.



# mN601

## INPUT/OUTPUT INSTRUCTIONS

When the instructions described below are executed, the CPU sends the I/O instruction, exactly as it is stored in memory, to the I/O bus. Following the instruction, data is transferred either from the CPU or the I/O controller. See I/O Operations section.

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																															
NIO	No I/O Transfer	The Busy and Done flags of the addressed device are set according to the convention specified in bits 8-9 of the instruction.  The data transferred by the CPU is meaningless and is ignored by the I/O controller.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	0	0	0	0	0	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	0	0	0	0	F	DEVICE CODE																									
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DIA	Data In A	Following the receipt of this instruction, the addressed I/O controller is expected to send a 16-bit word to the I/O bus from its A buffer. This word is retrieved from the I/O bus by the CPU and placed in the accumulator specified in bits 3-4 of the instruction. After the data transfer, the Busy and Done flags are set according to the convention specified in bits 8-9 of the instruction.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>0</td><td>0</td><td>1</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	0	0	1	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	0	0	1	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DIB	Data In B	Same as DIA, except B buffer.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>0</td><td>1</td><td>1</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	0	1	1	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	0	1	1	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DIC	Data In C	Same as DIA, except C buffer.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>1</td><td>0</td><td>1</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	1	0	1	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	1	0	1	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DOA	Data Out A	Following the receipt of this instruction, the addressed I/O controller should retrieve the next 16 bits of data appearing on the I/O bus and load them into its A buffer. The data bits are the contents of the accumulator specified in bits 3-4 of the instruction. After the data transfer, the Busy and Done flags are set according to the convention specified in bits 8-9 of the instruction.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>0</td><td>1</td><td>0</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	0	1	0	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	0	1	0	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DOB	Data Out B	Same as DOA, except B buffer.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>1</td><td>0</td><td>0</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	1	0	0	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	1	0	0	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
DOC	Data Out C	Same as DOA, except C buffer.	<table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>AC</td><td>1</td><td>1</td><td>0</td><td>F</td><td colspan="6">DEVICE CODE</td> </tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> </table>	0	1	1	AC	1	1	0	F	DEVICE CODE						0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	AC	1	1	0	F	DEVICE CODE																										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			

# mN601

## INPUT/OUTPUT INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
SKP	I/O Skip	<p>Following the receipt of this instruction, the addressed I/O controller sends the state, 0 or 1, of its Busy and Done flags to the I/O bus. This information is contained in a 16-bit data word. Data bit 0 contains the complement of the state of the Done flag. Data bit 1 contains the complement of the state of the Busy flag. The CPU retrieves this information from the I/O bus and ignores the contents of bits 2-15. It takes the appropriate action depending upon the content of the Busy and Done bits and the specified test condition contained in bits 8-9 of the instruction. If the test condition is true, the program counter is incremented an additional time.</p>	<pre> 0 1 1 0 0 1 1 1 T  DEVICE CODE 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15           </pre>



# mN601

## PROGRAM INTERRUPTS

### PROGRAM INTERRUPTS

Program interrupts are sequences of operations during which the CPU writes the contents of the program counter into a specified memory location and loads the program counter with the address of another program. In the mN601, interrupts are performed at the request of a peripheral device, the real-time clock or the occurrence of a stack overflow condition.

The mN601 responds to program interrupts only when the interrupt enable flag contains a 1 and the CPU is either in the halted state or about to fetch an instruction. It performs an interrupt by incrementing the program counter, storing the updated contents of the program counter in memory location 0, performing a jump indirect through the appropriate memory location indicated below, and clearing the interrupt enable flag. When either a real-time clock or stack overflow interrupt request is honored, the corresponding request flag is cleared.

Type	Memory Location
External (I/O devices)	00001
CPU real-time clock	00002
Stack overflow	00003

When more than one type of interrupt is generated simultaneously, they are serviced according to the following priority scheme.

Type	Priority
Stack overflow	first
CPU real-time clock	second
External (I/O devices)	third

### 16-Level Priority Interrupts

The mN601 provides sixteen levels of interrupts. These levels are established by the program, using a 16-bit word mask together with a CPU mask instruction. (See CPU I/O instructions.) Each bit in the mask (0-15) is assigned to a particular device, or devices operating at similar data transmission rates. To disable a device from generating interrupts, a 1 is placed in that device's mask bit; to enable the device, a 0 is placed in the mask bit. To use the mask, each I/O controller must contain an interrupt Disable flag that is set by the CPU mask instruction to enable or disable interrupts.

# mN601

## CPU I/O INSTRUCTIONS

### CPU I/O INSTRUCTIONS

I/O instructions addressed to device code 77<sub>h</sub> are CPU I/O instructions. This instruction group performs special functions rather than controlling specific devices. Some CPU I/O instructions are global in nature and are directed to all I/O controllers; others are acted upon internally by the CPU.

In all but the I/O Skip instruction, I/O instructions addressed to device code 77<sub>h</sub> use bits 8-9 to control the condition of the Interrupt On flag (also referred to in the text as the interrupt enable flag). The table below shows the results of these bits when used with device code 77<sub>h</sub>.

CLASS ABBREV.	CODED CHARACTER	RESULT BITS	OPERATION
f	(option omitted)	00	Does not affect the state of the Interrupt On flag.
	S	01	Set the Interrupt On flag to 1.
	C	10	Set the Interrupt On flag to 0.
	P	11	Does not affect the state of the Interrupt On flag
t	BN	00	Tests for interrupt On= 1.
	BZ	01	Tests for interrupt On=0.

DG-04480

# mN601

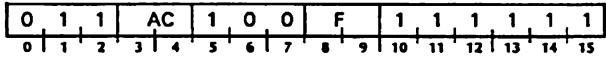
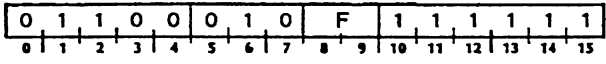
## CPU I/O INSTRUCTIONS

When the instructions described below are executed, the CPU sends the instruction, exactly as it is stored in memory, to the bus. In most cases, after the instruction is transmitted, data is transferred either from the CPU or the I/O controller. See the Operations section.

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
INTEN	Interrupt Enable	<p>This instruction is internal to the CPU. When executed, the Interrupt On flag is set to 1 after one more instruction is executed; then interrupt requests are honored by the CPU.</p> <p>The data transmitted by the CPU following this instruction should be ignored by the I/O controllers.</p>	
INTDS	Interrupt Disable	<p>This instruction is internal to the CPU. When executed, the Interrupt On flag is set to 0 and interrupt requests are not honored by the CPU.</p> <p>The data transmitted by the CPU following this instruction should be ignored by the I/O controllers.</p>	
INTA	Interrupt Acknowledge	<p>This is a global instruction directed to all I/O controllers. It is sent to the I/O bus in response to a program interrupt request. The highest priority device requesting an interrupt should respond by sending its device code to the I/O bus in a 16-bit data word. The device code is contained in data bits 10-15. When the CPU retrieves this information, the device code is placed in bits 10-15 of the accumulator specified in bits 3-4 of the instruction. Data bits 0-9 are transmitted and loaded in the specified AC as 0's.</p> <p>If no device is requesting an interrupt, bits 0-15 of the specified accumulator are set to ones.</p> <p>The Interrupt On flag is set according to the convention specified in bits 8-9 of the instruction.</p>	

# mN601

## CPU I/O INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
MSKO	Mask Out	<p>This is a global instruction directed to all I/O controllers. After the CPU sends the instruction, it sends the contents of the accumulator specified in bits 3-4 of the instruction. The contents of the accumulator are interpreted by the I/O controllers as a 16-bit software priority mask.</p> <p>Each I/O controller responds by turning its Interrupt Disable flag on or off according to the device's selected priority level. A 1 in a device's mask bit turns its Interrupt Disable flag on; a 0 turns it off.</p> <p>The Interrupt On flag is set according to the convention specified in bits 8-9 of the instruction.</p>	 <pre> 0 1 1   AC   1 0 0   F   1 1 1 1 1 1 0 1 2   3 4   5 6 7   8   9 10 11 12 13 14 15 </pre>
IORST	Reset	<p>This is a global instruction directed to all I/O controllers and the CPU.</p> <p>The I/O controllers respond by setting their Busy and Done flags to 0 and by turning their Interrupt Disable flags off.</p> <p>The CPU responds by disabling its real-time clock from generating interrupts and by setting its Interrupt On flag according to the convention specified in bits 8-9 of the instruction.</p> <p>The data transferred by the CPU following this instruction should be ignored by the I/O controllers.</p>	 <pre> 0 1 1 0 0   0 1 0   F   1 1 1 1 1 1 0 1 2 3 4   5 6 7   8   9 10 11 12 13 14 15 </pre>

# mN601

## CPU I/O INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT
HALT	Halt	<p>This instruction is internal to the CPU. It is used to stop the execution of instructions.</p> <p>The Interrupt On flag is set to 1. The CPU honors both program interrupt and data channel requests while it is in the halted state.</p> <p>When this instruction is executed, no data is transferred as part of the I/O transaction. See I/O Operations.</p>	<pre> 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15                     </pre>
SKP	CPU Skip	<p>This instruction is internal to the CPU. When executed, the CPU switches to I/O input mode (see I/O Operations section); but, no device should respond by placing data on the I/O bus.</p> <p>CPU Skip tests the state of the Interrupt On flag. If the test condition specified in bits 8-9 of the instruction is true, the program counter is incremented an additional time.</p>	<pre> 0 1 1 0 0 1 1 1 T 1 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15                     </pre>
RTCEN	Real-time Clock Enable	<p>This instruction is internal to the CPU. It enables the CPU's real-time clock to generate program interrupt requests.</p> <p>The Interrupt On flag is set according to the convention specified in bits 8-9 of the instruction.</p> <p>The data transmitted by the CPU following this instruction should be ignored by I/O controllers.</p>	<pre> 0 1 1 1 0 0 1 0 F 1 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15                     </pre>

# mN601

## CPU I/O INSTRUCTIONS

MNEMONIC	MEANING	FUNCTION	INSTRUCTION FORMAT																															
RTCDS	Real-time Clock Disable	<p>This instruction is internal to the CPU. It disables program interrupt requests by the CPU'S real-time clock.</p> <p>The Interrupt On flag is set according to the convention specified in bits 8-9 of the instruction.</p> <p>The data transmitted by the CPU following this instruction should be ignored by I/O controllers.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <table style="border-collapse: collapse; text-align: center;"> <tr> <td style="border: 1px solid black; padding: 2px 5px;">0</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">0</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">0</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">0</td> <td style="border: 1px solid black; padding: 2px 5px;">F</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> </tr> <tr> <td style="border: none; padding: 0 5px;">0</td> <td style="border: none; padding: 0 5px;">1</td> <td style="border: none; padding: 0 5px;">2</td> <td style="border: none; padding: 0 5px;">3</td> <td style="border: none; padding: 0 5px;">4</td> <td style="border: none; padding: 0 5px;">5</td> <td style="border: none; padding: 0 5px;">6</td> <td style="border: none; padding: 0 5px;">7</td> <td style="border: none; padding: 0 5px;">8</td> <td style="border: none; padding: 0 5px;">9</td> <td style="border: none; padding: 0 5px;">10</td> <td style="border: none; padding: 0 5px;">11</td> <td style="border: none; padding: 0 5px;">12</td> <td style="border: none; padding: 0 5px;">13</td> <td style="border: none; padding: 0 5px;">14</td> <td style="border: none; padding: 0 5px;">15</td> </tr> </table> </div>	0	1	1	0	1	0	1	0	F	1	1	1	1	1	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	0	1	0	1	0	F	1	1	1	1	1	1																				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			

72



# mN601

## OPERATIONS PROTOCOL

### OPERATIONS PROTOCOL

The mN601 communicates with memory elements and I/O interfaces via its memory and I/O data buses and their associated control signals. The protocol used by the mN601 in transmitting and receiving information via its pins is described below.

#### MEMORY OPERATIONS

The mN601 communicates with memory via the address/data pins, MB <0-15>, and three memory control signals, P, SAE and WE. Using these pins, it controls four kinds of memory operations, read, write, read-modify-write and refresh.

During read, write and read-modify-write operations, addresses and data are transferred between the pins and memory. During refresh operations, the CPU specifies 1/64 of the available memory locations to be refreshed by transferring a 6-bit refresh address to memory together with the appropriate control signals. MB <10-15> carry the refresh block address and MB <0-9> contain 1's.

#### Address/Data Pins

The address/data pins, referred to as the memory address/data bus, are precharged to a logic high state for a period of 120ns every 480ns, regardless of the operation being performed by the CPU. Information is placed on the bus by discharging those lines which carry zeroes and leaving undischarged those lines which carry ones. Once the lines are discharged during a read or write operation, or by external devices, they cannot be used again until they are precharged.

#### Control

Prior to initiating a memory operation, the CPU transmits the memory address via MB <1-15>, and specifies the type of memory operation by transmitting MBO in either the high or low state. When MBO is at a high logic level, a refresh operation is selected. When it is at a low logic level, a read or write operation is selected.

A memory operation is initiated on the rising edge of P. At this time, the memory samples the address on the address/data bus and tests the state of MBO.

**NOTE:** The memory operations timing diagrams depicted on the following pages show the  $\phi 2$  clock as a reference. Each action occurring on the mN601 output pins is synchronized to the rising or falling edge of a clock.

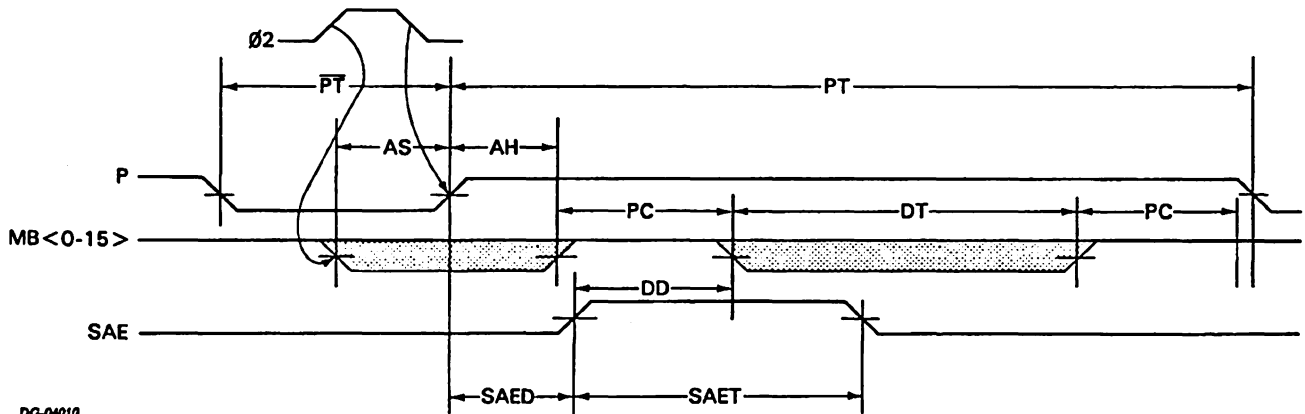
# mN601

## MEMORY OPERATIONS

### Read

After the memory operation is initiated, the CPU transmits the SAE (SENSE AMP ENABLE) signal. After this signal goes to the high state, information is read from memory and placed on the address/data bus, and the CPU retrieves the information via the MB <0-15> pins.

### READ TIMING DIAGRAM



DG-04010

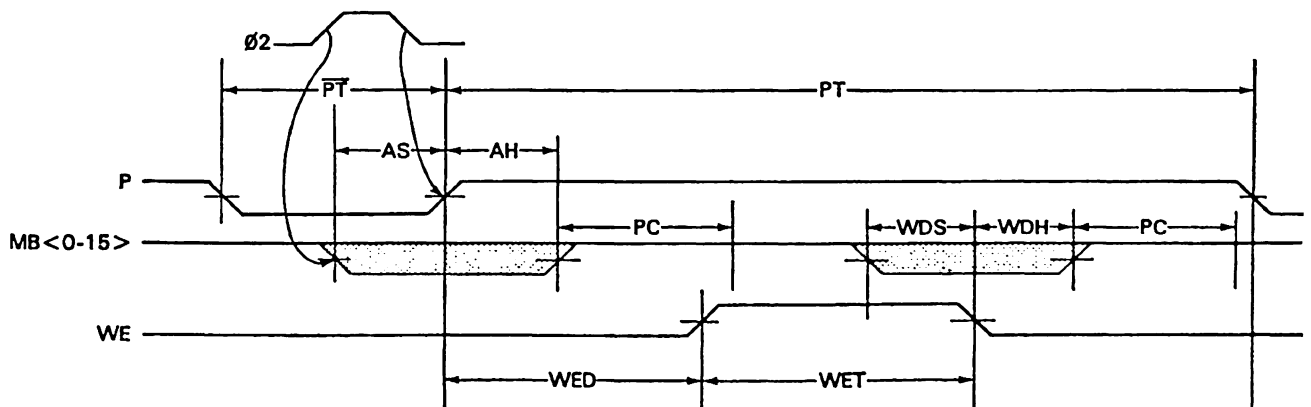
#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

### Write

After the memory operation is initiated, the CPU transmits both the data to be written into memory via MB <0-15> and the WE (WRITE ENABLE) signal. When WE goes from the high to the low logic level, the information on the address/data bus is valid and is written into memory.

### WRITE TIMING DIAGRAM



DG-04011

#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

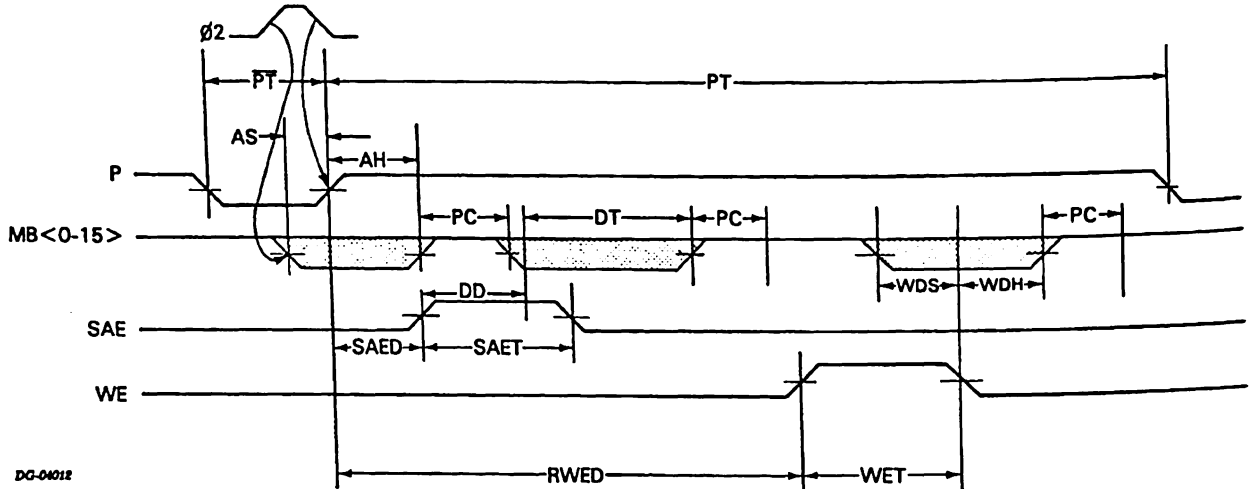
# mN601

## MEMORY OPERATIONS

### Read-Modify-Write

After the memory operation is initiated, a read operation is performed. Then, the address/data bus is precharged and a write operation follows. P remains in the high state until after the memory operation is completed.

READ-MODIFY-WRITE TIMING DIAGRAM



DG-04012

#### NOTES

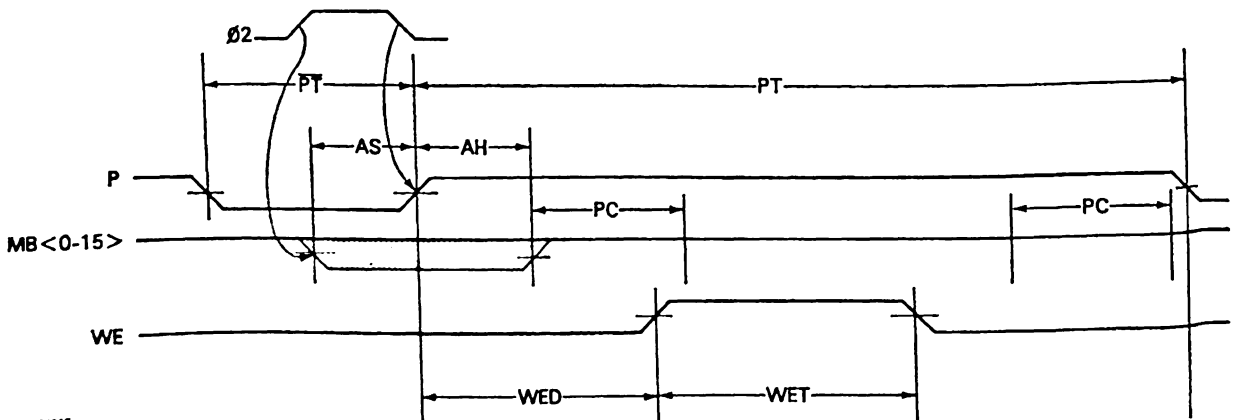
- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

### Refresh

After the refresh operation is initiated, the CPU transmits the WE signal, but does not transmit data, and the information selected by the address is refreshed.

The mN601 supports dynamic RAM elements by refreshing all memory locations at least once every 1.8432ms.

REFRESH TIMING DIAGRAM



DG-04013

#### NOTES

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

**NOTE:** At times, a memory operation is initiated and then aborted by the CPU. This condition arises when the CPU makes an "intelligent decision" while performing a task; for example, during indirect addressing chains. When this occurs, the memory control signal P is asserted and the address appears on the memory bus; but, no memory operation is performed.

# mN601

## I/O OPERATIONS

### I/O OPERATIONS

Data is passed between the mN601 and the mN603 I/O controllers connected to the microNOVA I/O bus under two operating modes: programmed I/O and data channel break. The CPU is switched to one of these operating modes at the request of the I/O controllers, through the use of the program interrupt and data channel request facilities.

#### Programmed I/O

In programmed I/O, the CPU moves information, consisting of a command followed by data, between its accumulators and the I/O bus. There is one exception to this rule; that is, when an I/O Skip Instruction is executed. The data received in response to this command contains the state, 0 or 1, of the addressed device's Busy and Done flags. This information is not loaded into an accumulator. It is acted upon directly by the CPU. Using this information together with the command (e.g., SKPDN - Skip If Done Is Non-Zero), the CPU determines whether the test condition specified in the command is true or false. If it is true, the program counter is incremented an additional time.

#### Data Channel Break

In a data channel break, the CPU moves information between the I/O bus and memory without altering the program flow. This information transfer is transparent to the executing program. When the CPU honors a data channel request, it sends an acknowledgement to the I/O bus. This is responded to by the I/O controller with the highest priority requesting a data channel break. The I/O controller sends to the CPU the appropriate 15-bit memory address and a direction of transfer mode bit (0 equals data out; 1 equals data in). All references to direction of transfer are always in relation to the CPU; that is, data in to the CPU, data out from the CPU. When the CPU receives the address, it begins a read-modify-write operation using the address received from the I/O controller. Then, it examines the mode bit.

If the mode bit is set for a data out, the information received by the CPU during the read portion of the memory operation is sent to the I/O bus. The write portion of the memory operation is not executed.

If the mode bit is set for a data in, the CPU switches to I/O input mode and then waits a certain period of time to receive the data. The 16 data bits received from the I/O controller during this time period are sent to the memory bus and are written into memory during the write portion of the memory operation being performed. The information received by the CPU during the read portion of the memory operation is ignored.

#### I/O Bus Transfers

Information is passed in serial form between the CPU and the I/O bus via the CPU I/O bus pins. The pins and their functions are listed below.

PIN MNEMONIC	FUNCTION
I/O DATA 1	Bi-directional, data bus. This line carries the 8 most significant bits (0-7) of a 16-bit word in serial form, plus one control code prefix bit. (Code bits are explained below.)
I/O DATA 2	Bi-directional data bus. This line carries the 8 least significant bits (8-15) of a 16-bit word in serial form, plus one control code prefix bit.
I/O CLOCK	Bi-directional, synchronizing clock line. The I/O CLOCK strobes data into the receiving device at a rate of 8.33Mbits per line.
$\overline{\text{DCHINT}}$	This is a wired-OR of all I/O controllers' data channel request lines. When asserted low by any of the I/O controllers, it tells the CPU that a device, or devices, is requesting a data channel break.
$\overline{\text{EXTINT}}$	This is a wired-OR of all I/O controllers' program interrupt lines. When asserted low by any of the I/O controllers, it tells the CPU that a device, or devices, is requesting a program interrupt.
I/O INPUT	This line indicates the operating mode of the mN601 to the CPU I/O transceiver connected to the I/O bus. When it is at the low logic level, it tells the transceiver that the CPU is sending information, and the transceiver should receive the information and pass it to the I/O bus. When the line is at a high logic level, it tells the transceiver that the CPU is ready to receive data, and the information on the I/O bus should be retrieved and sent to the CPU. Except for the intervals during which the CPU is transmitting information, this line remains in the high logic state.

# mN601

## I/O OPERATIONS

The I/O data lines (I/O DATA<1-2>) are used to transfer four types of information: Request Enable, Data Channel Address Request, programmed I/O commands, and data. Each type consists of either 2 bits or 18 bits (1 or 9 bits on each I/O data line) and is identified by the first bit transmitted on each line. The encoding is shown below.

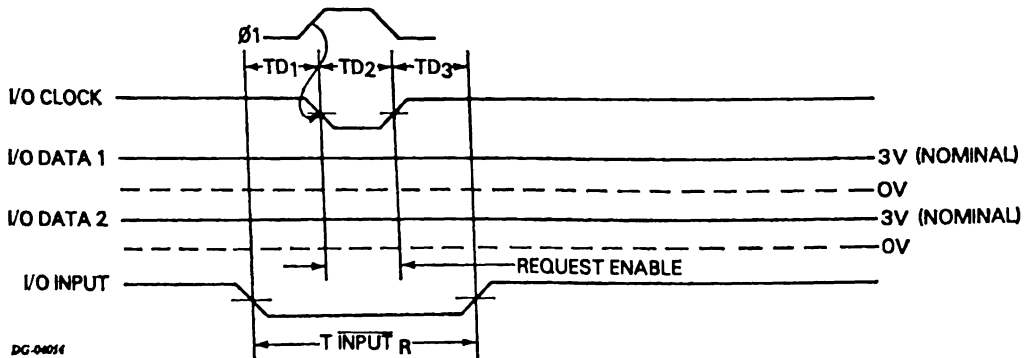
FIRST BIT CODE I/O DATA 1	FIRST BIT CODE I/O DATA 2	I/O TRANSFER TYPE	TOTAL NO. OF BITS TRANSFERRED
1	1	Request Enable	2 code bits only
1	0	Data Channel Address Request	2 code bits only
0	1	Data	18 bits (2 code bits plus 16 data bits)
0	0	I/O Command	18 bits (2 code bits plus 16 command bits)

**NOTE:** The I/O operations timing diagrams depicted below show the  $\phi 1$  clock as a reference. Each action occurring on the mN601 output pins is synchronized to the rising or falling edge of a clock.

### Request Enable

This is a signal sent by the CPU, in code form, to all devices connected to the I/O bus. It is sent at irregular intervals and is used by the individual I/O controllers to synchronize the assertion of program interrupt and data channel request lines. Request Enable ensures that the EXTINT, DCHINT, and priority lines are at a steady state when they are examined by the CPU.

REQUEST ENABLE TIMING DIAGRAM



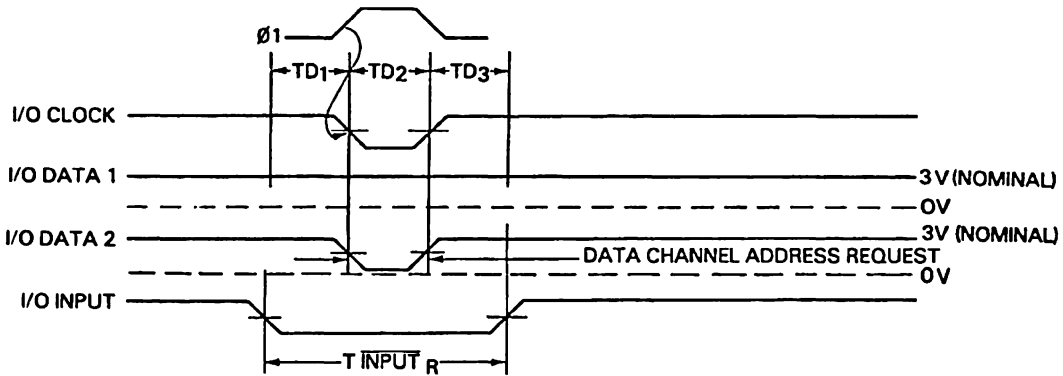
#### NOTES

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

### Data Channel Address Request

This is a signal sent by the CPU, in code form, to the I/O bus in response to a data channel request. It tells the I/O controllers that the CPU is ready to receive the data channel address and mode bit. It is responded to by the highest priority device requesting a data channel break. Data Channel Address Request also acts as a Request Enable.

**DATA CHANNEL ADDRESS REQUEST TIMING DIAGRAM**



DG-04015

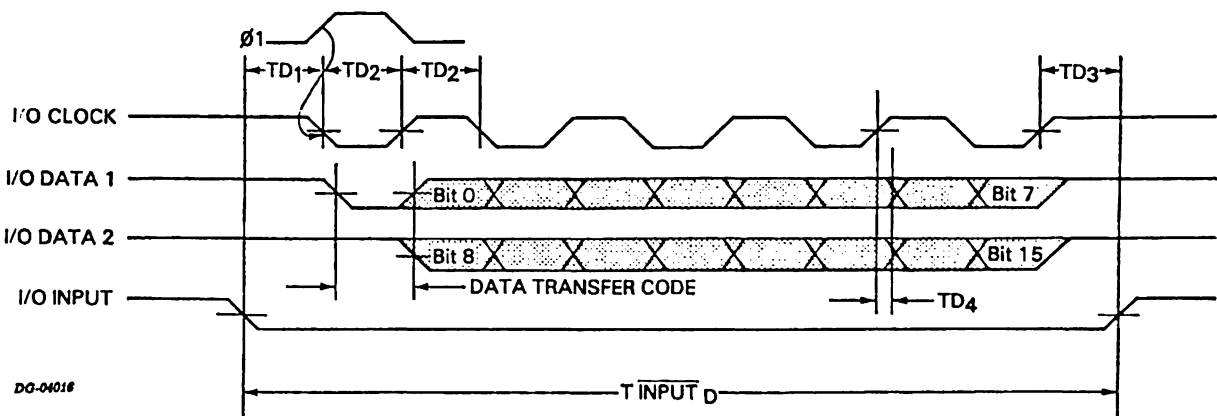
**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

### Data

The format shown in the Data Timing Diagram is used to transfer 16 data bits on the I/O bus. In programmed I/O, data is any 16-bit transfer which is not an I/O command. In a data channel break, this format is used to transfer the 15-bit memory address plus mode bit and the 16-bit word.

**DATA TIMING DIAGRAM**



DG-04016

**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

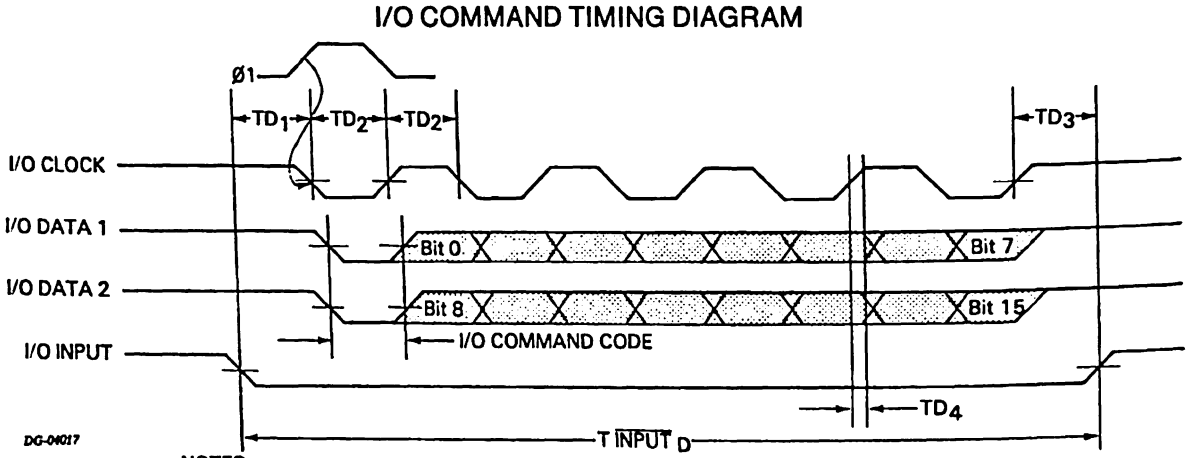
**NOTE:** The timing for incoming data to the CPU is the same as that for outgoing data, except that the I/O INPUT signal remains at a high logic level.

# mN601

## I/O OPERATIONS

### I/O Command

The format shown in the I/O Command Timing Diagram is used by the CPU to transfer all instructions to the I/O bus in which bits 0-2 contain 011 (I/O instruction format).



#### NOTES

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

**RULE:** 00<sub>g</sub>, 01<sub>g</sub>, 02<sub>g</sub>, 03<sub>g</sub>, and 77<sub>g</sub> are ILLEGAL DEVICE CODES for any I/O device connected to the microNOVA I/O bus.

The mN601 Instruction Set section describes several groups of instructions which incorporate the I/O format: I/O, CPU I/O, Multiply/Divide, and Stack. Each of these groups is discussed below.

Each I/O instruction should contain a device code in bits 10-15 which specifies one of the existing devices connected to the I/O bus. When one of these instructions is sent to the I/O bus by the CPU, each mN603 I/O controller examines the device code and determines what action, if any, should be taken. Most instructions in this group are followed by a data transfer, either from the CPU or the I/O controller. See mN601 Instruction Set.

Each CPU I/O instruction contains device code 77<sub>g</sub> (illegal device code). Some of these instructions are global in nature and are directed to all I/O controllers; others are acted upon internally by the CPU. The latter should be ignored by all I/O controllers. The effect of each is described in the CPU I/O Instructions section of the mN601 Instruction Set.

Both the Multiply and Divide instructions address device code 01<sub>g</sub> (illegal device code). Although these instructions are internal to the CPU, the instruction is sent to the I/O bus when executed; but, no data is transferred. I/O controllers should ignore these instructions.

Stack instructions also address device code 01<sub>g</sub> (illegal device code). Their effect is the same as that for Multiply/Divide instructions. I/O controllers should ignore these instructions.

# mN601

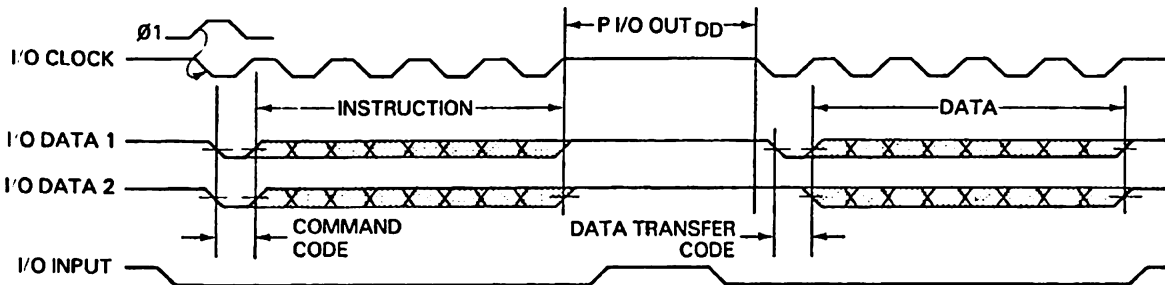
## I/O TRANSFER PROTOCOL

### I/O Transfer Protocol

#### Programmed I/O Data Out

The programmed data out timing diagram depicts a transfer of an I/O instruction followed by a data transfer from the CPU; e.g., DOA, DOB, DOC, MSKO. After the instruction is sent to the I/O bus, the CPU waits for a fixed interval of time before sending the data.

**PROGRAMMED DATA OUT TIMING DIAGRAM**



DG-04018

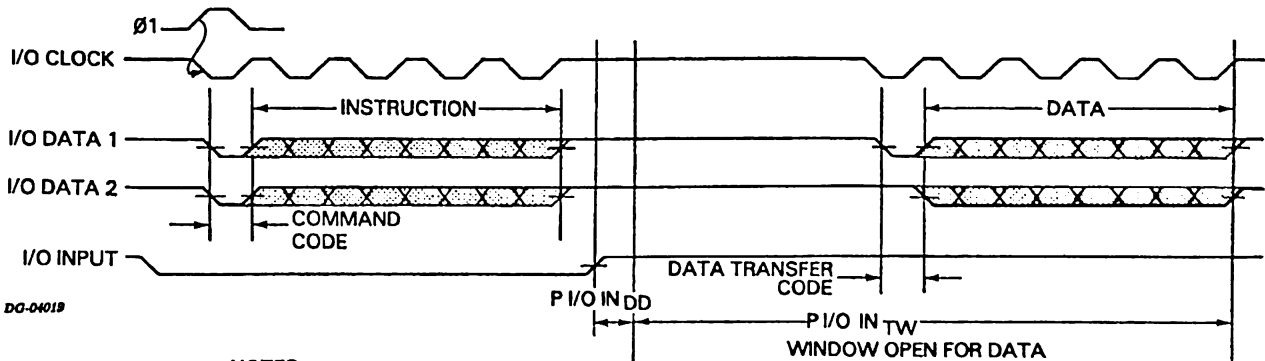
**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

#### Programmed I/O Data In

The programmed data in timing diagram depicts a transfer of an I/O instruction followed by a data transfer from an mN603 I/O controller; e.g., DIA, DIB, DIC, INTA, I/O SKIP. After the CPU sends the instruction to the I/O bus, it switches to I/O input mode and opens a window for the reception of a data word for a fixed interval of time. It is expected that the I/O controller addressed by the instruction will send 16 data bits plus the control bits during this time frame. When the window closes, the CPU performs the command specified in the instruction using the data bits received.

**PROGRAMMED DATA IN TIMING DIAGRAM**



DG-04019

**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.



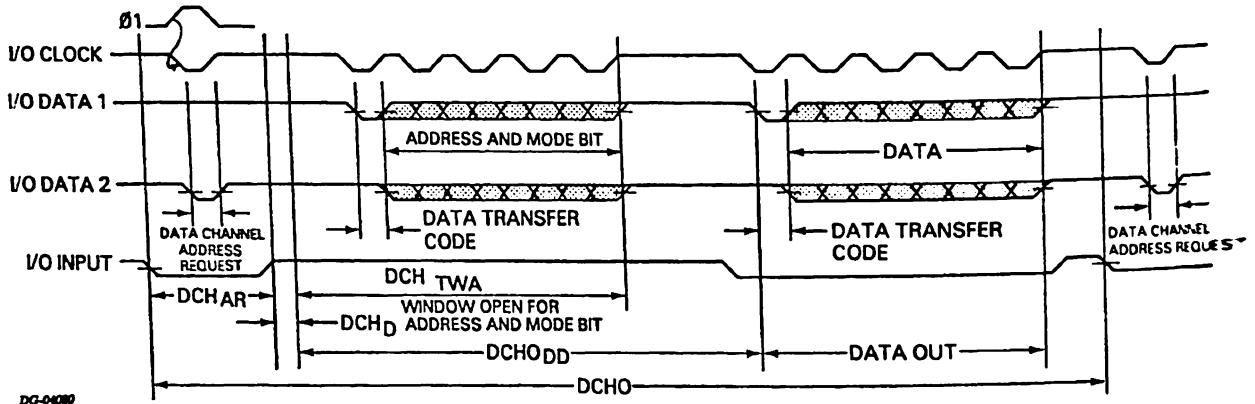
# mN601

## I/O TRANSFER PROTOCOL

### Data Channel Data Out

The sequence of events from the time the CPU acknowledges a data channel request through the time the data is placed on the I/O bus is depicted in the timing diagram below. After the CPU sends a Data Channel Address Request to the bus, it switches to data channel input mode. At this time the CPU opens a data-in window for a fixed interval of time. It is expected that the I/O controller with the highest priority requesting a data channel break will send the memory address and the direction mode bit during this time frame. When the window closes, the memory address contained in the data transfer is used by the CPU in performing a memory read operation. When the data is retrieved from memory, the CPU transfers it to the I/O bus by performing a data out operation.

DATA CHANNEL MEMORY TO DEVICE TIMING DIAGRAM



DG-04080

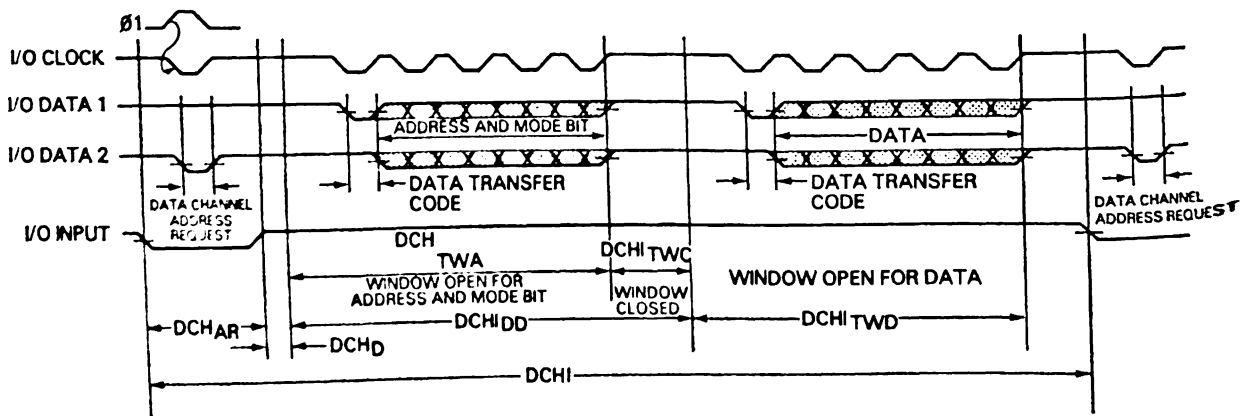
**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

### Data Channel Data In

The sequence of events from the time the CPU acknowledges a data channel request through the time the data is received by the CPU from the I/O bus is depicted in the timing diagram below. After the CPU sends a Data Channel Address Request, it switches to data channel input mode. Then, it opens the data-in window twice for fixed intervals of time. The first time is to retrieve the memory address and mode bit. The second time it is expected that the I/O controller will send one 16-bit word plus the control bits. When the window closes, the data bits received are sent to the memory bus and a memory write operation is performed.

DATA CHANNEL DEVICE TO MEMORY TIMING DIAGRAM



DG-04081

**NOTES**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 49.

# mN601

## PROGRAMMED I/O LATENCY

### Timing

On systems which have a number of I/O controllers and depend heavily on input/output, both programmed I/O and data channel facilities can be overloaded. This overloading causes certain peripherals to lose data due to the inability of the system to respond to them in time.

#### Programmed I/O Latency

Most peripherals operating under program control request processor time by setting their Done flags to 1. Whether the CPU determines that a device is requesting service by repeatedly checking the state of the flags using I/O Skip instructions or by means of program interrupts, there is a delay between the time the peripheral requests service (sets its Done flag to 1) and the time the CPU fetches the first instruction of the peripheral service routine. This delay is called programmed I/O latency.

When the program interrupt facility is not used, this time is dependent upon the following: frequency of the I/O Skip instructions and efficiency of the software servicing the peripherals; i.e., the time required by the peripheral service routine to transfer data to or from the peripheral and set the Done flag to 0.

When the program interrupt facility is used, I/O latency is defined as the sum of the following:

1. The time from the setting of the Done flag to 1 to the end of the instruction currently being executed by the CPU.
2. The time when CPU operation is suspended while data channel transfers are in progress. (See the following section.)
3. The time program interrupts are turned off, e.g., during the servicing of an interrupt from another peripheral.
4. The time required to execute two of the longest consecutive instructions present in the program. See the table of Instruction Execution Times at the end of this section.
5. Interrupt cycle time, which is comprised of the following: 6 CPU cycles plus 2 cycles per indirect level, where 1 CPU cycle equals  $0.480\mu\text{sec}$ .  
Interrupt cycle time is the time required for the interrupt facility hardware to store the contents of the program counter plus 1 in location 0, and to simulate a Jump indirect instruction to location 1 (or to location 2 or 3 if the interrupt is generated by the CPU's real-time clock or stack overflow, respectively).
6. The time required by the interrupt handler to identify the peripheral and transfer control to that peripheral's service routine.
7. The time required by the service routine to transfer data to or from the peripheral and set Done to 0.

Items 1, 4, and 5 are machine dependent. Item 2 depends upon the number of data channel devices and their activity rate. Items 3, 6 and 7 are determined by the software and represent the bulk of the programmed I/O interrupt latency.

Additionally, if any higher priority program interrupt, i.e., stack overflow or real-time clock, has a pending interrupt, the time calculation is the same. The CPU always honors the interrupt with the highest priority.

Programmed I/O interrupt latency is important because a peripheral device that must wait too long to be serviced may suffer from loss of data. For this reason, the mN601 incorporates the priority interrupt facility. To minimize the loss of important data, the software priority levels should be assigned on the basis of the following considerations:

- A. The maximum allowable programmed I/O latency for each peripheral device.
- B. The result of exceeding the maximum allowable programmed I/O latency for each peripheral device; i.e., slowdown or data loss.

# mN601

## DATA CHANNEL LATENCY

### Data Channel Latency

Latency is also a factor when data is transferred between peripheral devices and memory via data channel breaks. The CPU allows data channel devices to access memory only during predefined times. In addition, more than one peripheral may be waiting for service at any given time.

Data channel latency is defined as the time between the I/O device's request for service (the  $\overline{\text{DCH SYNC}}$  line is asserted low to the device - see mN603) and the time when data is strobed, either during data reception or transmission. Data channel latency consists of the sum of either of the following values:

Data Channel Data In  
(Device to Memory)

The time required to execute the two longest consecutive instructions, excluding Multiply and Divide, (see the table of Instruction Execution Times) plus 6 CPU cycles, where 1 CPU cycle equals 0.480  $\mu$ sec.

Data Channel Data Out  
(Memory to Device)

The time required to execute the two longest consecutive instructions, excluding Multiply and Divide, plus 11 CPU cycles.

For example, if the two longest consecutive instructions in the program are 15 CPU cycles each, the maximum latency for a data channel data out transfer is:

$$15 + 15 + 11 = 41 \text{ CPU cycles;} \\ 41 \times .480 \mu\text{sec} = 19.68 \mu\text{sec}$$

Additionally, there is a minimum data channel latency. The minimum latency for a data channel data in is 11 CPU cycles; the minimum for a data channel data out is 16 CPU cycles.

Minimum latency can be used in computing data channel latency when it is known that the device will request a data transfer. Thus, a data channel request can be initiated 11 CPU cycles (for a data channel data in) or 16 CPU cycles (for a data channel data out) before the device is ready to transfer data; thereby reducing the latency factor from data ready to data transfer.

The above computations assume the request is by the highest priority device requesting service.

When there are other, higher priority devices requesting data channel transfers, the time required for these transfers must be added to the latency. For each data channel out transfer the time is 7.2  $\mu$ sec and for each data channel in transfer the time is 5.8  $\mu$ sec.

# mN601 STATUS

## STATUS

### Initialization

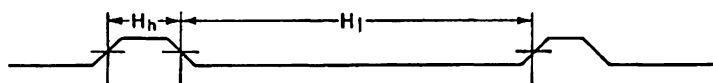
On power up,  $V_{BB}$  must be at  $-3V$  before  $V_{GG}$ ,  $V_{DD}$  and  $V_{CC}$  start going positive.

After the mN601 is initially brought up to the recommended operating specifications (see DC Characteristics), the  $\overline{CLAMP}$  pin must be held at a low logic level for a minimum of  $100\mu\text{sec}$ . Then, when  $\overline{CLAMP}$  goes from the low to high state, the CPU is properly initialized and enters the HALT state (see below). After initialization, pulling  $\overline{CLAMP}$  low for a period of  $100\mu\text{sec}$  causes the mN601 to reset and enter the HALT state when  $\overline{CLAMP}$  goes from the low to high logic state. The contents of internal registers and RAM memory are indeterminate whenever  $\overline{CLAMP}$  is low for a period greater than  $100\mu\text{sec}$ .

### Halt

When the mN601 is in the HALT state, it generates a sequence of  $0.417\text{MHz}$  pulses from the HALT pin as shown in the timing diagram below; otherwise, the HALT pin remains in the low logic state.

HALT TIMING DIAGRAM



#### NOTES

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT  $1.5V$ .
- SEE TIMING TABLE ON PAGE 49.

DG-04329

In the HALT state, the CPU performs the following functions: it generates Request Enable signals, it responds to both program interrupts and data channel requests, and it generates dynamic RAM refresh signals. The normal way to start the mN601 when it is in the HALT state is to generate an interrupt.

### Reset

The mN601 is reset by pulling the I/O CLOCK pin low for a period of  $10\mu\text{sec}$ . When I/O CLOCK rises to the high logic state after the CPU is reset, the mN601 enters the HALT state and the contents of internal registers are indeterminate.

**NOTE:** Do NOT pull I/O CLOCK low when I/O INPUT is low.

# mN601

## INSTRUCTION EXECUTION TIMES

### INSTRUCTION EXECUTION TIMES (IN CPU CYCLES<sup>(1)</sup>)

INSTRUCTION GROUP TYPE	ADDRESSING MODES		
	DIRECT	INDIRECT (2)	AUTO INCREMENT/DECREMENT
<b>MEMORY REFERENCE GROUP</b>			
DSZ	8 (3)	10 (3)	13 (3)
ISZ	8 (3)	10 (3)	13 (3)
JMP	6	7	10
JSR	7	9	12
LDA	6	8	11
STA	6	8	11
<b>ARITHMETIC AND LOGICAL GROUP</b>			
<b>REGISTER ADDRESSING</b>			
ADC		5 (3)	
ADD		5 (3)	
AND		5 (3)	
COM		5 (3)	
INC		5 (3)	
MOV		5 (3)	
NEG		5 (3)	
SUB		5 (3)	
<b>MULTIPLY/DIVIDE GROUP</b>			
<b>REGISTER ADDRESSING</b>			
DIV		122	
MUL		87	
<b>INPUT/OUTPUT GROUP</b>			
<b>REGISTER ADDRESSING</b>			
DIA		15	
DIB		15	
DIC		15	
DOA		10	
DOB		10	
DOC		10	
NIO		10	
SKP		15 (3)	
<b>STACK CONTROL GROUP</b>			
<b>REGISTER ADDRESSING</b>			
MFFP		8	
MFSP		7	
MTFP		6	
MTSP		6	
POPA		7	
PSHA		7	
RET		15	
SAV		16	
<b>CPU CONTROL GROUP</b>			
<b>REGISTER ADDRESSING</b>			
HALT		10	
INTA		15	
INTDS		10	
INTEN		10	
IORST		10	
MSKO		10	
RTCEN		10	
RTCDS		10	
TRAP		9 (2)	

**NOTE:** (1) ONE CPU CYCLE = 0.480  $\mu$  sec  
 (2) FOR EACH ADDITIONAL INDIRECT LEVEL, ADD 2 CYCLES;  
 FOR EACH AUTO-INCREMENT/DECREMENT, ADD 5 CYCLES  
 (3) IF SKIP OCCURS, ADD 2 CYCLES

# mN601

## ELECTRICAL SPECIFICATIONS

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{BB}$ .....	-7V
Supply voltage, $V_{CC}$ .....	7V
Supply voltage, $V_{DD}$ .....	13V
Supply voltage, $V_{GG}$ .....	17V
Input voltage, $V_I$ .....	7V
Operating free air temperature range, $T_A$ .....	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg.C

22

**NOTE:** All voltages are referenced to ground. Subjecting a circuit to conditions either outside or at these limits for an extended period of time may cause irreparable damage to the circuit. As such, these ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{BB}$ .....	$-4.25V \pm 0.5V$
Supply voltage, $V_{CC}$ .....	$5V \pm 0.25V$
Supply voltage, $V_{DD}$ .....	$10V \pm 1V$
Supply voltage, $V_{GG}$ .....	$14V \pm 1V$
Operating free air temperature range, $T_A$ .....	0 deg. to 70 deg.C

#### C CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN	TYP	MAX	UNITS
L	$\phi 1, \phi 2$ (clock input) MB <0-15>, $\overline{CLAMP}$ , $\overline{EXTINT}$ , $\overline{DCHINT}$ I/O CLOCK, I/O DATA 1, I/O DATA 2		-2.0		0.8	V
			-1.0		1.0	V
			-0.5		0.5	V
H	$\phi 1, \phi 2$ MB <0-15>, $\overline{CLAMP}$ , $\overline{EXTINT}$ , $\overline{DCHINT}$ I/O CLOCK, I/O DATA 1, I/O DATA 2		13		15	V
			3.75		5.8	V
			2.70		5.8	V
L	$\phi 1, \phi 2$ MB <0-15> $\overline{EXTINT}$ , $\overline{DCHINT}$ $\overline{CLAMP}$ I/O CLOCK, I/O DATA 1, I/O DATA 2	$V_{IL} = 0.8V$ $V_{IL} = 1V$ $V_{IL} = 1V$ $V_{IL} = 1V$ $V_{IL} = 0.5V$			10	$\mu A$
					-2	mA
					-4	mA
					10	$\mu A$
					-4	mA

# mN601

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (Cont.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN	TYP	MAX	UNITS
I <sub>IH</sub>	$\phi 1, \phi 2$ MB <0-15> EXTINT, DCHINT CLAMP I/O CLOCK, I/O DATA 1, I/O DATA 2	V <sub>IH</sub> = 15V			10	$\mu A$
		V <sub>IH</sub> = 5.8V			60	$\mu A$
		V <sub>IH</sub> = 5.8V			100	$\mu A$
		V <sub>IH</sub> = 5.8V			10	$\mu A$
		V <sub>IH</sub> = 5.8V			100	$\mu A$
VPCHS	MB <0-15> (VOLTAGE AFTER PRECHARGE)	IPCHS = -60 $\mu A$	4.25	4.75	5.25	V
IPCHS	MB <0-15> (CURRENT AFTER PRECHARGE)	VPCHS = 4.25V			-60	$\mu A$
V <sub>OL</sub>	MB <0-15>, SAE, WE, PAUSE, P, I/O INPUT, I/O CLOCK, I/O DATA 1 I/O DATA 2, HALT				0.5	V
V <sub>OH</sub>	MB <0-15>, SAE, WE, PAUSE, P, I/O INPUT, I/O CLOCK, I/O DATA 1 I/O DATA 2, HALT		2.75			V
I <sub>OL</sub>	MB <0-15>, SAE, WE, PAUSE, HALT I/O CLOCK, I/O DATA 1, I/O DATA 2 I/O INPUT, P	V <sub>OL</sub> = 0.5V			2.0 2.0 4.0	$mA$ $mA$ $mA$
I <sub>OH</sub>	MB <0-15>, SAE, WE, PAUSE, HALT I/O CLOCK, I/O DATA 1, I/O DATA 2 P, I/O INPUT.	V <sub>OH</sub> = 2.75V			-100 -100 -200	$\mu A$ $\mu A$ $\mu A$
I <sub>BB</sub>	Average				-250	$\mu A$
I <sub>CC</sub>	Average				20	$mA$
I <sub>DD</sub>	Average				60	$mA$
I <sub>GG</sub>	Average				30	$mA$
<b>CAPACITANCE</b>						
Clock line ( $\phi 1, \phi 2$ ) capacitance					100	$pF$
Capacitance of all other signal pins					10	$pF$
External load capacitance on MB <0-15>. I/O CLOCK, I/O DATA 1, I/O DATA 2					20	$pF$
External load capacitance between any two MB lines.					10	$pF$
External load capacitance on P. I/O INPUT, SAE, WE, PAUSE, HALT					30	$pF$

NOTE: Positive current is into the pin.

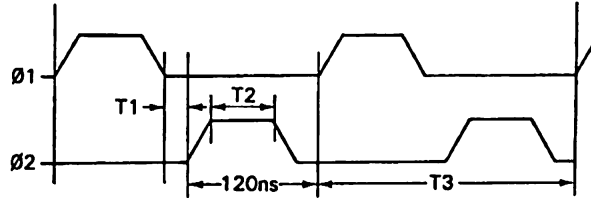
# mN601

## ELECTRICAL SPECIFICATIONS

### AC CHARACTERISTICS

#### Switching Diagrams

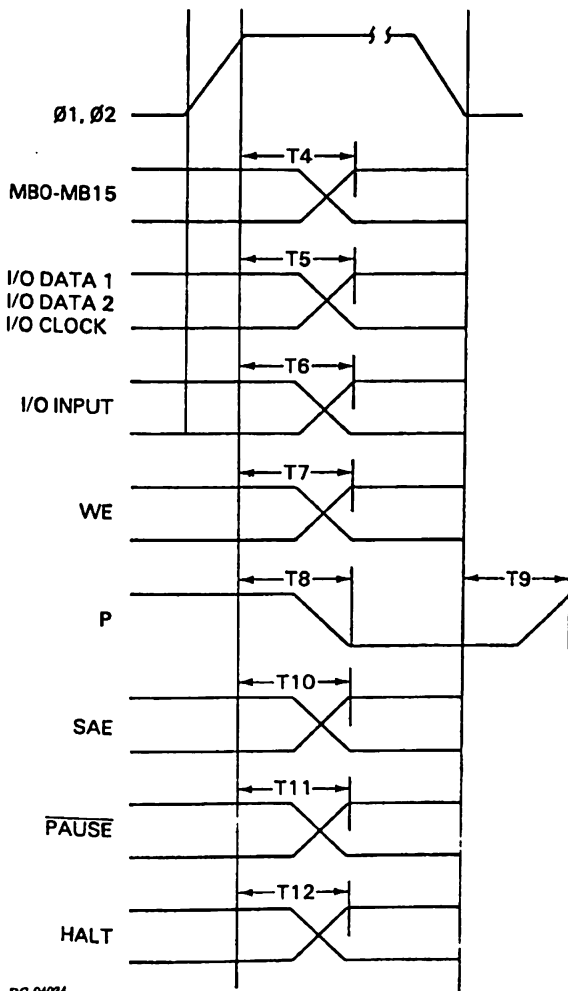
#### CLOCKS



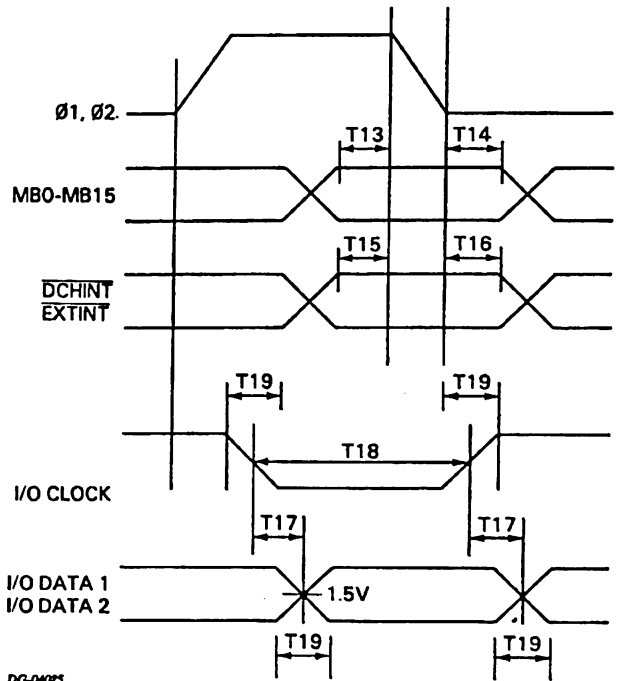
DG-04023

#### OUTPUT MODE

#### INPUT MODE



DG-04024



DG-04025

**NOTE:** Time intervals are all measured between 10% and/or 90% points, unless otherwise specified.



# mN601

## ELECTRICAL SPECIFICATIONS

### Transition Timing

FUNCTION	SYMBOL	PIN	MIN.	MAX.	UNITS
MASTER CLOCK FREQUENCY		ACCURACY .1%		8.333	MHz
CLOCKS	T1	$\phi 1, \phi 2$ SEPARATION	5	25	ns
	T2	$\phi 1, \phi 2$ WIDTH	75	95	ns
	T3	$\phi 1, \phi 2$ CYCLE	235	245	ns
OUTPUTS	T4	MB 0 - MB 15		80	ns
	T5	I/O DATA 1, I/O DATA 2, I/O CLOCK		25	ns
	T6	I/O INPUT		25	ns
	T7	WE		25	ns
	T8	P FALL		25	ns
	T9	P RISE		25	ns
	T10	SAE		40	ns
	T11	$\overline{\text{PAUSE}}$		25	ns
T12	HALT		40	ns	
INPUTS	T13	MB 0 - MB 15 SET-UP TIME	40		ns
	T14	MB 0 - MB 15 HOLD TIME	20		ns
	T15	$\overline{\text{DCHINT}}, \overline{\text{EXTINT}}$ SET-UP TIME	100		ns
	T16	$\overline{\text{DCHINT}}, \overline{\text{EXTINT}}$ HOLD TIME	20		ns
	T17	I/O<CLOCK/DATA 1/DATA 2> SKEW	-5	+15	ns
	T18	I/O<CLOCK/DATA 1/DATA 2> WIDTH	115	125	ns
	T19	I/O<CLOCK/DATA 1/DATA 2> RISE/FALL		10	ns

TIMING TABLES

MEMORY READ/WRITE TIMING  
(See Memory Operations)

OPERATION	SYMBOL	DESCRIPTION	MIN.	MAX.	UNITS
COMMON TO ALL MEM CYCLES	PT	P LOW DURATION	0.230	5.570	μs
	PT	P HIGH DURATION	0.710	3.180	μs
	AS	ADDRESS SETUP TIME	0.040	0.130	μs
	AH	ADDRESS HOLD TIME	0.110	0.190	μs
	PC	PRECHARGE		0.120	μs
READ	SAED	P TO SAE DELAY	0.110	1.150	μs
	DD	SAE TO DATA DELAY	0.060	0.130	μs
	SAET	SAE DURATION	0.200	0.260	μs
	DT	DATA DURATION	0.080	0.360	μs
WRITE/ REFRESH(1)	WED	P TO WE DELAY	0.230	0.290	μs
	WET	WE TIME DURATION	0.220	0.260	μs
	WDS	WRITE DATA SETUP TIME	0.060	0.140	μs
	WDH	WRITE DATA HOLD TIME	0.100	0.160	μs
READ MODIFY WRITE	RWED	READ MODIFY WRITE WE DELAY	1.190	1.730	μs

DC-0418

(1) Applies to both write and refresh operations.  
WDS and WDH apply to write operation only.

NOTE: Rise and fall times for input signals are assumed ≤ 10ns.

I/O BUS TIMING  
(See I/O Operations)

OPERATION	SYMBOL	DESCRIPTION	MIN.	MAX.	UNITS
COMMON TO ALL OUTPUT OPERATIONS	TD <sub>1</sub>	I/O INPUT TO I/O CLOCK TIME (START OF TRANSACTION)	0.090	0.150	μs
	TD <sub>2</sub>	I/O CLOCK 1/2 PERIOD (BIT TIME)	0.110	0.130	μs
	TD <sub>3</sub>	I/O CLOCK TO I/O INPUT TIME (END OF TRANSACTION)	0.090	0.150	μs
	TD <sub>4</sub>	I/O CLOCK TO I/O DATA SKEW	-5	+15	ns
I/O COMMAND OR DATA OUTPUT	T <sub>INPUT D</sub>	COMMAND OR DATA TRANSFER TIME	1.310	1.330	μs
I/O REQUEST ENABLE OR DATA CHANNEL ADDRESS REQUEST	T <sub>INPUT R</sub>	REQUEST ENABLE OR DATA CHANNEL ADDRESS REQUEST TRANSFER TIME	0.350	0.370	μs

DATA TRANSFER TIMING

PROGRAMMED I/O DATA OUT	PI/O OUT <sub>DD</sub>	PROGRAMMED I/O DATA OUT - DATA DELAY	0.83	0.85	μs
PROGRAMMED I/O DATA IN	PI/O IN <sub>TW</sub>	PROGRAMMED I/O DATA IN - DATA RECEPTION WINDOW	--	3.12	μs
	PI/O IN <sub>DD</sub>	PROGRAMMED I/O DATA IN - DATA DELAY	0.11	0.13	μs
COMMON TO ALL DATA CHANNEL OPERATIONS	DCH <sub>AR</sub>	DATA CHANNEL ADDRESS REQUEST	0.350	0.370	μs
	DCH <sub>D</sub> DCH <sub>TWA</sub>	DATA CHANNEL DELAY DATA CHANNEL ADDRESS AND MODE BIT RECEPTION WINDOW	0.11 --	0.13 2.6	μs
DATA CHANNEL OUT - MEMORY TO DEVICE	DCHO	CYCLE TIME FOR 16-BIT WORD TRANSFER DATA CHANNEL DATA OUT DELAY	--	5.8	μs
	DCHO <sub>DD</sub>		3.11	3.13	μs
DATA CHANNEL IN - DEVICE TO MEMORY	DCHI	CYCLE TIME FOR 16-BIT WORD TRANSFER DATA CHANNEL DATA IN DELAY DATA CHANNEL WINDOW CLOSED DATA CHANNEL DATA RECEPTION WINDOW	--	7.2	μs
	DCHI <sub>DD</sub>		3.54	3.56	μs
	DCHI <sub>TWC</sub> DCHI <sub>TWD</sub>		--	0.96 1.9	μs

HALT TIMING  
(See Status)

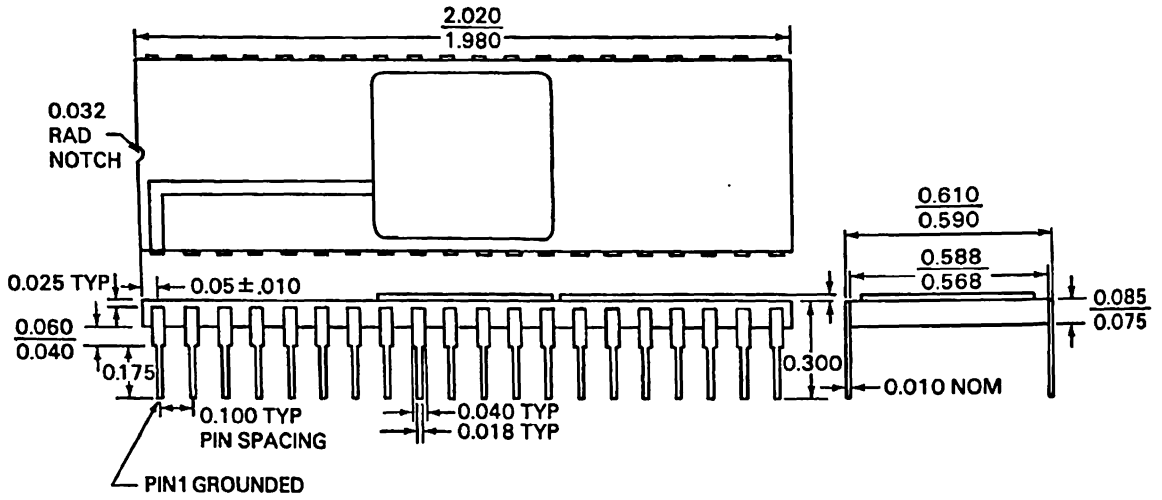
OPERATION	SYMBOL	DESCRIPTION	TYP.	UNITS
HALT	H <sub>H</sub>	DENOTES HALT SIGNAL IN HIGH STATE	0.24	μs
	H <sub>L</sub>	DENOTES HALT SIGNAL IN LOW STATE	2.16	μs

DC-0419

# mN601

## PACKAGE SPECIFICATIONS

### PACKAGE SPECIFICATIONS



DG-04001

This page intentionally left blank.

**mN603  
I/O CONTROLLER  
CONTENTS**



<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>5</b>	<b>OVERVIEW</b>
<b>6</b>	<b>PERIPHERAL PORT</b>
<b>6</b>	<b>DATA LINES</b>
<b>6</b>	<b>CONTROL LINES</b>
<b>8</b>	<b>PERIPHERAL PORT TIMING</b>
<b>9</b>	<b>DECODING THE CONTROL LINES</b>
<b>11</b>	<b>PROGRAMMED I/O INSTRUCTIONS</b>
<b>11</b>	<b>Data Out Instructions</b>
<b>12</b>	<b>Data In Instructions</b>
<b>12</b>	<b>Mask Out and I/O Reset Instructions</b>
<b>13</b>	<b>DATA CHANNEL TRANSACTIONS</b>
<b>15</b>	<b>Data Channel Timing</b>
<b>17</b>	<b>I/O DATA PORT</b>
<b>17</b>	<b>INFORMATION TYPES</b>
<b>18</b>	<b>Request Enable</b>
<b>18</b>	<b>Data Channel Address Request</b>
<b>19</b>	<b>I/O Instructions</b>
<b>20</b>	<b>Data</b>
<b>21</b>	<b>TRANSACTION PROTOCOLS</b>
<b>21</b>	<b>Request Enable Transactions</b>
<b>22</b>	<b>Programmed I/O Transactions</b>
<b>22</b>	1) CPU to IOC Data Out Transactions
<b>25</b>	2) IOC to CPU Data In Transactions
<b>27</b>	<b>Data Channel Transactions</b>
<b>28</b>	1) CPU to IOC Data Channel Out Transactions
<b>28</b>	2) IOC to CPU Data Channel In Transactions



# mN603

## CONTENTS

29	<b>INTERNAL STRUCTURE</b>
30	INTERNAL BUSY/DONE LOGIC
32	INTERRUPT REQUEST LOGIC
33	DATA CHANNEL LOGIC
34	INTERRUPT AND DATA CHANNEL SEQUENCING DIAGRAM
35	<b>REQUEST CONTROL</b>
35	PRIORITY NETWORKS
36	INTERRUPT REQUESTS
36	EXTERNAL BUSY/DONE LOGIC
37	DATA CHANNEL REQUESTS
37	DATA CHANNEL REGISTERS
37	INTERRUPT AND DATA CHANNEL SERVICE
38	<b>SYSTEM REQUIREMENTS</b>
38	POWER UP
38	INITIALIZATION
38	CLOCKS
39	<b>ELECTRICAL SPECIFICATIONS</b>
39	ABSOLUTE MAXIMUM RATINGS
39	RECOMMENDED OPERATING CONDITIONS
39	DC CHARACTERISTICS
41	AC CHARACTERISTICS
41	Switching Diagrams
43	Transition Timing
43	<b>TIMING TABLES</b>
44	<b>PACKAGE SPECIFICATIONS</b>

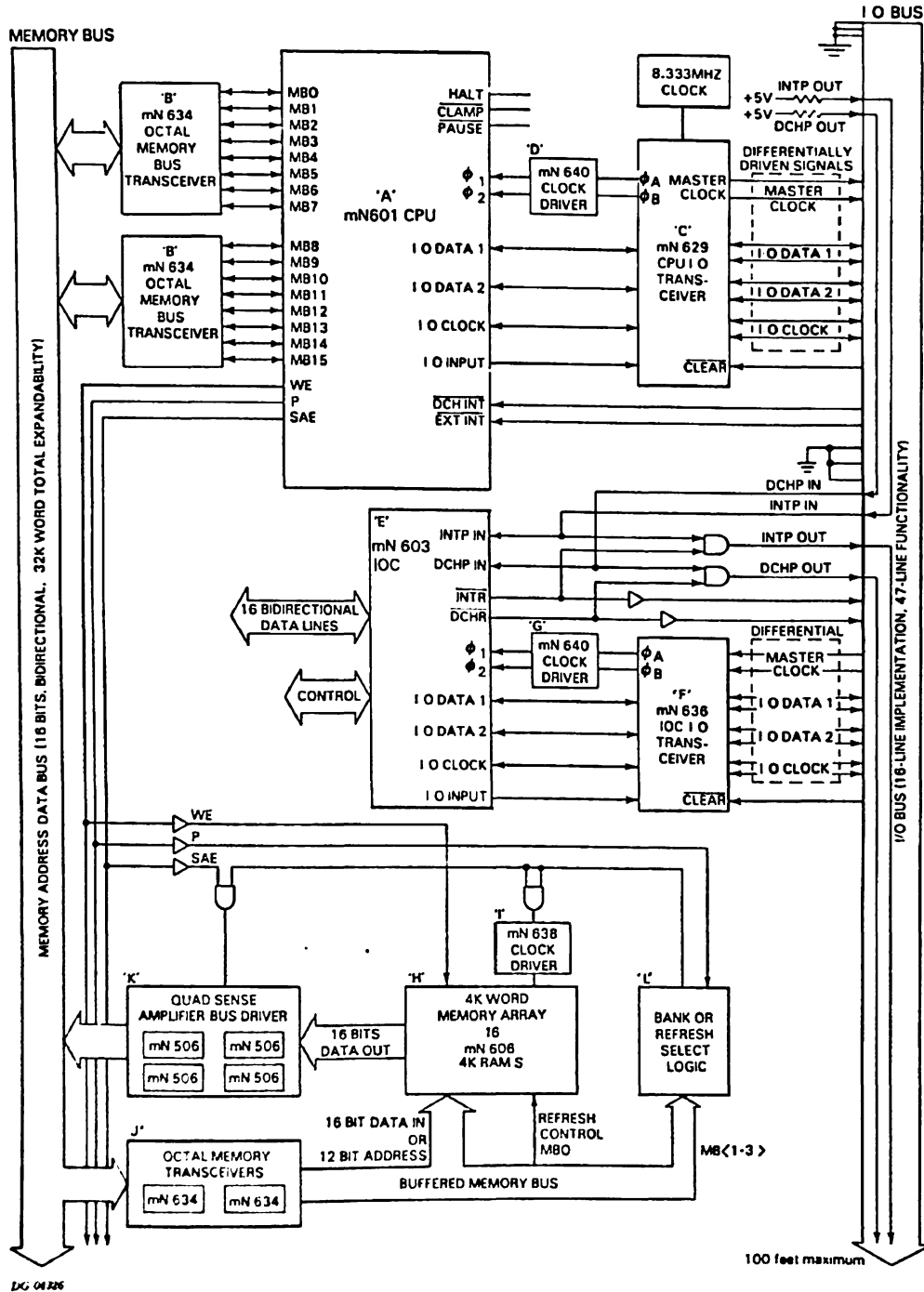




This page intentionally left blank.



# mN603 I/O CONTROLLER



DG 04226

100 feet maximum



Data General Corporation, Westboro, Massachusetts 01581

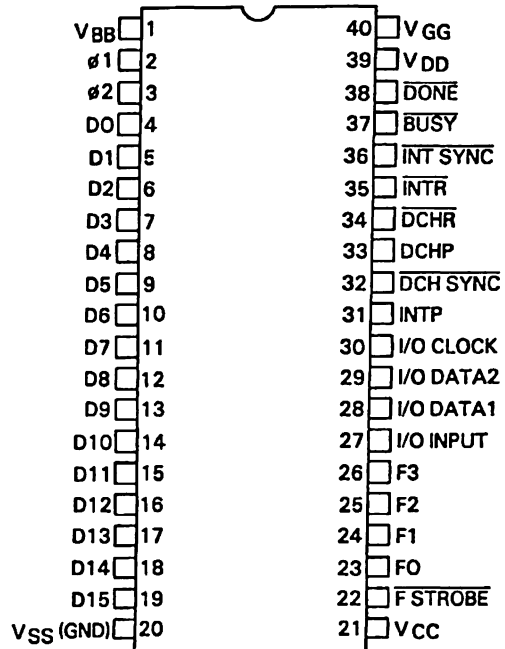
# mN603

## I/O CONTROLLER

### FEATURES

- I/O DEVICE CONTROLLER ON A SINGLE 40-PIN SILICON-GATE NMOS CHIP
- INTERPRETS AND EXECUTES DATA GENERAL'S STANDARD NOVA COMPUTER I/O INSTRUCTION SET
- PROVIDES SIMPLE 16-BIT PARALLEL USER INTERFACE
- INTEGRAL BUSY AND DONE CONTROL
- INTEGRAL DEVICE IDENTIFICATION AND INTERRUPT CONTROL
- CONTAINS DATA CHANNEL CONTROL LOGIC
- FULL ADDRESS AND WORD COUNT REGISTERS FOR DATA CHANNEL OPERATION
- USER SELECTABLE DATA BUS SIGNAL POLARITY

### PACKAGE



DG-01137

### GENERAL DESCRIPTION

The mN603 is an I/O controller (IOC) that provides the full functional capability of Data General's 47-line NOVA I/O bus. The IOC decodes an encoded data stream from the CPU and presents a parallel 16-bit bidirectional interface, four encoded function bits, and a function strobe, to the peripheral for simple interfacing.

The IOC includes a number of bus adapter functions found in the most powerful minicomputer systems. It includes integral device identification, BUSY/DONE interrupt logic, and a per-device interrupt masking capability. For block-oriented controllers, the IOC includes data channel bus hand shaking and full address and word counters.

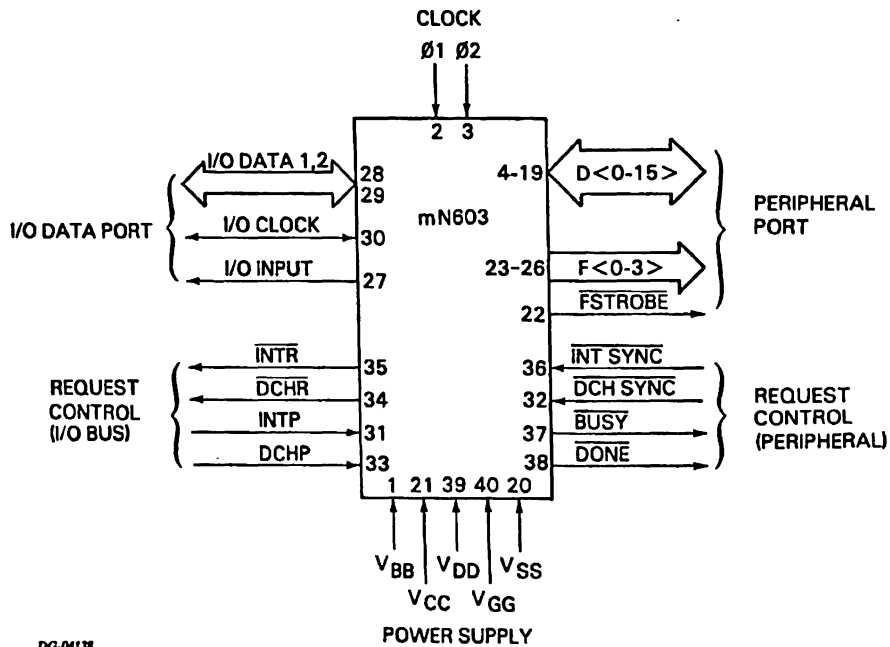
The IOC allows the construction of a complete I/O controller module using a minimum of additional components. Those necessary are:

- 1) an mN640 Clock Driver
- 2) an mN636 IOC I/O Transceiver
- 3) Interrupt and data channel priority circuitry
- 4) decode logic for the IOC's four encoded function bits
- 5) additional gating to interface a peripheral and buffer the IOC

# mN603

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



DG-04138

The following table describes the function of each pin shown in the pin connection diagram:

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
<b>CLOCKS</b>			
Ø1	2	IN	Two-phase non-overlapping clock. Operates between 0 and 14V amplitude. Generates 4-phase internal clock, providing internal timing.
Ø2	3	IN	
<b>I/O DATA PORT</b>			
I/O DATA1	28	I/O	Two-line, bi-directional bus. Receives all data and I/O command information transmitted serially from the CPU at the Master Clock rate. Transmits data to the CPU at this rate. During a 16 bit transfer, I/O DATA1 carries bits 0-7 while I/O DATA2 carries bits 8-15.
I/O DATA2	29	I/O	
I/O CLOCK	30	I/O	Synchronizes all I/O transfers on I/O DATA <1,2>. If receiving, I/O CLOCK strobbs information received on I/O DATA <1,2> into the IOC. If transmitting, I/O CLOCK is generated by the IOC. Note: The IOC will be reset if I/O CLOCK is held low for more than 8 cycles of MASTER CLOCK*.
I/O INPUT	27	OUT	Indicates whether the IOC is transmitting or receiving on I/O DATA <1,2>. High = IOC receiving. Low = IOC transmitting.

# mN603

## PIN DESCRIPTIONS

### PIN DESCRIPTIONS (CONT.)

MNEMONIC	PIN NO.	IN/ OUT	FUNCTION
<b>PERIPHERAL PORT</b>			
$\overline{D}<0-15>$	4-19	I/O	16-line, bi-directional data bus. Transfers all data between an IOC and a peripheral device. Data is interpreted using either positive or negative logic as selected during initialization.
$\overline{F}<0-3>$	23-26	OUT	Four line encoded control bus. Four bit Function codes may be decoded into one of 16 separate functions which control the gating of data between a peripheral and $\overline{D}<0-15>$ . Code is valid while $\overline{FSTROBE}$ is low.
$\overline{FSTROBE}$	22	OUT	Used to synchronize function code. Code is valid when $\overline{FSTROBE}$ is low. (One $\overline{FSTROBE}$ clock period is equal to four MASTER CLOCK* periods.)
<b>REQUEST CONTROL</b>			
$\overline{INTR}$	35	OUT	Asserted by an IOC to request program interrupts from the CPU.
$\overline{DCHR}$	34	OUT	Asserted by an IOC to request a Data Channel Break from the CPU.
$\overline{INTP}$	31	IN	Indicates whether an IOC requesting a program interrupt has priority to respond to an Interrupt Acknowledge instruction. $\overline{INTP}$ high indicates priority.
$\overline{DCHP}$	33	IN	Indicates whether an IOC requesting data channel service has priority to respond to a Data Channel Address Request. $\overline{DCHP}$ high indicates priority.
$\overline{INT SYNC}$	36	IN	Asserted by peripheral; prompts the IOC to request a program interrupt from the CPU (assert its $\overline{INTR}$ line).
$\overline{DCH SYNC}$	32	IN	Asserted by peripheral; prompts the IOC to request data channel service from the CPU (assert its $\overline{DCHR}$ line).
$\overline{BUSY}$	37	I/O	Indicates the state of the internal BUSY flag; also used by the device to set the BUSY flag. When $\overline{BUSY}$ is low, the peripheral device is performing an operation.  <b>BUSY set to 1 by:</b> <ul style="list-style-type: none"> <li>● device asserting <math>\overline{BUSY}</math></li> <li>● an I/O instruction with an S in the F field (see I/O Data Port, Programmed I/O Transactions)</li> </ul> <b>BUSY set to 0 by:</b> <ul style="list-style-type: none"> <li>● device asserting <math>\overline{DONE}</math></li> <li>● an I/O instruction with a C in the F field (see I/O Data Port, Programmed I/O Transactions)</li> <li>● I/O Reset Instruction</li> </ul>
$\overline{DONE}$	38	I/O	Indicates the state of the internal DONE flag; also used by the device to set the DONE flag. When $\overline{DONE}$ is low, the peripheral device has finished an operation. A Program interrupt will be requested when $\overline{DONE}$ is low if the Interrupt Disable Flag is not set (see Internal Structure, Interrupt Request Logic).  <b>DONE set to 1 by:</b> <ul style="list-style-type: none"> <li>● device asserting <math>\overline{DONE}</math></li> </ul> <b>DONE set to 0 by:</b> <ul style="list-style-type: none"> <li>● an I/O instruction with an S, C, or P in the F field (see I/O Data Port, Programmed I/O Transactions)</li> <li>● an I/O Reset Instruction</li> </ul>

# mN603

## PIN DESCRIPTIONS

### PIN DESCRIPTIONS (CONT.)

MNEMONIC	PIN NO.	IN/ OUT	POWER
V <sub>BB</sub>	1		-4.25 ± 0.5 V
V <sub>CC</sub>	21		+5 ± 0.25 V
V <sub>DD</sub>	39		+10 ± 1.0 V
V <sub>GG</sub>	40		+14 ± 1.0 V
V <sub>SS</sub>	20		GROUND

NOTE: \* MASTER CLOCK refers to the system clock, see Overview Section of the microNOVA Line Integrated Circuits.

### OVERVIEW

The IOC, together with its supporting elements, provides a parallel user interface from the serial microNOVA I/O bus. This resulting interface is functionally equivalent to but not compatible with the 47-line bus used on Data General's NOVA and ECLIPSE line computer systems.

Each time the CPU fetches an I/O instruction from memory, it transmits the complete instruction over the I/O bus. Every IOC connected to the bus receives a copy of this instruction and internally decodes it. If the IOC determines that the instruction is directed to it, the IOC executes the instruction. Since it has an exact copy of the I/O instruction, the IOC can emulate all the programmed I/O control signals found in the NOVA/ECLIPSE I/O bus.

In addition to decoding programmed I/O instructions, the IOC contains all the logic necessary to implement a complete program interrupt system. This system is compatible with the mN601 and includes Busy and Done flags, an interrupt request line, interrupt masking logic, and interrupt source identification.

The IOC also provides the logic necessary to perform data channel transfers. This includes the request logic and two internal registers, a memory address register and a word count register, which control the data transfer.

The IOC automatically manages all the protocols found on the microNOVA I/O bus. These protocols include the transfer of data, of programmed I/O instructions, and of the system synchronization signals needed by the interrupt and data channel facilities.

Four major areas of the IOC perform the above tasks.

- The Peripheral Data Port provides the device with a 16-bit wide, bidirectional data bus and four function code control lines.
- The I/O Data Port performs the serial/parallel conversions of information transferred to and from the I/O bus.
- The internal registers, data paths, and logic arrays decode instructions and connect the two data ports.
- The Request Control Facility integrates the service requests into the rest of the microNOVA system.

These four areas along with the System Requirements are discussed in the following sections as described below:

**PERIPHERAL DATA PORT** - This section discusses how data is transferred between a peripheral device and an IOC. The Peripheral Data Port includes the 16 data lines (D<0-15>), the four function code lines (F<0-3>), and the clock (FSTROBE). The 16 data lines transfer data to and from the peripheral device. The function codes, with their clock, control the gating of data on these data lines.

**I/O DATA PORT** - This section explains how data is transferred between the serial I/O bus and an IOC. The I/O data port includes two serial data lines (I/O DATA1 and I/O DATA2), their strobe (I/O CLOCK), and a transmit/receive control line (I/O INPUT). The I/O DATA <1,2> lines receive commands and data from the I/O bus and transmit data to the I/O bus. I/O CLOCK strobes the information transmitted on these data lines. I/O INPUT indicates whether the IOC is transmitting or receiving on the data lines.

**INTERNAL STRUCTURE** - This section describes the internal data flow between the two data ports. A set of internal registers, together with their connecting data paths, join the I/O data port to the peripheral data port. All control of information transfers and service requests comes from the internal logic array and state change logic. Two of the internal registers used during data channel breaks may be replaced with external registers. Interrupt requests and data channel break requests are controlled by the Request Logic.

**REQUEST CONTROL** - This section describes the program interrupt and data channel break facilities. Each of these facilities has separate request and priority networks. Most of this logic is internal to the mN603. However, the designer may choose to use either internal or external BUSY and DONE logic.

**SYSTEM REQUIREMENTS** - This section covers the IOC's power-up and initialization requirements. The Power-Up portion discusses the sequence required when first applying power and the clock to the chip. The Initialization portion discusses how synchronization with the CPU is achieved and how an IOC's internal registers are initialized.



# mN603

## PERIPHERAL PORT

### PERIPHERAL PORT

The Peripheral Port transfers data between an I/O Controller and the associated peripheral device. The port consists of 16 bi-directional data lines (D<0-15>), four function code control lines (F<0-3>), and a function code clock (FSTROBE).

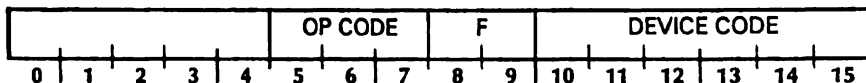
#### DATA LINES

The data lines transfer data to or from an IOC in a 16 bit parallel format. Data may be interpreted using either positive or negative logic; selectable during initialization. Some of the data lines are used to load initialization information (see System Requirements, Initialization). Each data line can drive one Schottky TTL load.

#### CONTROL LINES

The function code lines, with their clock, either indicate how the data lines are to be used or provide control pulses. These lines react in particular ways according to the information received from the I/O bus by the I/O Data Port. In particular, there are 9 I/O Instructions and two types of Data Channel Transactions which invoke specific responses from the Peripheral Port.

The four function code lines specify one of 16 possible codes at any one time. Most of these function codes are generated as a result of I/O instructions received from the CPU via the I/O Data Port. The I/O instruction format, the Op Code and F field mnemonics, and their corresponding bit patterns are shown below:



I/O INSTRUCTION	OP CODE (BITS 5,6,7)	F FIELD (BITS 8,9)
NO I/O Transfer	NIO - 000	NONE - 00
DATA IN A	DIA - 001	S(STRT) - 01
DATA OUT A	DOA - 010	C(CLR) - 10
DATA IN B	DIB - 011	P(IOPLS) - 11
DATA OUT B	DOB - 100	
DATA IN C	DIC - 101	
DATA OUT C	DOC - 110	
I/O SKIP	SKP - 111	

**NOTE:** The I/O Skip instruction is listed above for completeness; however, no function codes are generated as a result of this instruction. Likewise, no specific function code is generated by a No I/O Transfer instruction.

There are three I/O instructions not listed above which follow a slightly different format (see I/O Data Port, Programmed I/O Instructions). Of these three, the Interrupt Acknowledge instruction generates no function code. However, the Mask-out and I/O Reset instructions each produce a specific function code. Data Channel Transactions account for the four remaining function codes.

# mN603

## CONTROL LINES

The various function codes, their labels, and their purpose are listed below:

### FUNCTION CODE TABLE

FUNCTION CODE F<0-3>	LABEL	FUNCTION	FUNCTION CODE F<0-3>	LABEL	FUNCTION
0000	DOA	Load the A register with data from IOC data lines D <0-15>.	1001	IORST*	Gate device code, Polarity, External Register Enable, and External BUSY/DONE Enable bits into IOC via the following data lines: D <0-15> = DEV CODE D9 = POLARITY D8 = EXT REG ENB D7 = EXT BUSY/DONE ENB  NOTES: This data is always interpreted using negative logic (high = 0). See System Requirements, Initialization.
0001	DIA	Gate data into IOC from the A register via data lines D <0-15>.			
0010	DOB	Load the B register with data from IOC data lines D <0-15> (see Internal Structure, Data Channel Registers).			
0011	DIB	Gate data into IOC from the B register via data lines D <0-15>. If internal data channel registers are being used, D <0-15> are ignored (see Internal Structure, Data Channel Registers).	1010	MSKO	Gate device maskout priority bit into IOC via data lines D <0-15>.
0100	DOC	Load the C register with data from IOC data lines D <0-15> (see Internal Structure, Data Channel Registers).	1011	DCHA	Gate data channel direction bit into IOC via data line D0. If external registers are being used, gate external Address Register into IOC via data lines D <0-15>.
0101	DIC	Gate data into IOC from the C register via data lines D <0-15>.	1100	DCHI	Gate data from peripheral data register onto IOC's data lines (D <0-15>) during a Data Channel In Transaction.
0110	STRT	Start I/O device.	1101	DCHO	Load the peripheral data register with data from the IOC's data lines (D <0-15>) during a Data Channel Out Transaction.
0111	CLR	Clear I/O device.	1110	WCEZ	Indicates that the internal data channel Word Count Register has overflowed (equals zero). Denotes the end of a block data channel transfer.
1000	IOPLS	I/O pulse.	1111	NOP	No function.

\* Should be used to clear all external registers and flags (Busy/Done flags, word count register and address register).

DO-0411

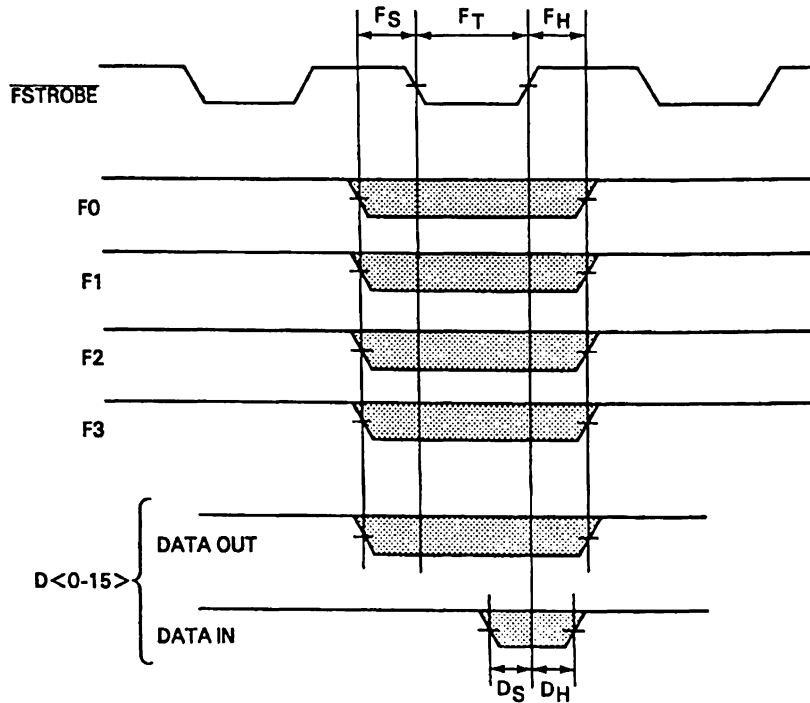


# mN603

## CONTROL LINES

### PERIPHERAL PORT TIMING

Function codes are valid only when  $\overline{\text{FSTROBE}}$  is low. Each function code lasts for one cycle of  $\overline{\text{FSTROBE}}$ . If either the data lines are not being used by the IOC or control pulses are not occurring, the NOP code is transmitted. If the data lines are being used to transfer data to the IOC, they are sampled on the rising edge of  $\overline{\text{FSTROBE}}$ . Timing constraints are shown below:



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.
- $\text{D}\langle 0-15 \rangle$  MUST NOT BE PULLED LOW AT ANY TIME OTHER THAN THAT INDICATED BY  $\text{D}_S$  AND  $\text{D}_H$  IN THE DIAGRAM ABOVE.

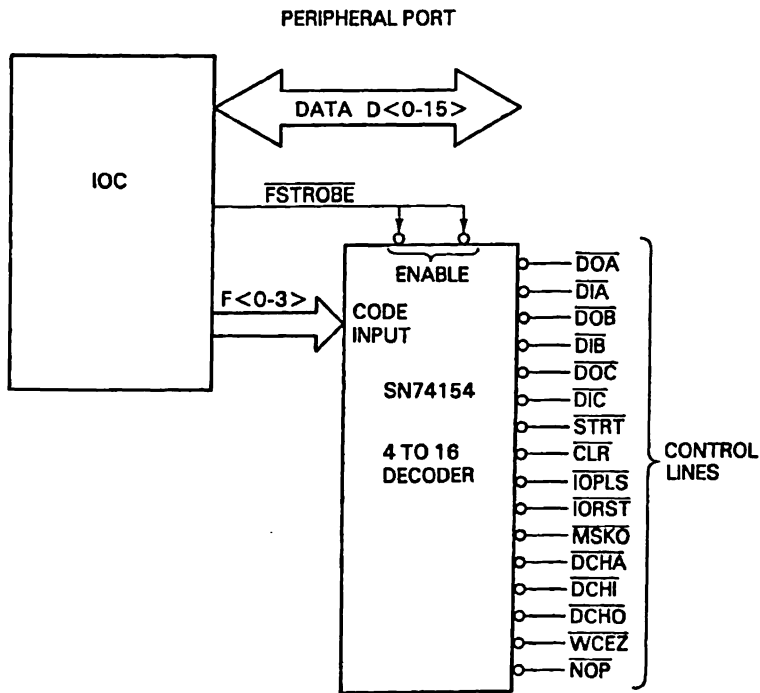
DG-04129

# mN603 CONTROL LINES

## DECODING THE CONTROL LINES

There are many ways to decode the four bit code produced by the four function code lines into separate control lines. Described below are two such methods:

The first approach incorporates a 4 to 16 decoder (TI SN74154 or equivalent) which decodes the 4 bit function codes into individual control lines.  $\overline{FSTROBE}$  should enable the decoder when the function codes are valid. In the arrangement shown below, only one control line is asserted during each cycle of  $\overline{FSTROBE}$ :



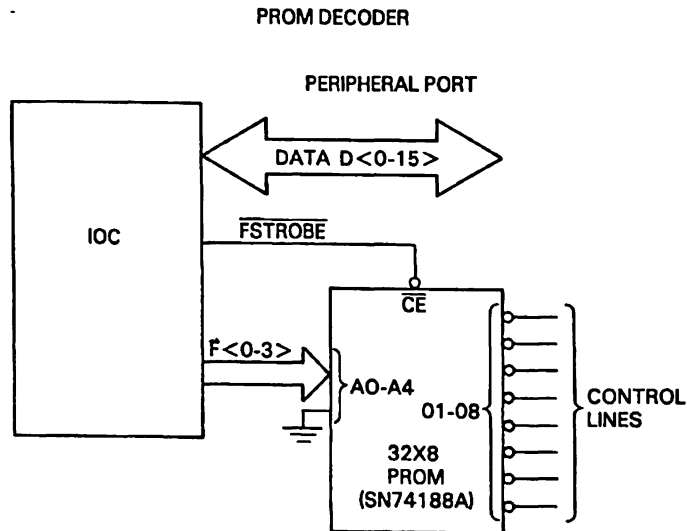
DG-04140

# mN603

## CONTROL LINES

A second approach is to use one or two PROM(s) (TI SN74188A or equivalent) to decode the function code lines. The advantage of using PROMs is that they allow two or more function codes to assert the same control line without requiring additional logic. For example, when using external BUSY and DONE logic, it is frequently necessary to clear both flags when either a CLR function code or an IORST code is issued. PROMs allow full flexibility in decoding both these function codes.

In the application shown below,  $\overline{\text{FSTROBE}}$  enables the outputs of the PROM by pulling the chip enable ( $\overline{\text{CE}}$ ) input low. Only four of the five address lines are used.



DG-04141

Most of the control lines resulting from the decoding of F <0-3> are used to control the flow of the data to and from the Peripheral Port Data lines. These data lines are used when an IOC executes certain programmed I/O instructions or when it performs Data Channel Breaks. The following sections explain how the Peripheral Port reacts during the execution of these instructions and commands.

# mN603

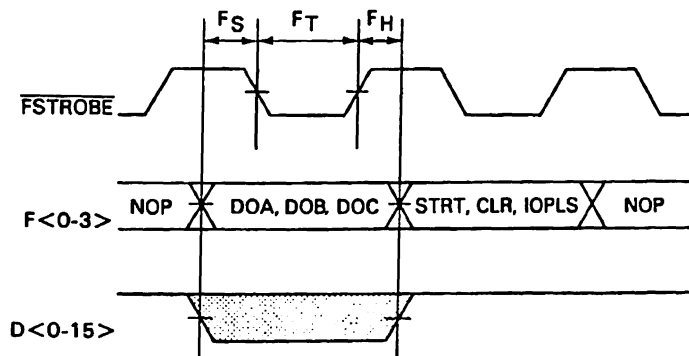
## PROGRAMMED I/O INSTRUCTIONS

### PROGRAMMED I/O INSTRUCTIONS

#### Data Out Instructions

Data Out Instructions (DOA, DOB, and DOC) perform a transfer of up to 16 bits of data from an IOC to a peripheral register. The IOC places the data on D<0-15> when the DOA, DOB, or DOC code is valid. These codes may be used to strobe information on D<0-15> into the appropriate register (A, B, or C).

In addition, one of three function codes (STRT, CLR, or IOPLS), if coded in the instruction, is issued during the next FSTROBE cycle. These codes may be used to control the state of the peripheral device (see Request Control, External Busy/Done Logic).



**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DC-0112

**NOTE:** The No I/O Transfer Instruction is similar to a Data Out Instruction in that a STRT, CLR, or IOPLS function code is issued if coded in the original instruction. However, the NOP function code is always issued in place of the DOA, DOB, or DOC codes.

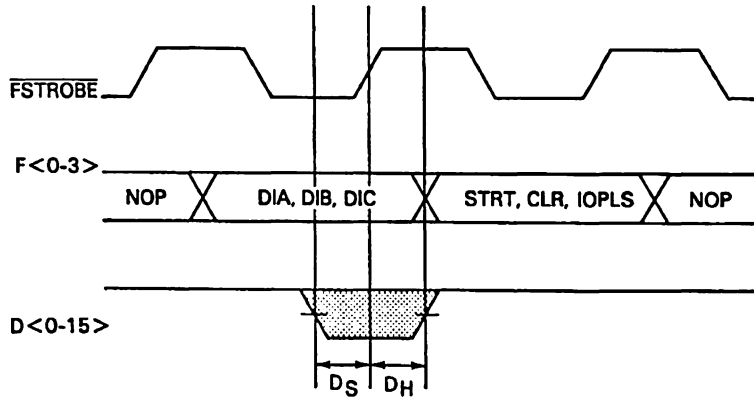
# mN603

## PROGRAMMED I/O INSTRUCTIONS

### Data In Instructions

Data In Instructions (DIA, DIB, and DIC) perform a transfer of up to 16 bits of data from a peripheral register to an IOC. The function codes may be used to strobe information from the respective peripheral register (A,B, or C) to D <0-15>. The IOC samples D <0-15> on the rising edge of  $\overline{\text{FSTROBE}}$  when these codes are valid.

In addition, one of three function codes (STRT, CLR, and IOPLS), if coded in the instruction, is issued during the next  $\overline{\text{FSTROBE}}$  cycle. These codes may be used to control the state of the peripheral device (see Request Control, External Busy/Done Logic).



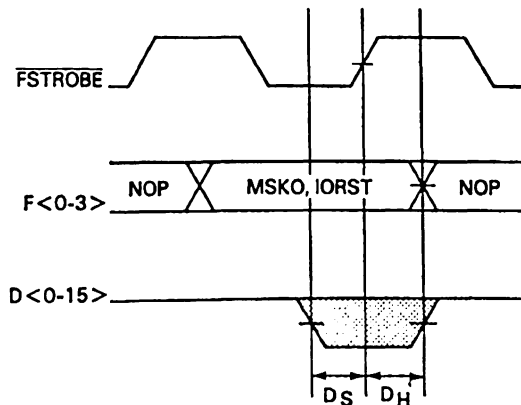
#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0113

### Maskout and I/O Reset Instructions

The Maskout and I/O Reset Instructions both require the transfer of data to an IOC. The IOC samples D <0-15> on the rising edge of  $\overline{\text{FSTROBE}}$  when either the MSKO or the IORST code is valid. Therefore, these codes should be used to gate data onto these lines (for MSKO see Internal Structure, Interrupt Request Logic; for IORST see System Requirements, Initialization).



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

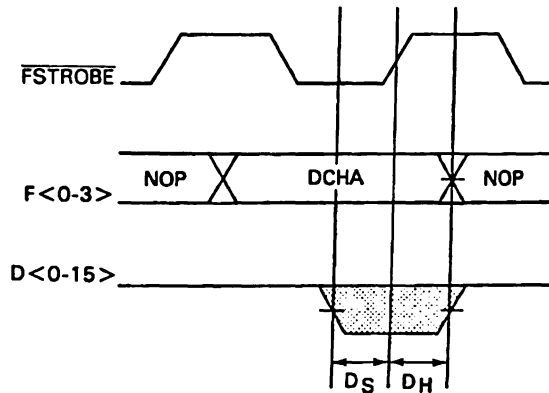
DG-0114

## DATA CHANNEL TRANSACTIONS

### DATA CHANNEL TRANSACTIONS

The Data Channel transfers blocks of data between memory and a peripheral device via the I/O bus and CPU. Each word of a block requires a complete Data Channel Transaction. A Data Channel Transaction requires two information transfers between a peripheral device and an IOC. The first transfer of the pair is to the IOC. This first transfer specifies the direction of the second transfer and may provide the address of the memory location to be accessed. The second transfer of the pair is the data.

During the first transfer of the pair, the IOC samples D <0-15> on the rising edge of  $\overline{FSTROBE}$  when the DCHA function code is valid. DCHA may be used to gate the direction bit to D0 and the Data Channel Address to D <1-15>. The data on D <1-15> is ignored if the IOC has been initialized to use the internal Address Register (see Internal Structure, Data Channel Registers).



**NOTES:**

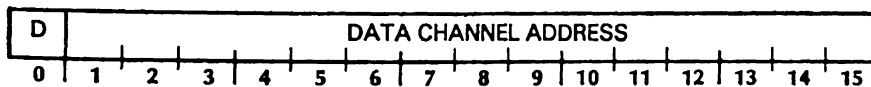
- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.

DG-0145

- SEE TIMING TABLE ON PAGE 43.

If the direction bit is a 0, a Data Channel Out Transaction is performed. This transfers a 16 bit word "out" from the memory location specified by the Data Channel Address to the peripheral device. If the direction bit is a 1, a Data Channel In Transaction is performed. In this case, a 16 bit word is transferred from the peripheral device "in" to the memory location specified by the Data Channel Address.

### DATA CHANNEL CONTROL WORD

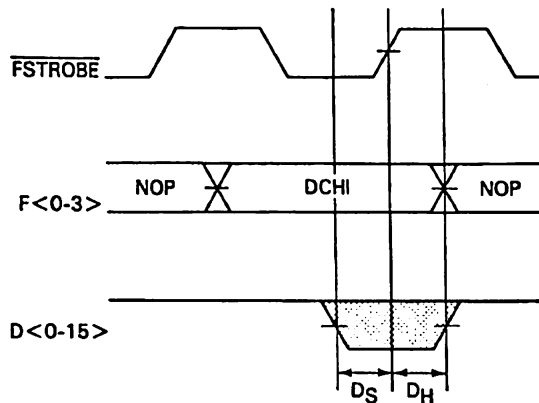


- D = Direction Bit
- 1 = Data Channel In Transaction
- 0 = Data Channel Out Transaction

# mN603

## DATA CHANNEL TRANSACTIONS

If the first transfer specifies a Data Channel In Transaction, the DCHI code is issued during the second transfer. This code may be used to strobe the data word onto D < 0-15 >. The IOC samples these lines on the rising edge of  $\overline{FSTROBE}$  when the DCHI code is valid.

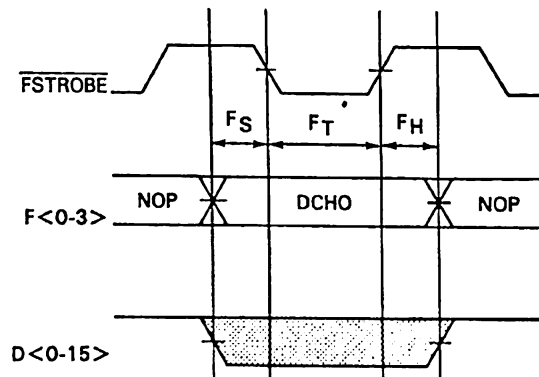


**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0116

If the first transfer specifies a Data Channel Out Transaction, the DCHO code is issued during the second transfer. The IOC places the contents of the second data transfer on D < 0-15 > when the DCHO code is valid. This code may be used to strobe the data word into the appropriate peripheral register.



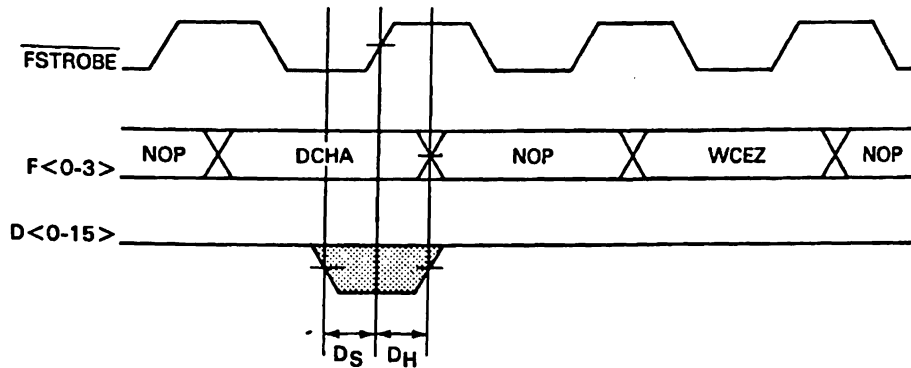
**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0117

## DATA CHANNEL TRANSACTIONS

When the last transaction of a block transfer begins, the internal word count register (if in use) overflows (see Internal Structure, Data Channel Registers). The WCEZ (word count equals zero) function code is issued as shown below. This code may be used to set the DONE flag signaling the completion of a block transfer.



**NOTES:**

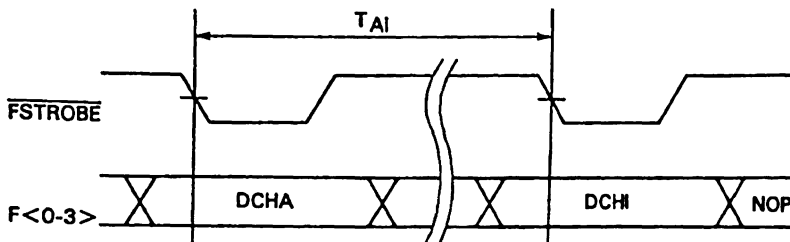
- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-01148

### Data Channel Timing

The minimum interval between the time a DCHA function code is issued and the time a DCHI or DCHO function code is issued is given below. This assumes that the IOC requesting Data Channel service is the highest priority device (see Request Control, Priority Networks).

### DATA CHANNEL IN TRANSACTION



NOTE:  
ANOTHER DCHA OR OTHER FUNCTION CODE  
MAY OCCUR HERE, SEE INTERNAL  
STRUCTURE, DATA CHANNEL LOGIC

**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

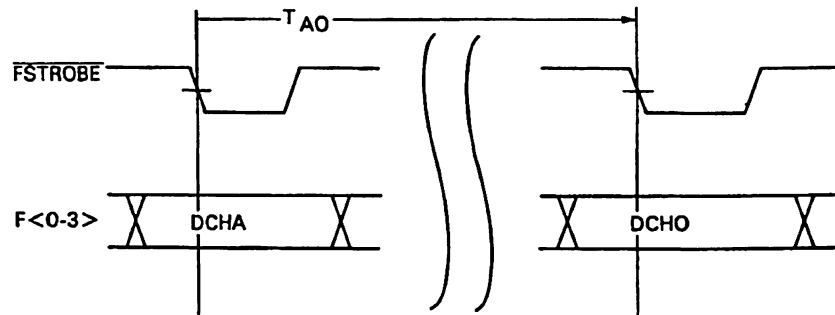
DG-01149



# mN603

## DATA CHANNEL TRANSACTIONS

### DATA CHANNEL OUT TRANSACTION



NOTE:  
ANOTHER DCHA OR OTHER FUNCTION CODE  
MAY OCCUR HERE, SEE INTERNAL  
STRUCTURE, DATA CHANNEL LOGIC

#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DC-04150

The maximum interval of time is calculated by adding the execution time of the longest instruction (excluding Multiply and Divide) to the minimum interval shown above. A table listing the mN601 (CPU) instruction execution times is given under that chip's section.

### I/O DATA PORT

The following section describes the operation of the I/O Data Port and the protocols it uses. Although most designers will concern themselves with only the Peripheral Port, the following description of the I/O Data Port provides a better understanding of the IOC's interaction with the system.

The I/O Data Port includes two serial data lines (I/O DATA <1,2> ), their strobe (I/O CLOCK), and the transmit/receive control line (I/O INPUT). The data lines transfer information between an IOC and the CPU (mN601) via the I/O bus. IOC and CPU I/O Transceivers (mN636 and mN629) are used to drive these data lines and their clock on the differential I/O bus. I/O CLOCK, generated by the transmitting device, strobes information into the receiving device. I/O INPUT, generated by the IOC, selects either the transmit mode (when low) or the receive mode (when high) of the associated Transceiver. The IOC monitors the I/O bus by remaining in receive mode unless required to respond to the CPU.

#### INFORMATION TYPES

Four types of information are received by the I/O Data Port; Request Enable, Data Channel Address Request, I/O Instructions, and Data. However, only data is transmitted by this port.

These information types are transferred in one of two formats; short and long. Both Request Enable and Data Channel Address Request use the short format which requires only one I/O CLOCK pulse. I/O Instructions and Data use the long format which requires five I/O CLOCK pulses. In all cases, the first bits of a transfer on the I/O DATA <1,2> lines make up a two bit code indicating the information type. The encoding of these first bits is shown below:

FIRST BIT I/O DATA1	FIRST BIT I/O DATA2	INFORMATION TYPE
1	1	REQUEST ENABLE
1	0	DATA CHANNEL ADDRESS REQUEST
0	1	DATA
0	0	I/O INSTRUCTION

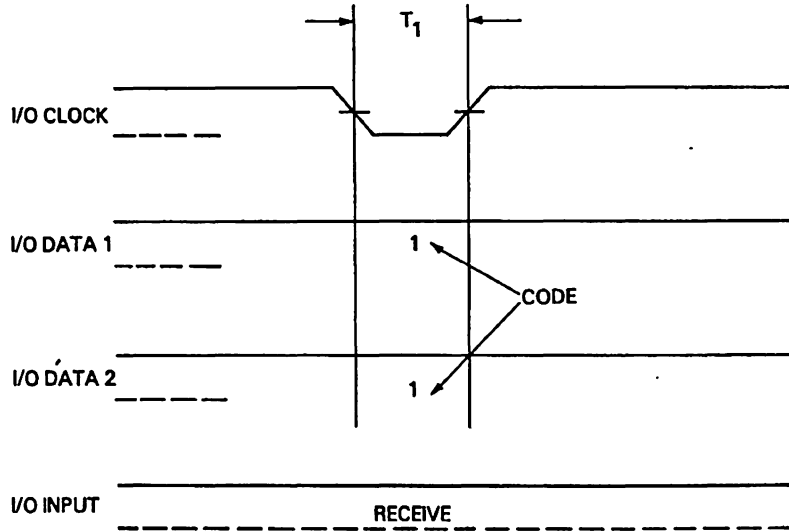
DG-0416

# mN603

## INFORMATION TYPES

### Request Enable

Request Enables are issued at intervals by the CPU. Only the two code bits are transferred. They synchronize program interrupt requests and data channel requests with the CPU. This is necessary to ensure that the request lines to the CPU and priority lines are stable when they are sampled. Request Enables are also used for IOC initialization (see the System Requirements Initialization).



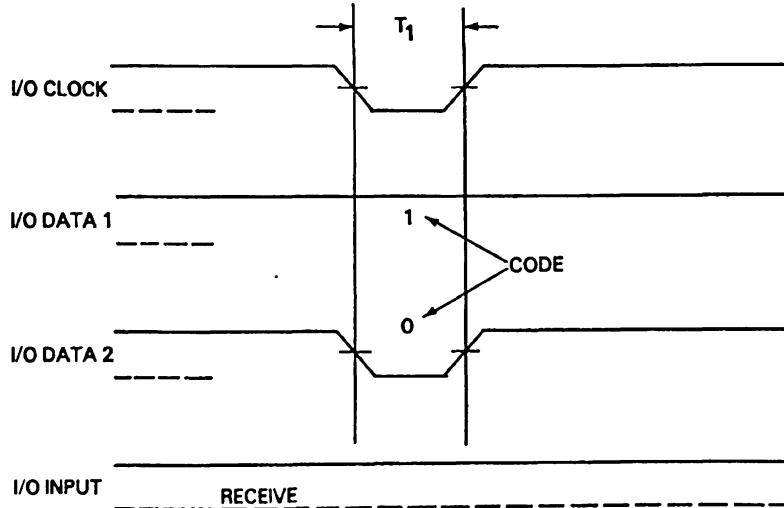
#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0151

### Data Channel Address Request

Data Channel Address Requests are issued by the CPU when a data channel break is executed. Only the two code bits are transferred. They allow the highest priority IOC requesting data channel service to perform a data channel transaction. In addition, like Request Enable, they synchronize program interrupt requests and additional data channel requests for all IOCs (see Request Enable and Data Channel Transaction sections).



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

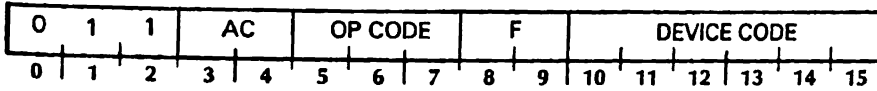
DG-0152

# mN603

## I/O INSTRUCTIONS

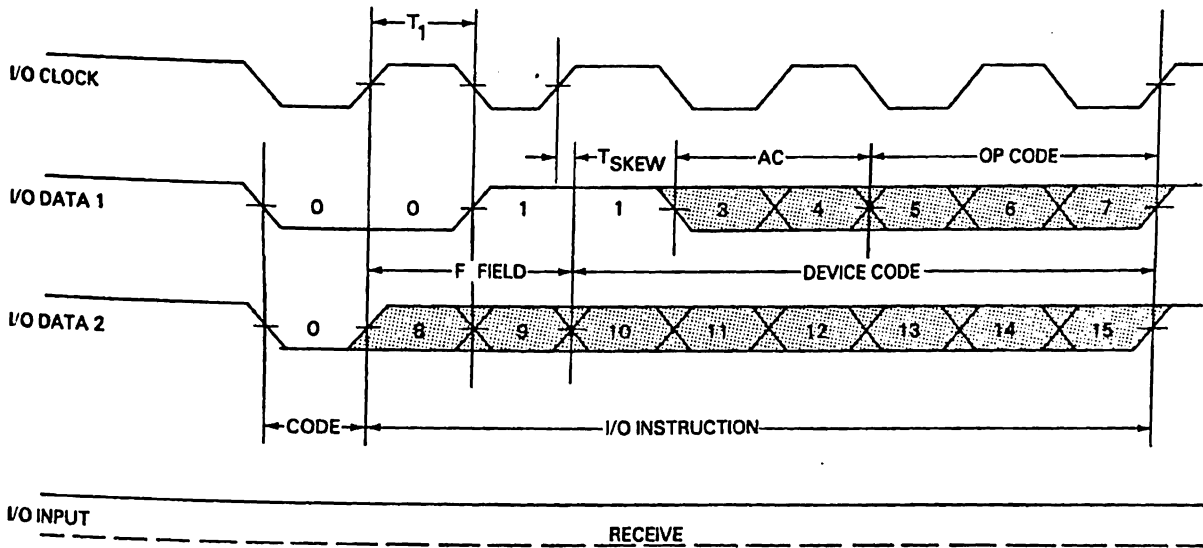
### I/O Instructions

I/O instructions are issued by the CPU. Eighteen bits are transferred; two code bits and a 16-bit instruction. This is the instruction exactly as fetched from memory by the CPU. The eight high order bits of the instruction (0-7) are transferred via the I/O DATA1 line while the eight low order bits (8-15) are transferred via the I/O DATA2 line. The instruction format is shown in the bit pattern below:



AC = accumulator, see mN601  
F = F field

The device code contained in the last six bits of the instruction is checked to see if the IOC should execute the instruction. An IOC executes instructions containing its own device code as well as some of those with device code 77<sub>8</sub> (which select all IOCs). Device codes 00<sub>8</sub> through 03<sub>8</sub>, and 77<sub>8</sub> are ILLEGAL DEVICE CODES for any device connected to the microNOVA I/O bus.



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

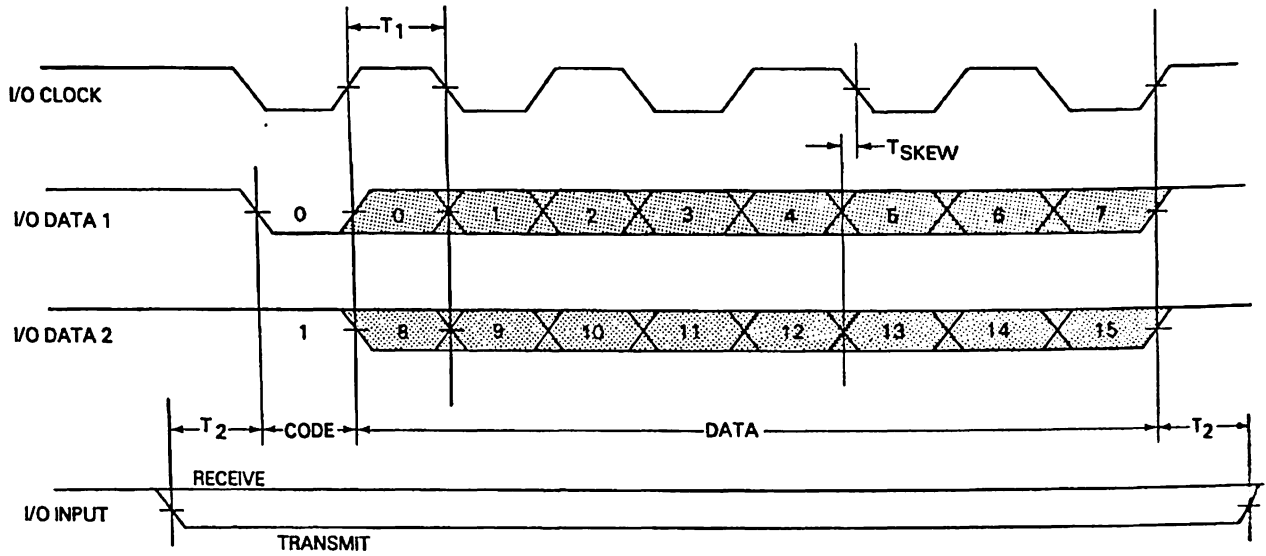
DG 0113

# mN603

## DATA TRANSFERS

### Data

Data is transmitted by either the CPU or an IOC. Eighteen bits are transferred; two code bits and a 16 bit data word. The I/O DATA1 line transfers the eight high order bits of the data word while the I/O DATA2 line transfers the eight low order bits. Data received by an IOC is ignored unless required by a Programmed I/O Transaction or a Data Channel Transaction. Likewise, data is transmitted only if required by one of these transactions.



### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0134

### TRANSACTION PROTOCOLS

The four types of information transfers discussed above are combined to form three types of transactions; Request Enable Transactions, Programmed I/O Transactions, and Data Channel Transactions. Each type of transaction has its own protocol. Request Enables occur as a single information transfer. Programmed I/O Transactions combine an I/O Instruction transfer with a Data transfer. Data Channel Transactions consist of a Data Channel Address Request followed by two Data transfers.

**NOTE:** There is a minimum time between two consecutive transactions. This is indicated in the timing diagrams by  $T_N$ .

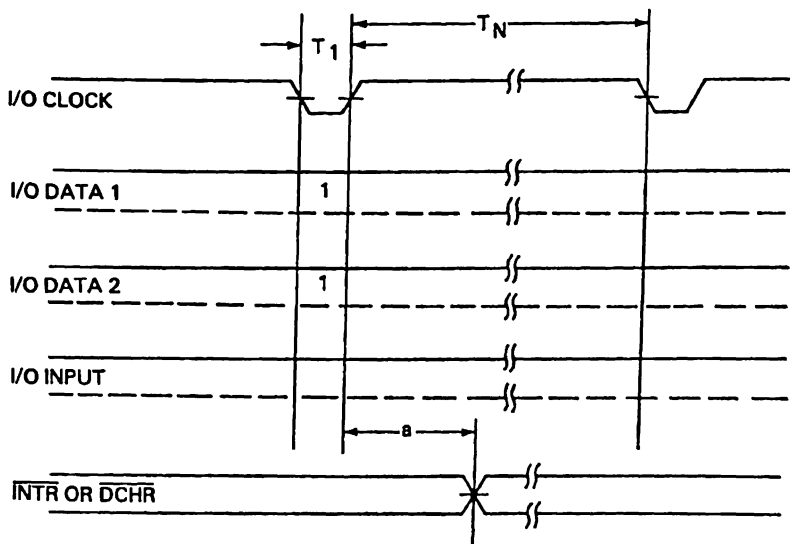
### Request Enable Transactions

Request Enables are used by an IOC to synchronize the assertion and release of program interrupt and data channel requests. If a peripheral device requires service, a Request Enable allows the associated IOC's interrupt request line ( $\overline{\text{INTR}}$ ) and/or data channel request line ( $\overline{\text{DCHR}}$ ) to be asserted, signaling the CPU for service. When service is received, a Request Enable allows these lines to be released. It is necessary that the transition of these lines occur at specific times to ensure their stability when sampled by the CPU. These lines are also cleared asynchronously if the IOC is reset.

One of two conditions must be met for  $\overline{\text{INTR}}$  to be asserted during a Request Enable. Either the peripheral device pulls the IOC's  $\overline{\text{INT SYNC}}$  line low or it sets the DONE flag. However, an internal Interrupt Disable Flag can block a program interrupt request; see the section on Request Control.

In order for  $\overline{\text{DCHR}}$  to be asserted during a Request Enable, the peripheral device must pull the IOC's  $\overline{\text{DCH SYNC}}$  line low.

**NOTE:** The first two Request Enables received after an IOC is powered up or reset initialize its internal four phase clock and internal registers. As a result, the IOC is synchronized with the CPU (see System Requirements, Initialization).



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DG-0133

# mN603

## PROGRAMMED I/O TRANSACTIONS

### Programmed I/O Transactions

There are two types of Programmed I/O Transactions:

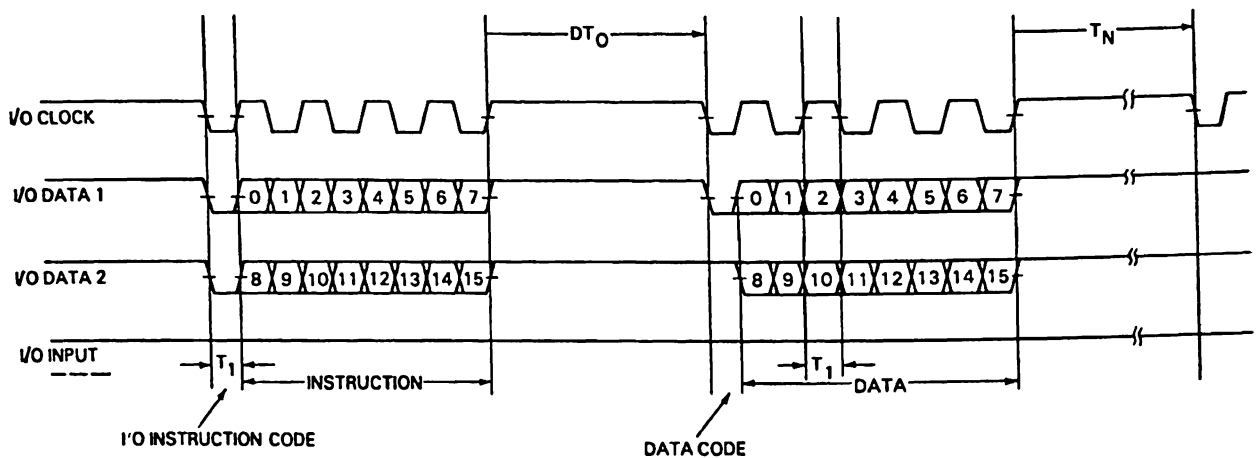
- 1) CPU to IOC Data Out Transaction
- 2) IOC to CPU Data In Transaction

Each type of transaction transfers a word of data between an IOC and the CPU.

#### 1) CPU to IOC Data Out Transaction

Data Out Transactions transfer a 16 bit word of data from the CPU to one or all IOCs. The transactions begin with an I/O Instruction transfer, issued by the CPU and are followed by a Data transfer, also issued by the CPU. The instruction transmitted is the I/O Instruction executed by the CPU, exactly as fetched from memory.

An IOC executing a Data Out Instruction must receive the Data transfer during a fixed interval of time after it has received the I/O Instruction transfer. This time interval is illustrated in the following timing diagram by  $DT_0$ .



#### NOTES.

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

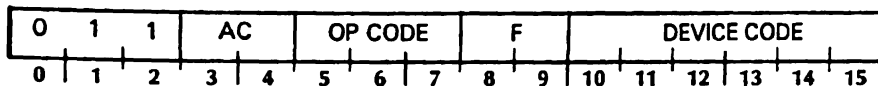
DC 04156

# mN603

## PROGRAMMED I/O TRANSACTIONS

There are two groups of instructions which use the Data Out format. The first group consists of four I/O Instructions which are executed by only one IOC per transaction. The second group consists of two I/O Instructions which are executed by all IOCs simultaneously.

The first group selects individual IOCs by specifying a particular device code. The bit patterns for these I/O Instructions are given below. The instruction is the same as fetched from memory by the mN601 CPU.



AC = see mN601, CPU  
 F = see Internal Structure  
 Op Code = see below

DOA (Data Out A) .....Op Code = 010  
 DOB (Data Out B) .....Op Code = 100  
 DOC (Data Out C) .....Op Code = 110

DOA, DOB, and DOC transfer data from the CPU to a specific peripheral device via an IOC. The data received by the selected IOC is placed on its Peripheral Data Port to be gated into one of three peripheral registers, A, B, or C. For more information on how these registers are loaded, see Peripheral Port.

In addition, the state of two internal flags (BUSY and DONE) are controlled by these instructions (see Internal Structure, Internal BUSY/DONE Logic).

**NOTE:** Two internal registers, used during Data Channel Breaks, are loaded using the DOB and DOC instructions (see Request Control, Data Channel Registers).

NIO (No I/O) .....Op Code = 000

The No I/O Transfer Instruction is a Data Out instruction in which data may be transferred from the CPU to the IOC. If data is received by the selected IOC, it may be placed on the Peripheral Port data lines. However, this data is unpredictable and no function code is generated.

Like DOA, DOB, and DOC, the NIO instruction can be used to control the state of the BUSY and DONE flags (see Internal Structure, Internal BUSY/DONE Logic).

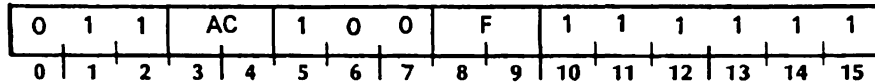


# mN603

## PROGRAMMED I/O TRANSACTIONS

The second group of data out transactions consists of two I/O Instructions which select all IOCs by specifying device code 77<sub>8</sub>. The bit patterns and timing for these two instructions are given below:

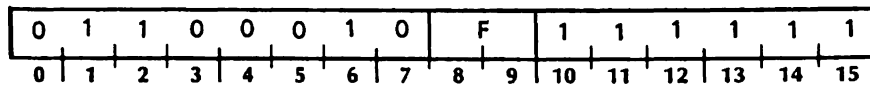
MSKO (Mask Out)  
DOB [i] ac, CPU



AC, F = see mN601, CPU

The Mask Out Instruction transfers a 16-bit mask (data) from the CPU to all IOCs connected to the I/O bus. This data affects the state of each IOC's Interrupt Disable flag. For more information on this flag, see Interrupt Request Logic under Internal Structure.

IORST (I/O Reset)  
DOA [i] 0, CPU



F = see mN601, CPU

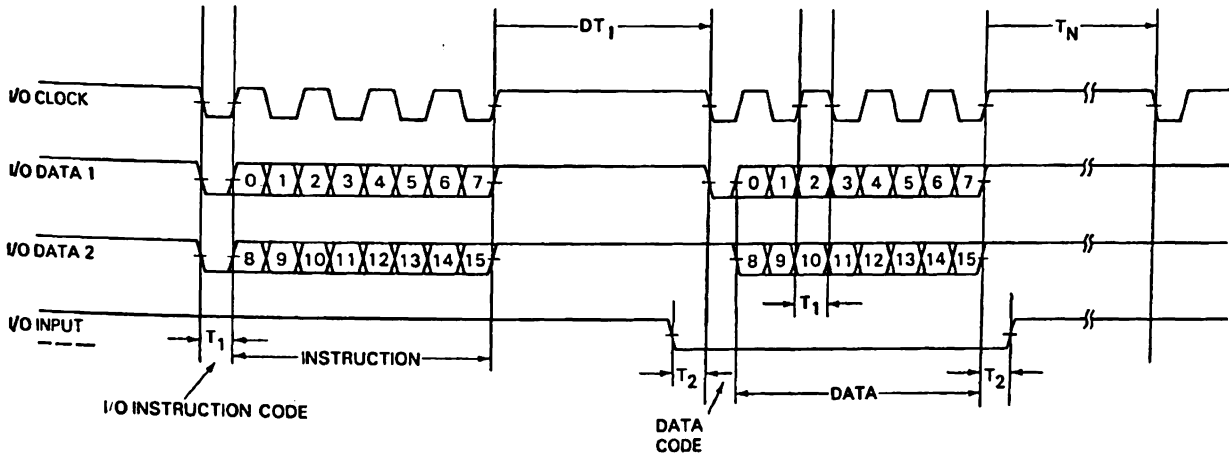
The I/O Reset Instruction is used to initialize all an IOC's internal registers and flags. The instruction portion of this transaction is executed by all IOCs, however, the data portion of the transaction is ignored. For further information see the section titled System Requirements, Initialization.

## PROGRAMMED I/O TRANSACTIONS

### 2) IOC to CPU Data In Transactions

Data In Transactions transfer a 16 bit word of data from an IOC to the CPU. These transactions begin with an I/O Instruction transfer issued by the CPU and are followed by a data transfer issued by the responding IOC.

An IOC executing a Data In Instruction begins transmitting the data within a fixed interval of time after it has received the I/O Instruction transfer. This time interval is illustrated in the following timing diagram by  $DT_1$ .



#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

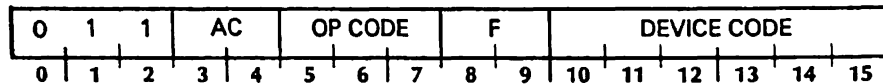
DG-0157

# mN603

## PROGRAMMED I/O TRANSACTIONS

There are two groups of instructions which use the Data In format. The first group consists of four I/O Instructions which select only one IOC per transaction. The second group consists of one I/O Instruction which selects all IOCs simultaneously.

The first group selects individual IOCs by specifying a particular device code. The bit pattern for these instructions is given below:



AC = see mN601, CPU  
F = see Internal Structure  
Op Code = see below

DIA (Data In A).....Op Code = 001  
DIB (Data In B).....Op Code = 011  
DIC (Data In C).....Op Code = 101

DIA, DIB, and DIC transfer data from a specific peripheral device to the CPU via an IOC. The selected IOC gates data into its Peripheral Data Port from one of three peripheral registers (A, B, or C) and transmits this data to the CPU.

In addition, the state of two internal flags (BUSY and DONE) may be controlled by these instructions (see Internal Structure, Internal Busy/Done Logic).

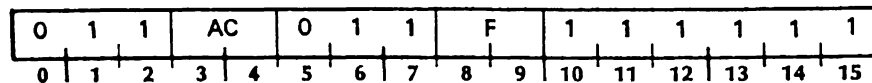
**NOTE:** There is one internal register used during data channel breaks whose contents may be transferred from an IOC to the CPU using the DIB Instruction (see Request Control, Data Channel Registers).

SKP (I/O Skip).....Op Code = 111

The I/O Skip Instruction transfers a word of data containing the state of the BUSY and DONE flags from the selected IOC to the CPU. Bit 0 of this word contains the complement of the state of the DONE flag while bit 1 contains the complement of the state of the BUSY flag. Bits 2-15 contain zeros.

The second group of Data In Instructions consists of the Interrupt Acknowledge Instruction. This instruction incorporates device code 77<sub>h</sub> which selects all IOCs.

INTA (Interrupt Acknowledge)  
DIB [f] ac, CPU



AC, F = see mN601, CPU

The Interrupt Acknowledge Instruction causes the responding IOC to transmit its device code to the CPU. This instruction is issued by the CPU in response to one or more IOCs requesting program interrupts (asserting their INTR line). Only the highest priority IOC responds (see Request Control, Priority Networks).

The responding IOC transmits the contents of an internal register containing its device code to the CPU. Bits 0-9 of the data transfer are zero while bits 10-15 contain the device code.

# mN603

## DATA CHANNEL TRANSACTIONS

### Data Channel Transactions

There are two types of Data Channel Transactions:

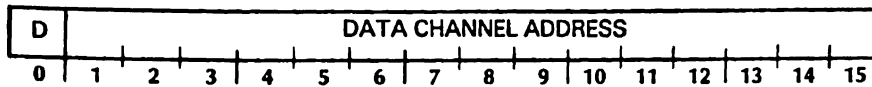
- 1) CPU to IOC Data Channel Out Transactions
- 2) IOC to CPU Data Channel In Transactions

Each type of transaction transfers a 16 bit word of data between a peripheral device and memory via the CPU. The direction of the transfer indicated by the name refers to the CPU.

Both types of Data Channel Transactions begin with a Data Channel Address Request, issued by the CPU and are followed by a Data transfer issued by the responding IOC. The second Data transfer is issued by the CPU or IOC depending on which type of transaction is taking place.

**NOTE:** Though all IOCs receive the Data Channel Address Request, only the highest priority IOC requesting a data channel break completes the transaction (see Request Control section).

The first data transfer, from IOC to CPU, provides the CPU with two kinds of information. Bit 0 of this 16 bit word indicates the direction of the second transfer (data). Bits 1-15 contain the address of the memory location to be accessed.



D = direction bit:  
0 = Data Channel Out Transaction  
1 = Data Channel In Transaction

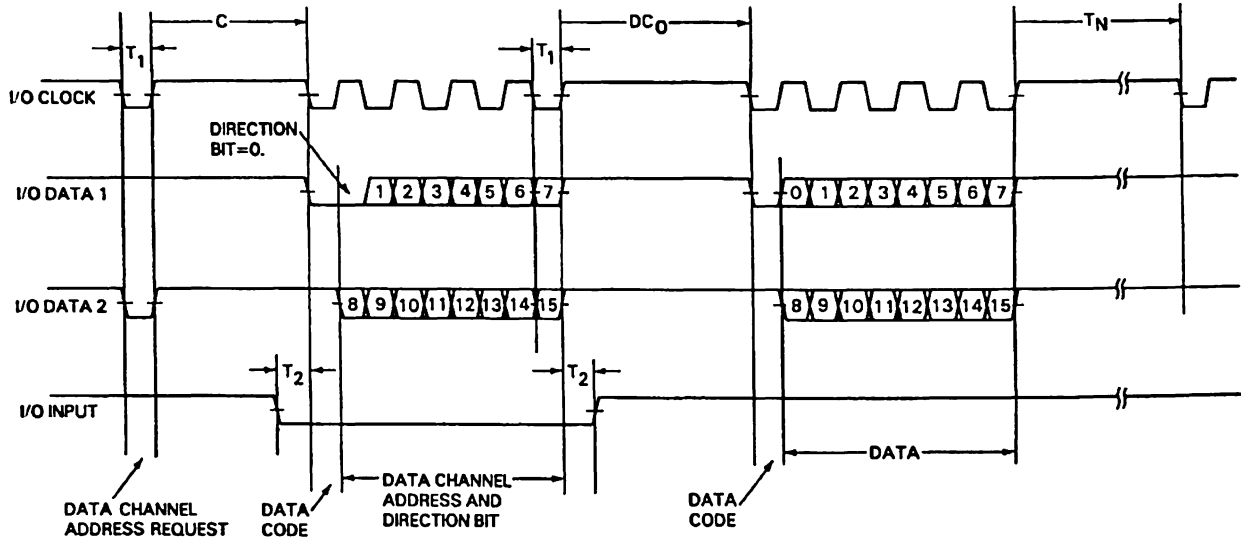
**NOTE:** See Internal Structure, Data Channel Logic.

# mN603

## DATA CHANNEL TRANSACTIONS

### 1) CPU to IOC Data Channel Out Transactions

A Data Channel Out Transaction transfers a word of data from memory "out" to an IOC. The IOC begins transmitting the address and direction bit to the CPU during a set interval of time after the reception of the Data Channel Address Request. This is illustrated by C in the timing diagram. Likewise, the IOC must begin reception of the Data transfer a set interval of time after it transmits the Control Word. This interval is illustrated by  $DC_0$ .



#### NOTES:

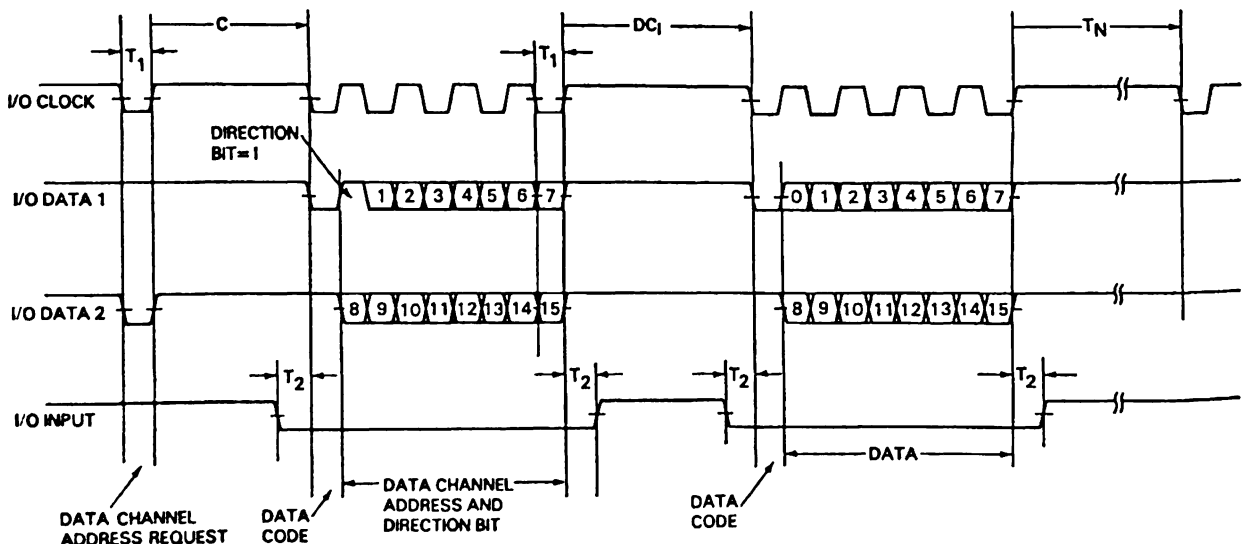
- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.

DG-0415B

- SEE TIMING TABLE ON PAGE 43.

### 2) IOC to CPU Data Channel In Transactions

A Data Channel In Transaction transfers a word of data from an IOC "in" to memory. The IOC begins transmitting the address and direction bit to the CPU a set interval of time after the reception of the Data Channel Address Request. This is illustrated in the timing diagram by C. Likewise, the IOC will begin transmitting the data to the CPU (memory) during a set interval of time after it transmits the Control Word. This interval is illustrated by  $DC_1$ .



#### NOTES

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.

DG-0415A

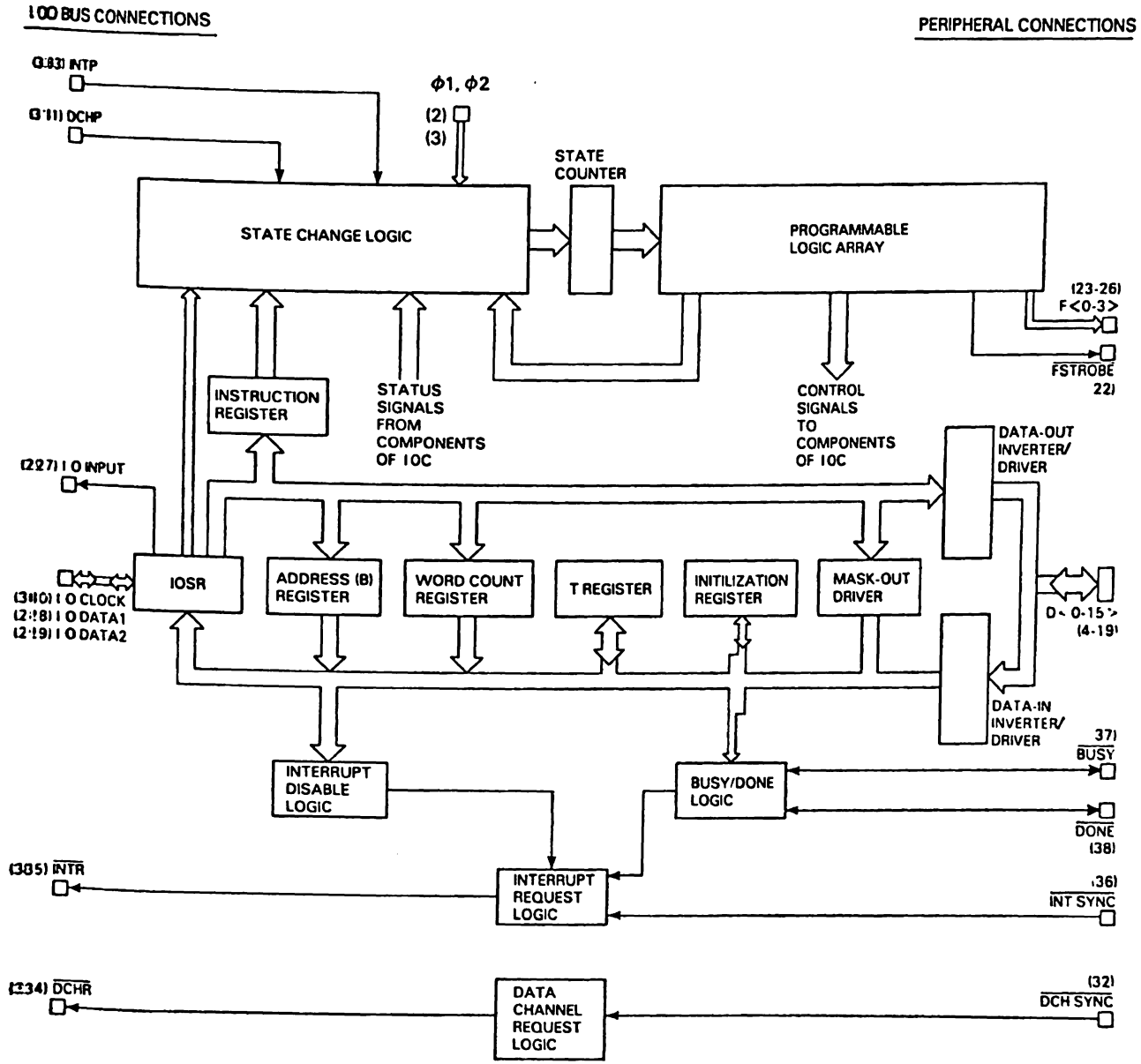
- SEE TIMING TABLE ON PAGE 43.

# mN603

## INTERNAL STRUCTURE

### INTERNAL STRUCTURE

In order to make full use of an IOC's capabilities, an understanding of some of its internal elements is necessary. The block diagram below is a simplified representation of the IOC's internal structure.



DOG-0100

# mN603

## INTERNAL STRUCTURE

Six internal registers are used during transactions as shown in the table below:

REGISTERS	FUNCTION
IOSR (I/O Shift Register)	Performs the serial/parallel conversion of information transferred on I/O DATA <1,2> .
Instruction Register	Holds all I/O instructions received by the IOC.
Initialization Register	Holds the device code, external register enable bit, external BUSY/DONE enable bit, and the polarity bit (see System Requirements, Initialization).
Address Register	Holds the 15-bit memory address used during data channel transactions when internal registers are used.
Word Count Register	Holds the two's complement of the number of words remaining in a data channel block transfer when internal registers are used.
T Register	Buffers the 15-bit memory address and the direction bit used during data channel transfers.

**NOTE:** Each IOC must have a unique device code. An IOC compares the device code contained in its Initialization Register with that received in all I/O instructions. If they match, the instruction is executed. In addition, some instructions specifying device code 77<sub>8</sub> are executed.

### INTERNAL BUSY/DONE LOGIC

The BUSY and DONE flags indicate the state of the peripheral device. In normal use, the BUSY flag being set to 1 indicates the peripheral device is performing an operation; the DONE flag being set to 1 indicates the peripheral device has completed an operation.

If the internal BUSY and DONE flags are active (External BUSY/DONE Enable bit is 0) they are controlled by both programmed I/O instructions and the peripheral device. For those programmed I/O instructions which contain an IOC's selected device code, bits 8 and 9 (flag control or F field) from the Instruction Register affect the BUSY and DONE flags as follows:

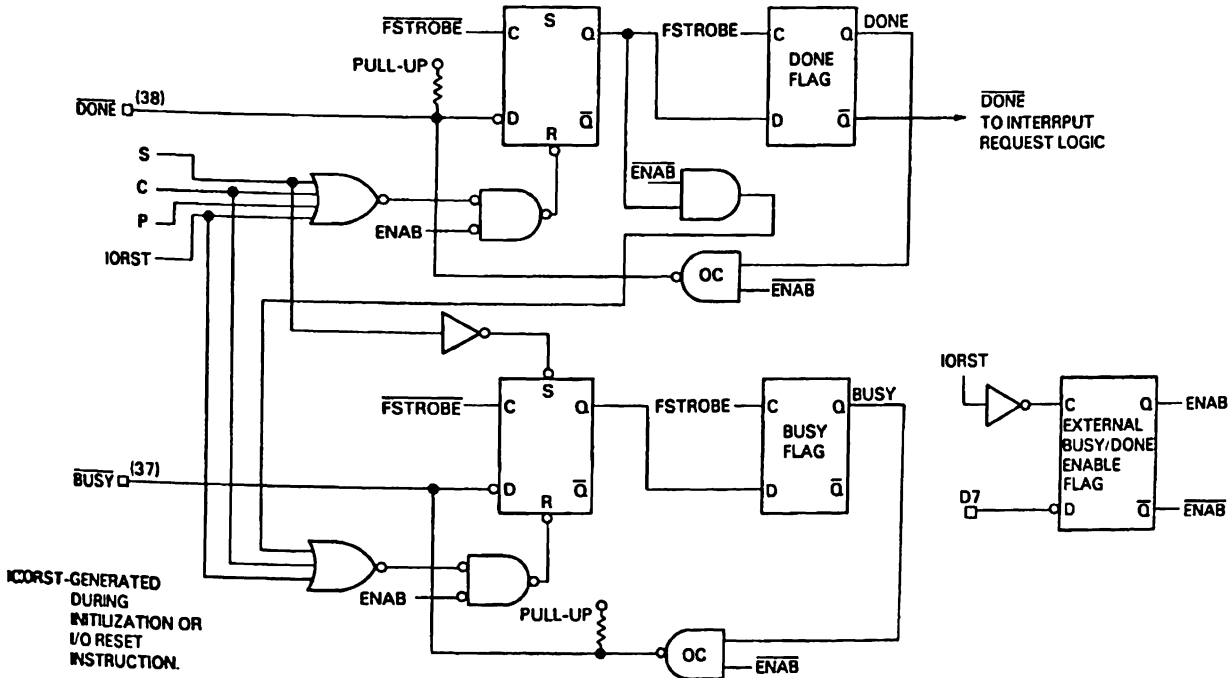
BITS		PROGRAM MNEMONIC	EFFECT
8	9		
0	0	-	No effect
0	1	S	set BUSY to 1, DONE to 0
1	0	C	set both BUSY and DONE to 0
1	1	P	set DONE to 0, no effect on BUSY

DG-0417

# mN603

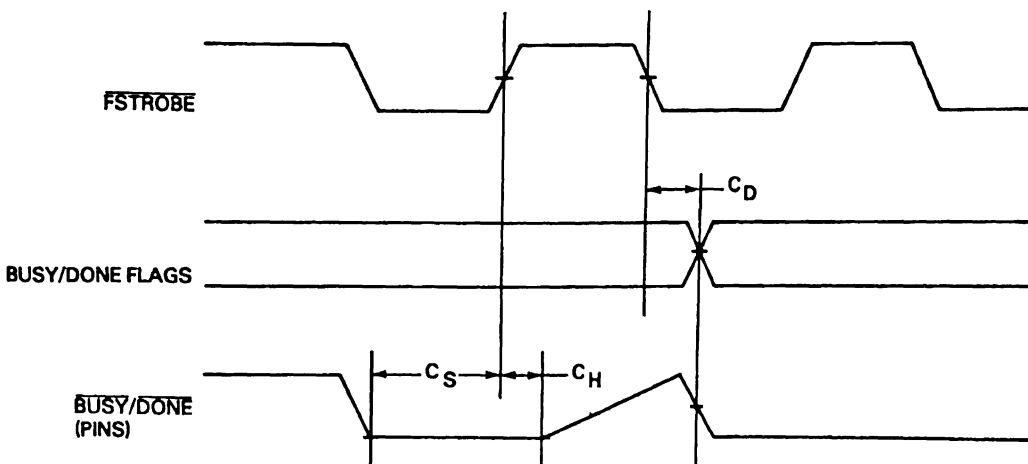
## INTERNAL BUSY/DONE LOGIC

The peripheral device can set either the BUSY or DONE flag to 1. When the DONE flag gets set to 1, the BUSY flag is set to 0. The peripheral device controls these flags by pulling either the  $\overline{\text{BUSY}}$  or the  $\overline{\text{DONE}}$  lines low. Shown below is a representation of the internal BUSY/DONE logic:



DCG-0111

There is a set relationship between the time the  $\overline{\text{BUSY/DONE}}$  lines are asserted by the peripheral device and the time that the internal BUSY/DONE flags are set. The state of the  $\overline{\text{BUSY/DONE}}$  lines are sampled on the rising edge of  $\overline{\text{FSTROBE}}$ . However, the internal flags are not set until the next falling edge of  $\overline{\text{FSTROBE}}$ . If the peripheral device does not hold the  $\overline{\text{BUSY/DONE}}$  lines low, they may float high until the internal flags are set.



**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED BETWEEN POINTS AT 1.5V.
- SEE TIMING TABLE ON PAGE 43.

DC-0112



# mN603

## INTERRUPT REQUEST LOGIC

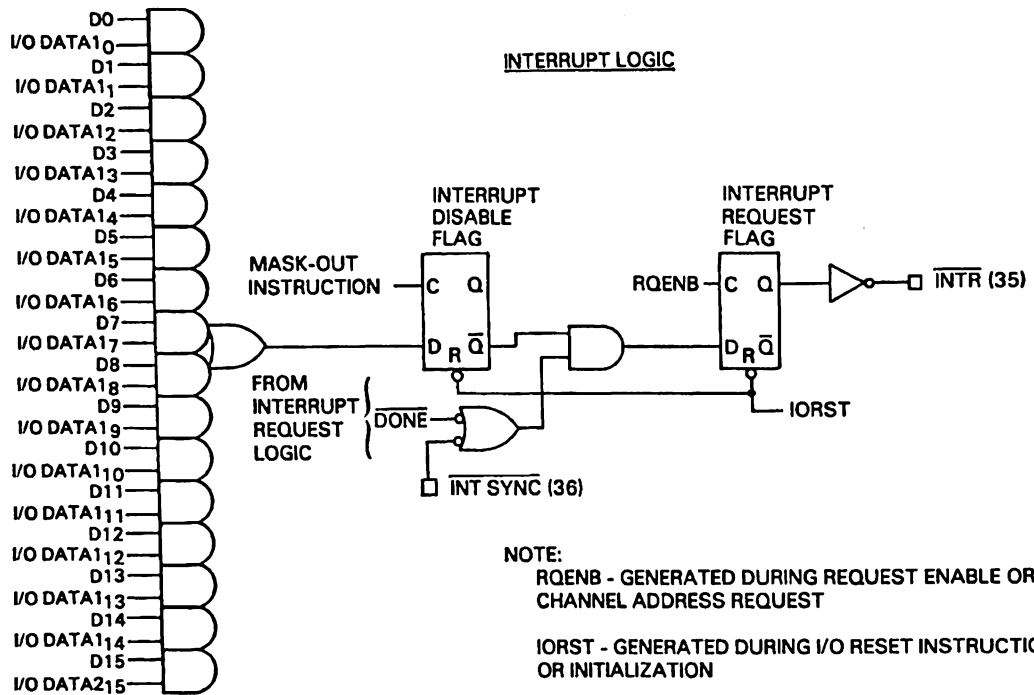
### INTERRUPT REQUEST LOGIC

The Interrupt Request Logic is responsible for requesting program interrupts from the CPU. An IOC requests interrupts by asserting its  $\overline{\text{INTR}}$  line. The state of this line is controlled by a flag called the Interrupt Request Flag. This flag is set or cleared during either a Request Enable or a Data Channel Address Request Transaction. This ensures that the request line ( $\overline{\text{INTR}}$ ) is stable when it is sampled by the CPU. The section titled "Request Control, Interrupt Requests" discusses how this flag may be set.

Each IOC contains an Interrupt Disable Flag which allows the program to disable interrupts from that IOC. When this flag is set to one, the IOC is prevented from asserting the  $\overline{\text{INTR}}$  line.

The Interrupt Disable Flags for all IOCs are manipulated by the Mask-Out Instruction. When this instruction is executed, the 16-bit mask received from the I/O Data Port is logically ANDed with the data received from the Peripheral Port. If any bit of the result is a one, the interrupt Disable Flag is set to one; otherwise, it is set to 0. It is unconditionally cleared by an I/O Reset Instruction.

Shown below is a representation of the Interrupt Request logic:

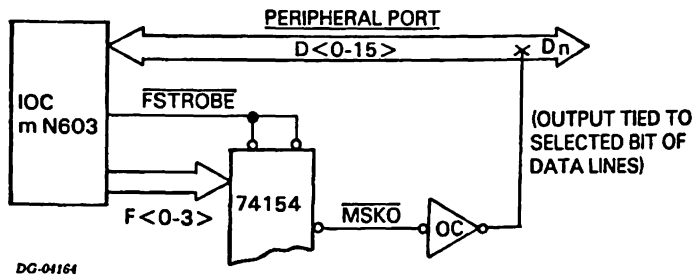


DG-0163

# mN603

## DATA CHANNEL LOGIC

A mask-out bit is selected by pulling one of the data lines low during the execution of the Mask-Out Instruction. The MSKO function code generated during this instruction should be used for this purpose. During the time that the MSKO code is valid, the Data Port is forced to interpret all data using negative logic (low = 1). As a result, the buffered output of the function code decoder may be used to tie the selected data line low for the duration of the code as shown below. (see mN601, Program Interrupts)



### DATA CHANNEL LOGIC

The Data Channel Logic is responsible for transferring device service requests to the CPU and performing data channel transfers. A peripheral device requests data channel service by asserting an IOC's DCH SYNC line. On the Request Enable or Data Channel Address Request following the assertion of  $\overline{\text{DCH SYNC}}$ , the IOC's Data Channel Request Flag is set to 1. This flag controls the state of the data channel request line ( $\overline{\text{DCHR}}$ ). When the flag is set to one, the  $\overline{\text{DCHR}}$  line is asserted requesting service from the CPU. Synchronizing the request in this way ensures the stability of the  $\overline{\text{DCHR}}$  line when it is sampled by the CPU.

Upon receiving data channel service, an IOC must transmit control information about the transfer to the CPU (see I/O Data Port, Data Channel Transactions). If the control information is ready for transmission when service is received, the entire transaction can be accomplished more quickly. Therefore, the internal T Register is loaded with this control information when the service request is made. This shortens the total transaction time.

The first thing an IOC does after receiving data channel service from the CPU (a Data Channel Address Request when the IOC has priority), is to transmit the contents of the T Register to the CPU. At the same time, the IOC checks to see if the peripheral is requesting another transfer ( $\overline{\text{DCH SYNC}}$  asserted). If the peripheral is ready for a second transfer, the T Register is loaded again. However, if no more transfers are being requested, the Data Channel Request Flag is cleared releasing the  $\overline{\text{DCHR}}$  line. In either case, the data word is then transferred completing the first transaction.

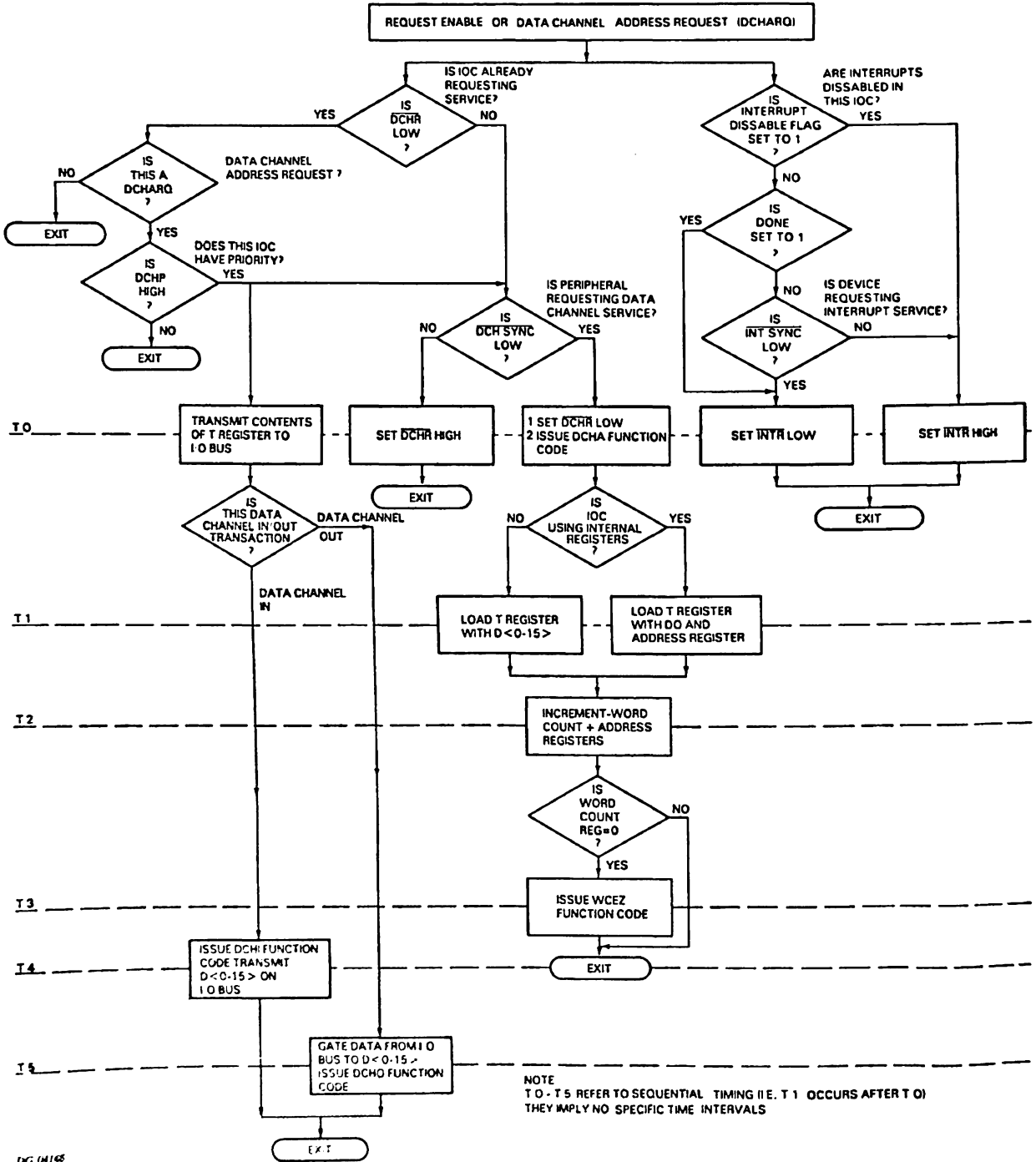
Each time the T Register is to be loaded, a DCHA function code is issued to the peripheral. This function code indicates that the IOC expects the appropriate control information on its data lines as described in the Peripheral Port section. Since the T Register may be loaded twice before the first data word is actually transferred, two DCHA function codes may be issued by the IOC before the DCHI/DCHO code which gates the first data word to/from the IOC. In addition, it is possible for one I/O instruction to be executed between the time data channel service is requested and the time the first transaction occurs. However, once the first transaction commences, the CPU continues to transmit Data Channel Address Requests until all requests are satisfied ( $\overline{\text{DCHR}}$  is released).

# mN603

## INTERNAL SEQUENCING DIAGRAM

### INTERRUPT AND DATA CHANNEL SEQUENCING DIAGRAM

The sequence of events which occur as a result of the IOC receiving a Request Enable or Data Channel Address Request are illustrated in the following flow chart. T0 - T5 refer to the sequential timing of specific events. They in no way imply specific time intervals.



DG 0416

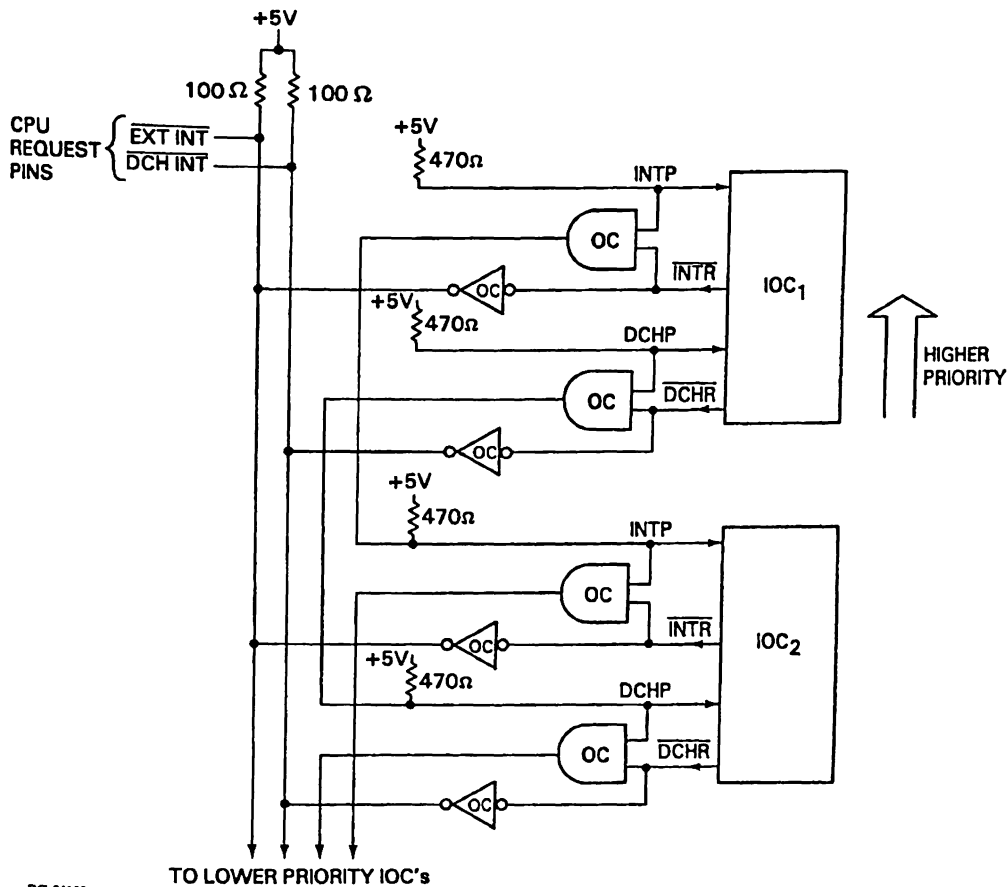
### REQUEST CONTROL

The request control facility uses 6 lines of the IOC; three for program interrupts,  $\overline{\text{INTP}}$ ,  $\overline{\text{INTR}}$ , and  $\overline{\text{INT SYNC}}$  and three for data channel requests, DCHP, DCHR, and DCH SYNC.

As mentioned previously, the IOC may request both interrupt and data channel service from the CPU. In both cases, the peripheral device prompts the IOC to request these services. When two or more IOCs are requesting the same type of service, a mechanism for determining which IOC receives service first is necessary. This mechanism is called the priority network.

#### PRIORITY NETWORKS

Since there are two types of service requests, interrupt and data channel, there are two priority networks. Each network contains a priority line (INTP or DCHP) which is daisy chained from controller to controller. The controller closest to the CPU requiring service can remove priority from all the other controllers further down the priority chain. An IOC has priority for a particular type of service if the appropriate priority pin is high. (See figure below)



DG-0168

# mN603

## REQUEST CONTROL

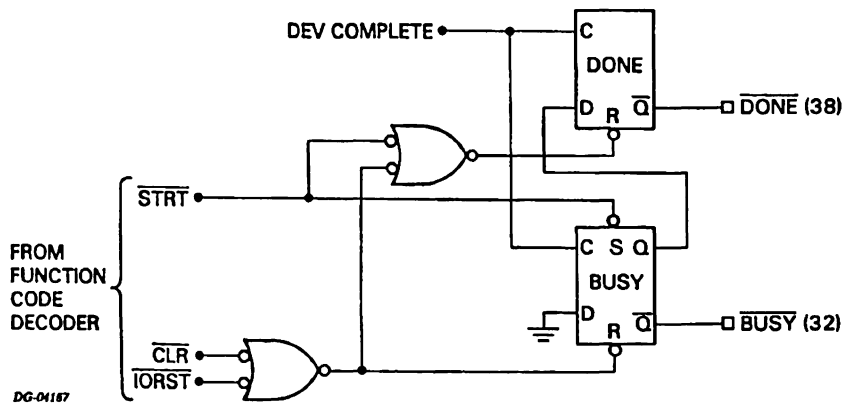
### INTERRUPT REQUESTS

An IOC requests a program interrupt if either its DONE flag is set to one or the peripheral device asserts the IOC's  $\overline{\text{INT SYNC}}$  line (assuming that a Mask-out Instruction has not set the Interrupt Disable Bit). On the Request Enable or Data Channel Address Request following the setting of  $\overline{\text{DONE}}$  or the assertion of  $\overline{\text{INT SYNC}}$ , the IOC requests service (asserts  $\overline{\text{INTR}}$ ). The service request should remain in effect ( $\overline{\text{DONE}}$  set or  $\overline{\text{INT SYNC}}$  asserted) until the CPU honors the interrupt request.

The CPU honors a request by issuing the appropriate I/O instruction. If the request was generated as a result of  $\overline{\text{DONE}}$  being set to one, the instruction should set  $\overline{\text{DONE}}$  to 0 (see Internal Structure, Internal BUSY/DONE Logic). If the request occurred as a result of  $\overline{\text{INT SYNC}}$  being asserted, the instruction should specify a function code which the peripheral should use to release  $\overline{\text{INT SYNC}}$ . In both cases, the IOC stops requesting interrupt service (asserting  $\overline{\text{INTR}}$ ) on the Request Enable or Data Channel Address following reception of the I/O instruction honoring the request.

### EXTERNAL BUSY/DONE LOGIC

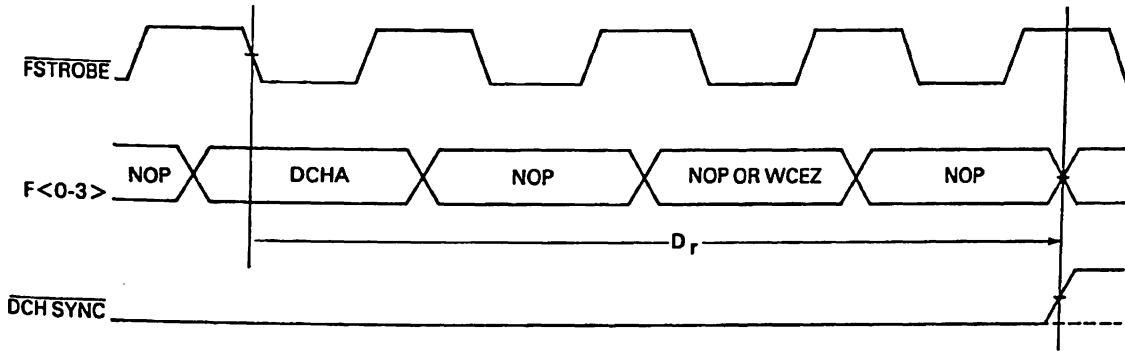
The following circuit is an example of how external BUSY/DONE flags may be connected to an IOC. In this case, the External BUSY/DONE Enable flag must be set to 1 (see Internal Structure and System Requirements, Initialization).



**NOTE:** This circuit is similar in operation to the internal BUSY/DONE logic. However, the designer may alter this circuit to fit specific applications.

### DATA CHANNEL REQUESTS

An IOC requests data channel service if a peripheral device asserts its  $\overline{\text{DCH SYNC}}$  line. On the first Request Enable or Data Channel Address Request following the assertion of  $\overline{\text{DCH SYNC}}$ , the  $\overline{\text{DCHR}}$  line is asserted by the IOC, requesting data channel service from the CPU. At this time a DCHA function code is issued to the peripheral. This function code should be used to release the  $\overline{\text{DCH SYNC}}$  line unless another data channel transfer is desired. The interval  $D_r$  in the following diagram indicates the time in which  $\overline{\text{DCH SYNC}}$  must be removed to prevent prompting extraneous data channel requests:



**NOTES:**

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED FROM A 1.5 VOLT LEVEL.
- SEE TIMING TABLE ON PAGE 43.

DC-0112

### DATA CHANNEL REGISTERS

Two registers are used during data channel transfers, a 15 bit address register and a 16 bit word count register. The address register contains the address of the memory location to be accessed during a particular data channel transfer. The word count register contains the 2's complement of the number of words to be transferred in a block transfer. Under normal operation, both registers are incremented each time a transfer takes place. This assumes that the data is being transferred in or out of a contiguous block of memory. When the word count register overflows, a block transfer is complete.

If internal registers are selected (see System Requirements, Initialization), the internal Address and Word Count Registers are used during data channel transactions. In this case, a DOB instruction loads the internal Address Register as well as placing the data on the Peripheral Port data lines. A DOC instruction loads the internal Word Count Register as well as placing the data on the data lines. A DIB instruction transmits the contents of the internal Address Register to the CPU instead of the data received on the data lines.

If external registers are used, they may be loaded and read using programmed I/O instructions as selected by the designer. However, a DCHA function code indicates that the IOC expects the contents of the Address Register to be gated on data lines  $\text{D}\langle 1-15 \rangle$  and the direction bit on  $\text{D}0$ . This code should also be used to increment both registers, readying them for a subsequent transfer.

### INTERRUPT AND DATA CHANNEL SERVICE

If a peripheral requires service, either a Request Enable or a Data Channel Address Request allows the assertion or clearing of an IOC's request lines. The priority networks determine which IOC receiving these commands is allowed to respond. How Request Enable and Data Channel Address Request affect an IOC's interrupt logic, data channel logic, I/O data port, and peripheral port is illustrated in the flow chart shown in the Internal Structure section.

# mN603

## SYSTEM REQUIREMENTS

### SYSTEM REQUIREMENTS

#### POWER UP

On power-up,  $V_{BB}$  must be brought within its specified operating range before the clocks  $\phi 1$  and  $\phi 2$  are applied to the chip. After all power supply voltages have reached their operating range, the chip is considered to be in a cleared or Reset state. The four phase internal clock generated by  $\phi 1$  and  $\phi 2$  is halted while an IOC is in this state.

Once initialized, an IOC may be placed in the reset state if its I/O CLOCK line is held low for more than 8 cycles of MASTER CLOCK during reception (I/O INPUT high). However, I/O CLOCK should never be held low while the IOC is transmitting to the CPU (I/O INPUT low).

#### INITIALIZATION

The initialization process performs two functions. First, the four phase internal clock is started synchronizing the IOC and the CPU. Second, the Interrupt Disable flag, Data Channel Request flag, Busy flag, Done flag, internal Word Count register, and Address register are cleared (set to 0) while the initialization register is loaded with information received on the Peripheral Port data lines.

The first two Request Enables received from the CPU while an IOC is in the reset state cause it to be initialized. When an IOC is connected to a microNOVA system, this initialization process occurs automatically. This happens because the CPU issues Request Enables even while it is in the Halt state.

If the IOC is already initialized and communicating with the CPU, its internal registers and flags may be reset and the Initialization Register reloaded if the CPU issues an I/O Reset Instruction.

In either case, the following information is loaded into the Initialization Register via data lines  $D\langle 7-15 \rangle$ :

**NOTE:** The contents of the polarity bit does not take effect until after initialization is complete.

DATA PIN(S)	LOADS	SIGNAL LEVEL	EFFECT
D 7	EXTERNAL BUSY/DONE ENABLE BIT	LOW	External BUSY/DONE flags are enabled.
		HIGH	Internal BUSY/DONE flags are enabled.
D 8	EXTERNAL REGISTER ENABLE BIT	LOW	External Address and Word Count Registers are enabled.
		HIGH	Internal Address and Word Count Registers are enabled.
D 9	POLARITY BIT	LOW	Data on $D\langle 0-15 \rangle$ interpreted using positive logic. (i.e. high = 1)
		HIGH	Data on $D\langle 0-15 \rangle$ interpreted using negative logic. (i.e. high = 0)
$D\langle 10-15 \rangle$	DEVICE CODE	LOW/HIGH	When loaded, the six bit device code on these lines is interpreted using negative logic (i.e. high = 0).

DG-0477

#### CLOCKS

The two phase, non-overlapping clock  $\phi 1$  and  $\phi 2$  generates the IOC's four phase internal clock.  $\phi 1$  and  $\phi 2$  are generated from MASTER CLOCK as received from the I/O bus by the IOC I/O Transceiver chip (mN636). A Clock Driver chip (mN640) drives the TTL clock outputs of the Transceiver to the MOS levels required by the IOC. All the specifications and timing in this section assume a MASTER CLOCK frequency of 8.333MHz.

# mN603

## ELECTRICAL SPECIFICATIONS

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{BB}$ .....	-7V
Supply voltage, $V_{CC}$ .....	7V
Supply voltage, $V_{DD}$ .....	13V
Supply voltage, $V_{GG}$ .....	17V
Input voltage, $V_I$ .....	7V
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg.C

**NOTE:** All voltages are measured with respect to ground. Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{BB}$ .....	-4.25V $\pm$ 0.5V
Supply voltage, $V_{CC}$ .....	5V $\pm$ 0.25V
Supply voltage, $V_{DD}$ .....	10V $\pm$ 1V
Supply voltage, $V_{GG}$ .....	14V $\pm$ 1V
Operating free air temperature range, $T_A$ .....	0 deg. to 70 deg. C

**NOTE:** All voltages are measured with respect to ground. On power-up,  $V_{BB}$  must be within its specified operating range before any other power supply voltages are applied to the circuit.

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{IH}$	$\emptyset 1, \emptyset 2$		13	15	V
	D <0-15> BUSY, DONE INTP, DCHP INT SYNC, DCH SYNC		3.75	6	V
	I/O CLOCK I/O DATA1, I/O DATA2		2.7	6	V
$V_{IL}$	$\emptyset 1, \emptyset 2$		-2	+0.8	V
	D <0-15> BUSY, DONE INTP, DCHP INT SYNC, DCH SYNC		-1	+0.8	V
	I/O CLOCK I/O DATA1, I/O DATA2		-0.5	+0.5	V



# mN603

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (CONT'D)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
I <sub>IH</sub>	Ø1, Ø2	V <sub>I</sub> = 15V		+01	mA
	D<0-15> BUSY, DONE INTP, DCHP INT SYNC, DCH SYNC	V <sub>I</sub> = 4V		+70	µA
	I/O CLOCK I/O DATA1, I/O DATA2	V <sub>I</sub> = 2.7V		+70	µA
I <sub>IL</sub>	Ø1, Ø2	V <sub>I</sub> = 0.8V		10	µA
	I/O CLOCK I/O DATA1, I/O DATA2	V <sub>I</sub> = 0V		-4	mA
	BUSY, DONE INTP, DCHP INT SYNC, DCH SYNC	V <sub>I</sub> = 0V		-2	mA
	D<0-15>	V <sub>I</sub> = 0V		-2	mA
V <sub>OH</sub>	D<0-15> BUSY, DONE	I <sub>O</sub> = -70µA	4	V <sub>CC</sub>	V
	I/O CLOCK I/O DATA1, I/O DATA2	I <sub>O</sub> = -40µA	3	V <sub>CC</sub>	V
	F<0-3>, FSTROBE DCHR, INTR I/O INPUT	I <sub>O</sub> = -0.1mA	2.7	V <sub>CC</sub>	V
V <sub>OL</sub>	D<0-15> I/O CLOCK I/O DATA1, I/O DATA2 BUSY, DONE	I <sub>O</sub> = 2mA	0	0.5	V
	I/O INPUT F<0-3> FSTROBE DCHR, INTR	I <sub>O</sub> = 4mA	0	0.5	V
I <sub>BB</sub>	MAX AVERAGE SUPPLY CURRENT	V <sub>BB</sub> = -4.25 ± 0.5V		-0.5	mA
I <sub>CC</sub>	MAX AVERAGE SUPPLY CURRENT	V <sub>CC</sub> = 5.0 ± 0.25V		25	mA
I <sub>DD</sub>	MAX AVERAGE SUPPLY CURRENT	V <sub>DD</sub> = 10.0 ± 1.0V		25	mA
I <sub>GG</sub>	MAX AVERAGE SUPPLY CURRENT	V <sub>GG</sub> = 14.0 ± 1.0V		20	mA
C <sub>I</sub>	Ø1, Ø2 BUSY, DONE			50	pF
	D<0-15> I/O CLOCK I/O DATA1, I/O DATA2 INTP, DCHP INT SYNC, DCH SYNC			30	pF

NOTE: Positive current is into the pin.

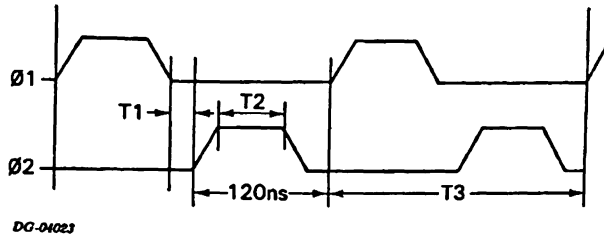
# mN603

## ELECTRICAL SPECIFICATIONS

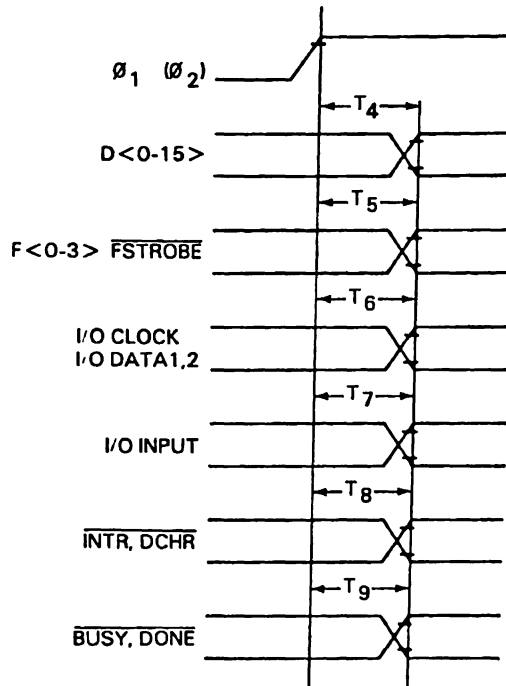
### AAC CHARACTERISTICS

#### Switching Diagrams

### IOC CLOCK TIMING



### DATA PINS - OUTPUT MODE



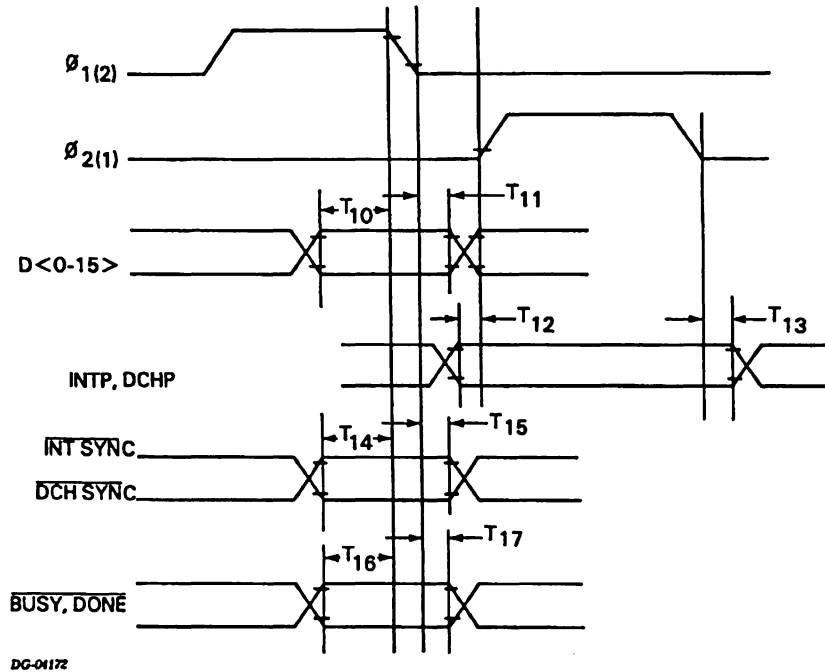
NOTE: TIME INTERVALS ARE MEASURED BETWEEN 10% AND/OR 90% POINTS, UNLESS OTHERWISE SPECIFIED.

DG-04173

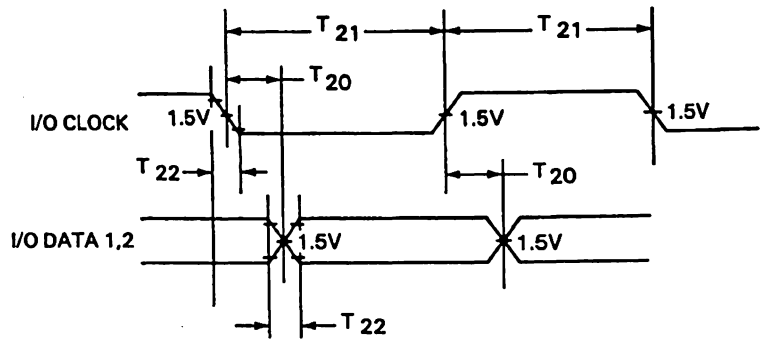
# mN603

## ELECTRICAL SPECIFICATIONS

### DATA PINS - INPUT MODE

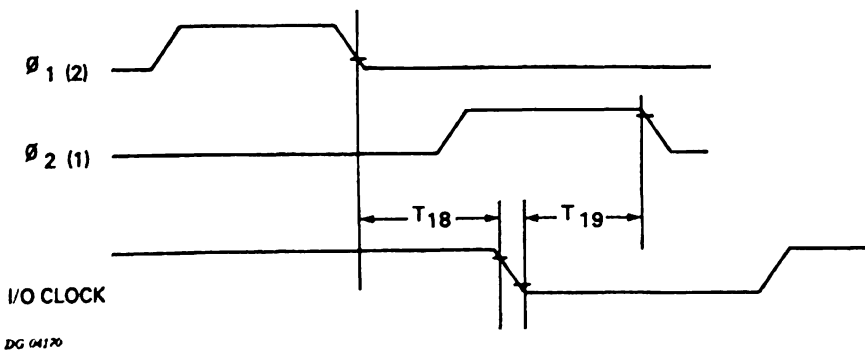


### I/O PORT - INPUT MODE



NOTE: TIME INTERVALS ARE MEASURED BETWEEN 10% AND/OR 90% POINTS, UNLESS OTHERWISE SPECIFIED.

### I/O CLOCK - OUTPUT MODE



Transition Timing

CLASS	SYMBOL	PIN	MIN	MAX	UNIT
CLOCKS	T <sub>1</sub>	Ø1, Ø2 SEPARATION	5	-	ns
	T <sub>2</sub>	WIDTH	75	-	ns
	T <sub>3</sub>	CYCLE	235	245	ns
DATA OUTPUTS	T <sub>4</sub>	D < 0-15 >	-	60	ns
	T <sub>5</sub>	F < 0-3 >, $\overline{\text{FSTROBE}}$	-	60	ns
	T <sub>6</sub>	I/O CLOCK, I/O DATA1 I/O DATA2	-	30	ns
	T <sub>7</sub>	I/O INPUT	-	30	ns
	T <sub>8</sub>	$\overline{\text{INTR}}$ , $\overline{\text{DCHR}}$	-	60	ns
	T <sub>9</sub>	$\overline{\text{BUSY}}$ , $\overline{\text{DONE}}$	-	60	ns
	DATA INPUTS	T <sub>10</sub>	D < 0-15 > SETUP	50	-
T <sub>11</sub>		D < 0-15 > HOLD	0	-	ns
T <sub>12</sub>		$\overline{\text{INTP}}$ , $\overline{\text{DCHP}}$ SETUP	0	-	ns
T <sub>13</sub>		$\overline{\text{INTP}}$ , $\overline{\text{DCHP}}$ HOLD	0	-	ns
T <sub>14</sub>		$\overline{\text{INT SYNC}}$ , $\overline{\text{DCH SYNC}}$ SETUP	50	-	ns
T <sub>15</sub>		$\overline{\text{INT SYNC}}$ , $\overline{\text{DCH SYNC}}$ HOLD	0	-	ns
T <sub>16</sub>		$\overline{\text{BUSY}}$ , $\overline{\text{DONE}}$ SETUP	50	-	ns
T <sub>17</sub>		$\overline{\text{BUSY}}$ , $\overline{\text{DONE}}$ HOLD	0	-	ns
I/O DATA PORT	T <sub>18</sub>	I/O CLOCK HOLD	0	-	ns
	T <sub>19</sub>	I/O CLOCK SETUP	70	-	ns
	T <sub>20</sub>	I/O SKEW	-10	+10	ns
	T <sub>21</sub>	I/O PULSE WIDTH	115	125	ns
	T <sub>22</sub>	RISE, FALL TIMES	-	10	ns

NOTE: All the above times assume a MASTER CLOCK frequency of 8.333 Mhz.

DG-0412

I/O DATA PORT TIMING TABLE

CLASS	MNEMONIC	MIN.	MAX.	UNITS
All	T <sub>1</sub>	110	130	ns
	T <sub>2</sub>	110	130	ns
	T <sub>SKEW</sub>	-5	+15	ns
	T <sub>N</sub> (next command)	840	-	ns
I/O Instructions	DT <sub>O</sub> (Data Out Transfer)	470	850	ns
	DT <sub>I</sub> (Data In Transfer)	1190	1330	ns
Request Enable and Data Channel Address Request	a	360	540	ns
Data Channel Transactions	C (Data Channel Address Request to first transfer)	710	850	ns
	DC <sub>O</sub> (Data Channel Out)	590	1930	ns
	DC <sub>I</sub> (Data Channel In)	1190	1210	ns

NOTE: All the above times assume a MASTER CLOCK frequency of 8.333Mhz  
DG-0413

PERIPHERAL PORT TIMING TABLE

OPERATION	MNEMONIC	DESCRIPTION	MIN.	MAX.	UNIT
Common to all operations	F <sub>S</sub>	Function code setup time prior to $\overline{\text{FSTROBE}}$	60	-	ns
	F <sub>T</sub>	$\overline{\text{FSTROBE}}$ duration	180	300	ns
	F <sub>H</sub>	Function pin hold time	60	-	ns
I/O data out Instruction and Data Channel Out Transfer		Data output timing same as function code timing above			
I/O data in Instruction and Data Channel In Transfer	D <sub>S</sub>	Data setup time prior to $\overline{\text{FSTROBE}}$	120	240	ns
	D <sub>H</sub>	Data hold time after $\overline{\text{FSTROBE}}$	0	120	ns
	T <sub>AI</sub>	DCHA to DHCI	336	-	µs
	T <sub>AO</sub>	DCHA to DCHO	656	-	µs

DG-0415

SYNCHRONIZATION TIMING TABLE

OPERATION	MNEMONIC	DESCRIPTION	MIN.	MAX.	UNIT
Request Control	D <sub>r</sub>	Time in which $\overline{\text{INT SYNC}}$ or $\overline{\text{DCH SYNC}}$ must be removed to prevent further requests		840	ns
BUSY and DONE	C <sub>D</sub>	Delay between $\overline{\text{FSTROBE}}$ and $\overline{\text{BUSY}}$ or $\overline{\text{DONE}}$ transition	-10	+10	ns
	C <sub>S</sub>	$\overline{\text{BUSY}}$ or $\overline{\text{DONE}}$ assertion setup time for internal recognition of desired functions	120	-	ns
	C <sub>H</sub>	$\overline{\text{BUSY}}$ or $\overline{\text{DONE}}$ assertion hold time for recognition (Note: if C <sub>H</sub> 240 ns then signal will float toward Vcc until next $\overline{\text{FSTROBE}}$ )	0	-	ns

NOTE: All the above times assume a MASTER CLOCK frequency of 8.333 Mhz

DG-0414





This page intentionally left blank.

**mN606**  
**4K DYNAMIC RANDOM ACCESS MEMORY**  
**CONTENTS**



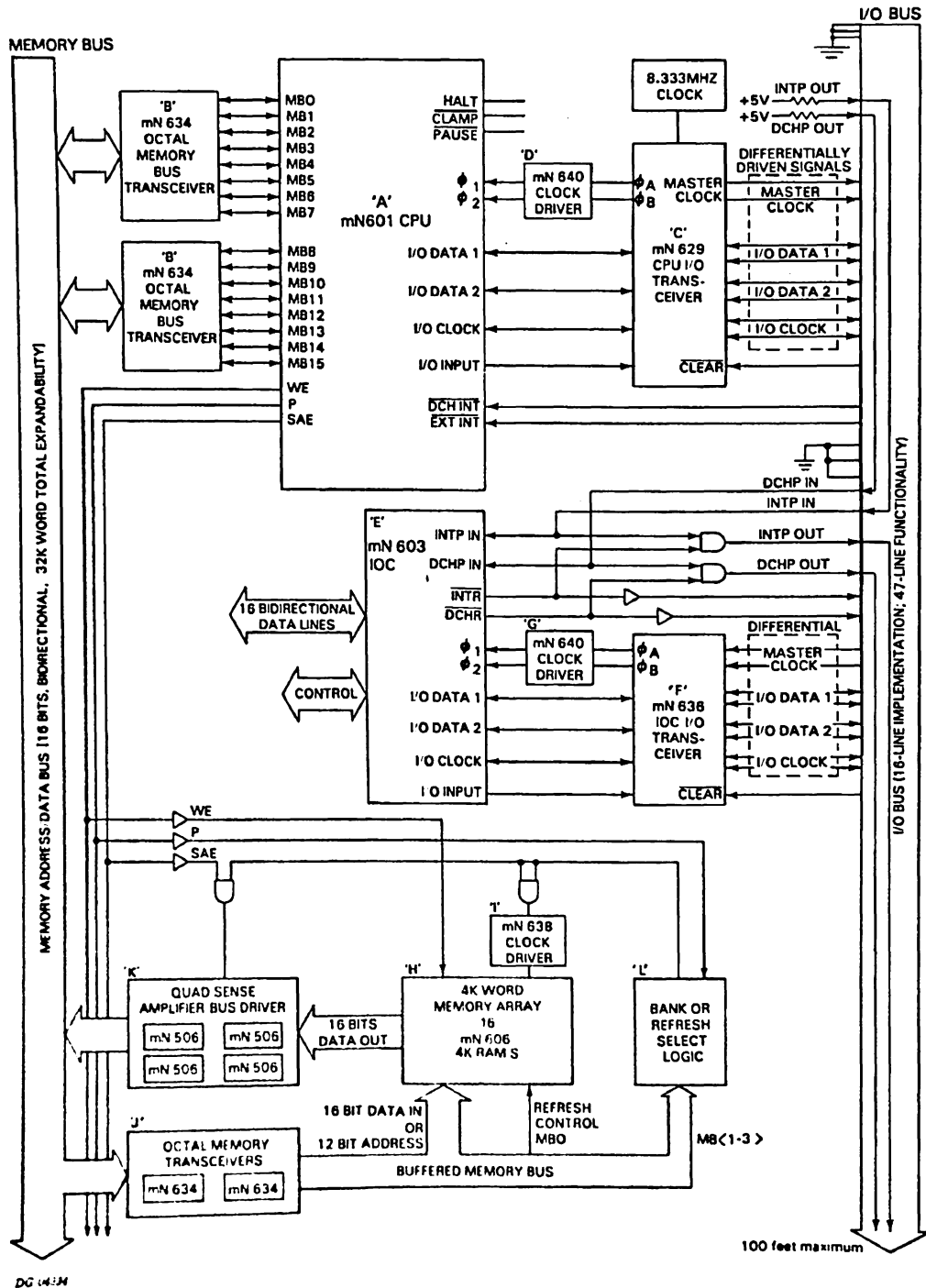
<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>3</b>	<b>INTERNAL STRUCTURE</b>
<b>4</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>4</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>4</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>4</b>	<b>DC CHARACTERISTICS</b>
<b>4</b>	<b>AC CHARACTERISTICS</b>
<b>5</b>	<b>Read Operation</b>
<b>5</b>	<b>Write Operation</b>
<b>6</b>	<b>Refresh Operation</b>
<b>7</b>	<b>Transition Timing</b>
<b>8</b>	<b>PACKAGE SPECIFICATIONS</b>

74



# mN606

## 4K DYNAMIC RANDOM ACCESS MEMORY



DC (4:34)



DataGeneral Corporation, Westboro, Massachusetts 01581

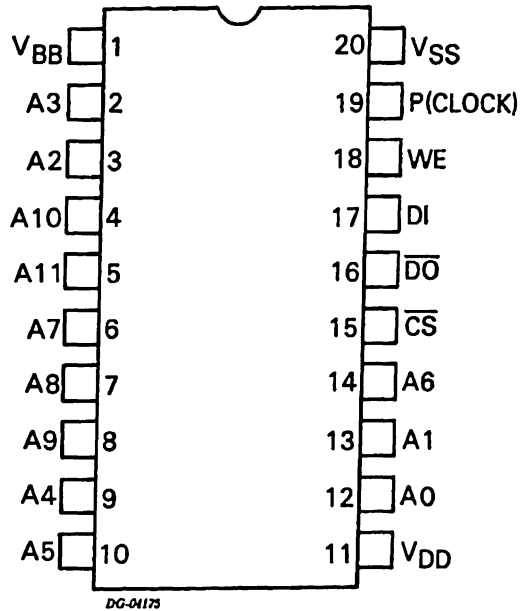
# mN606

## 4K X 1 DYNAMIC RANDOM ACCESS MEMORY

### FEATURES

- NMOS DYNAMIC RANDOM ACCESS MEMORY IN A 20 PIN PLASTIC PACKAGE
- ORGANIZED AS 4096 BY 1 BIT
- FAST 180ns ACCESS TIME
- FAST INTERNAL LATCHES FOR ADDRESS AND CHIP SELECT
- REFRESHES 64 BITS AT A TIME
- OPEN DRAIN OUTPUT

### PACKAGE



### GENERAL DESCRIPTION

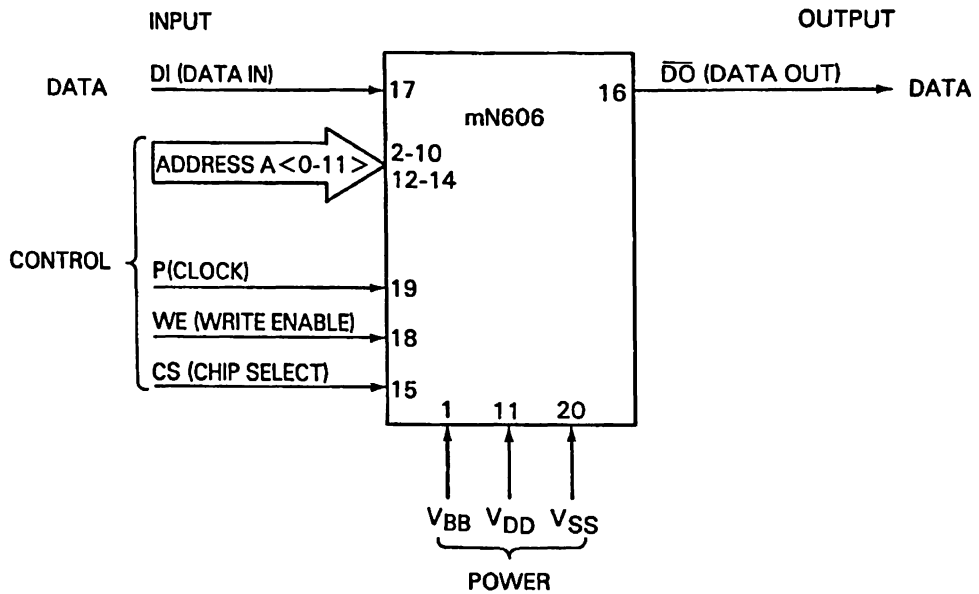
The mN606 is a 4096 by 1 bit NMOS memory using cost-efficient dynamic RAM technology to implement the microNOVA family's large memory orientation. Its 20-pin package permits separate pins for each of the 12 address lines as well as for the data in and data out lines. The chip's high speed access time of 180 nanoseconds contributes significantly to the microNOVA family's high performance.

To construct a 16 bit by 4K word memory board, 16 mN606 RAMs are needed as well as two mN634 Memory Transceivers, four mN506 Sense Amp/Bus Drivers, one mN638 Clock Driver, and supporting components. The system block diagram illustrates the basic interconnection of these components.

# mN606

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/ OUT	FUNCTION
A<0-11> (Address)	2-10 12-14	In	These pins enter an address which selects one of 4096 bits. A<0-5> is used during refresh operations for column selection (1/64th of the RAM). Address is latched on rising edge of P clock.
DI (Data In)	17	In	One bit of data written into location specified on address lines during memory write operation. High level logical one input.
$\overline{DO}$ (Data Out)	16	Out	One bit of data read from location specified by address pins during memory read operation. High impedance, inverted output (i.e., low = 1, high = 0).
P (Clock)	19	In	All memory operations initiated by latching Address Register and starting internal clock on rising edge of P clock.
WE (Write Enable)	18	In	When asserted high, enables data from DI to be written into specified memory location. When low, enables data at the specified address to be read out to the DO pin.
$\overline{CS}$ (Chip Select)	15	In	If asserted low when P goes high, a memory read or write operation is initiated. If high when P goes high, a memory refresh operation is initiated.
V <sub>BB</sub>	1		-4.25 ± 0.5V
V <sub>DD</sub>	11		+14.0 ± 1.0V
V <sub>SS</sub>	20		Ground

# mN606

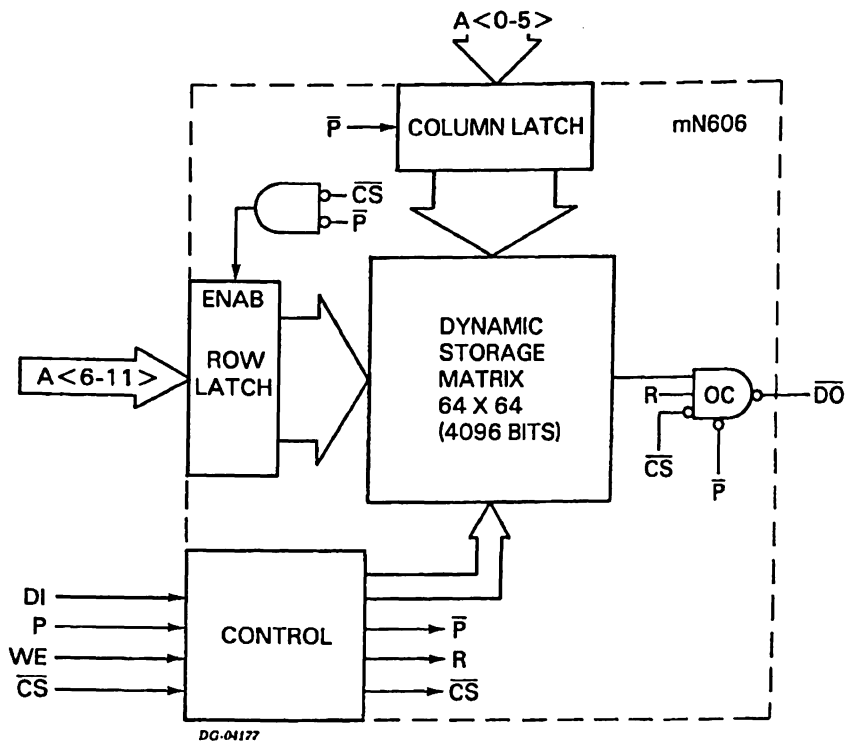
## INTERNAL STRUCTURE

### INTERNAL STRUCTURE

The operation performed by the RAM depends on the level of CS when P clock goes high. If CS is low, a memory Read or Write operation is performed. The address on pins A <0-11> is latched on the rising edge of P clock. If WE remains low, the data appears on DO. A write operation is initiated by pulling WE high. The rising edge of WE forces DO high while the falling edge loads the data from DI. Both operations may be combined to form a Read/Delay/Write (ie Read/Modify/ Write) operation.

Since the memory mechanism is dynamic, the chip must periodically be refreshed. A Refresh operation is initiated during a memory Write operation if CS remains high. In this case, A <0-5> are latched on the rising edge of P clock to select a particular 64 bit column of memory to be refreshed. Since there are 64 columns, 64 refresh operations are necessary to refresh all 4096 memory cells. Each location must be refreshed at least once every 2.0 mS. The Data Out (DO) line remains high throughout a refresh operation.

### INTERNAL BLOCK DIAGRAM



# mN606

## ELECTRICAL SPECIFICATIONS

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{BB}$ .....	-7V
Supply voltage, $V_{DD}$ .....	17V
Clock input voltage, $V_{iCLK}$ .....	17V
All other inputs, $V_i$ .....	-7V
Operating temperature range, (air moving at 200 feet/minute), $T_A$ .....	0 deg. to 70 deg. C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 85 deg. C

NOTE: Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{BB}$ .....	-4.25 $\pm$ 0.5V
Supply voltage, $V_{DD}$ .....	14 $\pm$ 1V
Operating temperature range, (air moving at 200 feet/minute) .....	0 deg. to 70 deg. C

NOTE: On power-up,  $V_{BB}$  must be within its specified operating range before  $V_{DD}$  is applied to the circuit.

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{IH}$	P(CLOCK)		+13.0	+15.5	V
	A <0-11> $\overline{CS}$ , WE, DI		+4.0	+6.0	V
$V_{IL}$	P(CLOCK), A <0-11> $\overline{CS}$ , WE, DI		-1.0	+0.8	V
$I_{IH}$	P(CLOCK)	$V_i = 15.5V$	---	+0.8	mA
	$\overline{CS}$ , WE, DI A <0-11>	$V_i = 6.0V$	---	+0.01	mA
$I_{IL}$	P(CLOCK) $\overline{CS}$ , WE, DI	$V_i = +0.8V$	---	+0.01	mA
$I_{OH}$	$\overline{D\overline{O}}$	SEE TEST CIRCUIT, UNDER AC CHARACTERISTICS	0	0.04	mA
$I_{OL}$	$\overline{D\overline{O}}$	SEE TEST CIRCUIT, UNDER AC CHARACTERISTICS	1.5	9.0	mA
$I_{BB}$	SUPPLY CURRENT	$V_{BB} = -4.25 \pm 0.5V$	---	-0.15	mA
$I_{DD}$	SUPPLY CURRENT	$V_{DD} = +14.0 \pm 1.0V$	---	+20.0	mA
$C_i$	P(CLOCK)			23	pF
	A <0-11> $\overline{CS}$ , WE, DI			7	pF
$R_T$	MEMORY CELL REFRESH TIME		---	2.0	mS

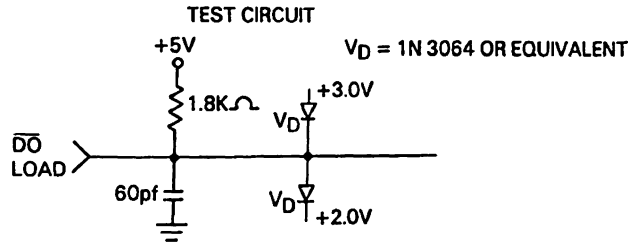
NOTE: Positive current is into the pin.

# mN606

## ELECTRICAL SPECIFICATIONS

### AC CHARACTERISTICS

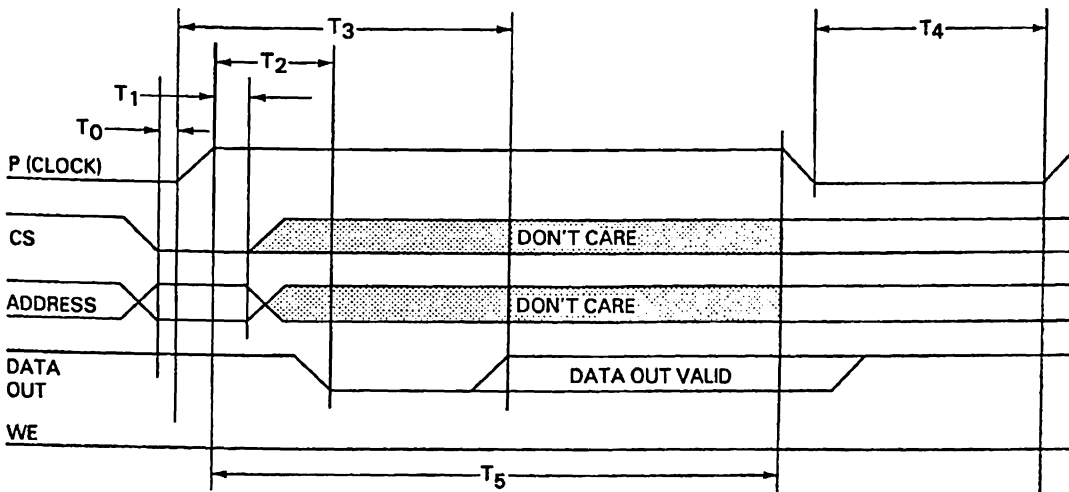
**TEST CONDITIONS**  
 PICLOCK) RISE AND FALL 20NS  
 ALL OTHERS 10NS  
 ALL RISE AND FALL TIMES ARE MEASURED  
 FROM THE 10 TO 90 PERCENT POINTS.  
 LOAD CAPACITOR INCLUDES  
 JIG AND WIRING CAPACITANCE.



DC-04178



### Read Operation



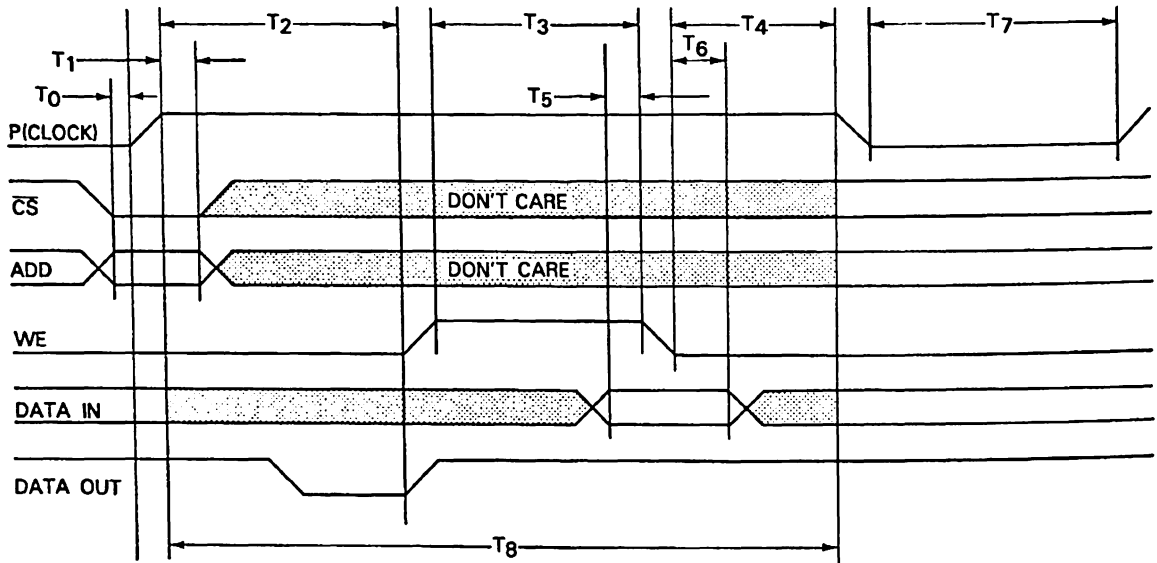
WE MUST REMAIN LOW DURING THE ENTIRE READ CYCLE.  
 NOTE: SEE AC CHARACTERISTICS TIMING TABLE ON PAGE 7.

DC-04331

# mN606

## ELECTRICAL SPECIFICATIONS

### Write Operation

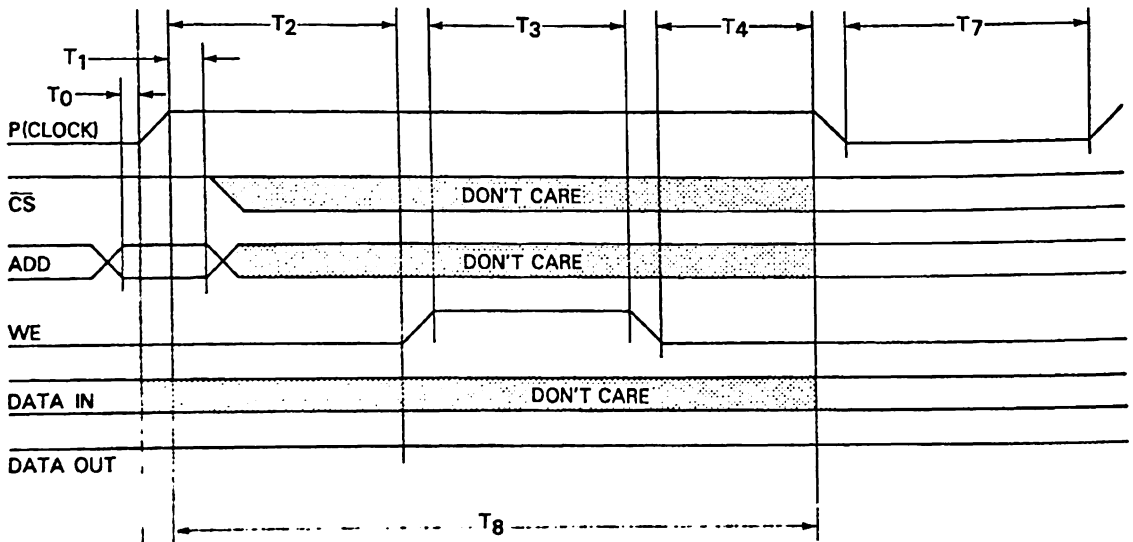


WHEN WE GOES HIGH, A WRITE CYCLE IS INITIATED AND DATA OUT IS DISABLED.

NOTE: SEE AC CHARACTERISTICS TIMING TABLE ON PAGE 7.

DG-01179

### Refresh Operation



WHEN REFRESHING, CS MUST BE HELD HIGH FOR THE TIME  $T_1$ . THE DATA INPUTS ARE IGNORED.

NOTE: SEE AC CHARACTERISTICS TIMING TABLE ON PAGE 7.

DG 01180

# mN606

## ELECTRICAL SPECIFICATIONS

### Transition Timing

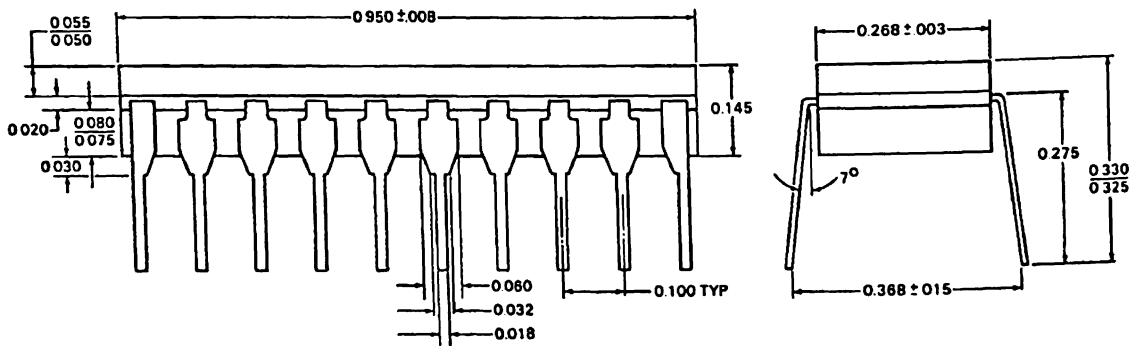
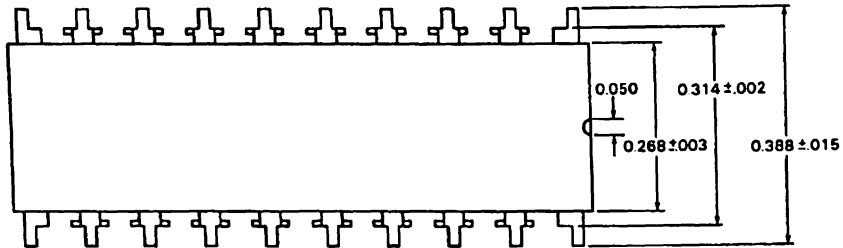
OPERATION	SIGNAL	DESCRIPTION	MIN.	MAX.	UNITS
READ	T <sub>0</sub>	SETUP TIME ADDRESS AND $\overline{CS}$	0		ns
	T <sub>1</sub>	HOLD TIME ADDRESS AND $\overline{CS}$	75		ns
	T <sub>2</sub>	DATA LOW TIME		65	ns
	T <sub>3</sub>	ACCESS TIME		180	ns
	T <sub>4</sub>	P (clock) OFF TIME	160		ns
	T <sub>5</sub>	P (clock) HIGH	220	3400	ns
WRITE AND REFRESH	T <sub>0</sub>	SETUP TIME ADDRESS AND $\overline{CS}$	0		ns
	T <sub>1</sub>	HOLD TIME ADDRESS AND $\overline{CS}$	75		ns
	T <sub>2</sub>	DELAY TIME P TO WE	160	2900	ns
	T <sub>3</sub>	WE PULSE WIDTH	180	250	ns
	T <sub>4</sub>	WRITE TIME	130	270	ns
	T <sub>5</sub>	DATA SETUP TIME	50		ns
	T <sub>6</sub>	DATA HOLD TIME	85		ns
	T <sub>7</sub>	P (clock) OFF TIME	160		ns
	T <sub>8</sub>	P (clock) WIDTH	490	3400	ns



# mN606

## PACKAGE SPECIFICATIONS

### PACKAGE SPECIFICATIONS



DG-0111

74

This page intentionally left blank.

**mN506**  
**QUAD SENSE AMP/BUS DRIVER**  
**CONTENTS**

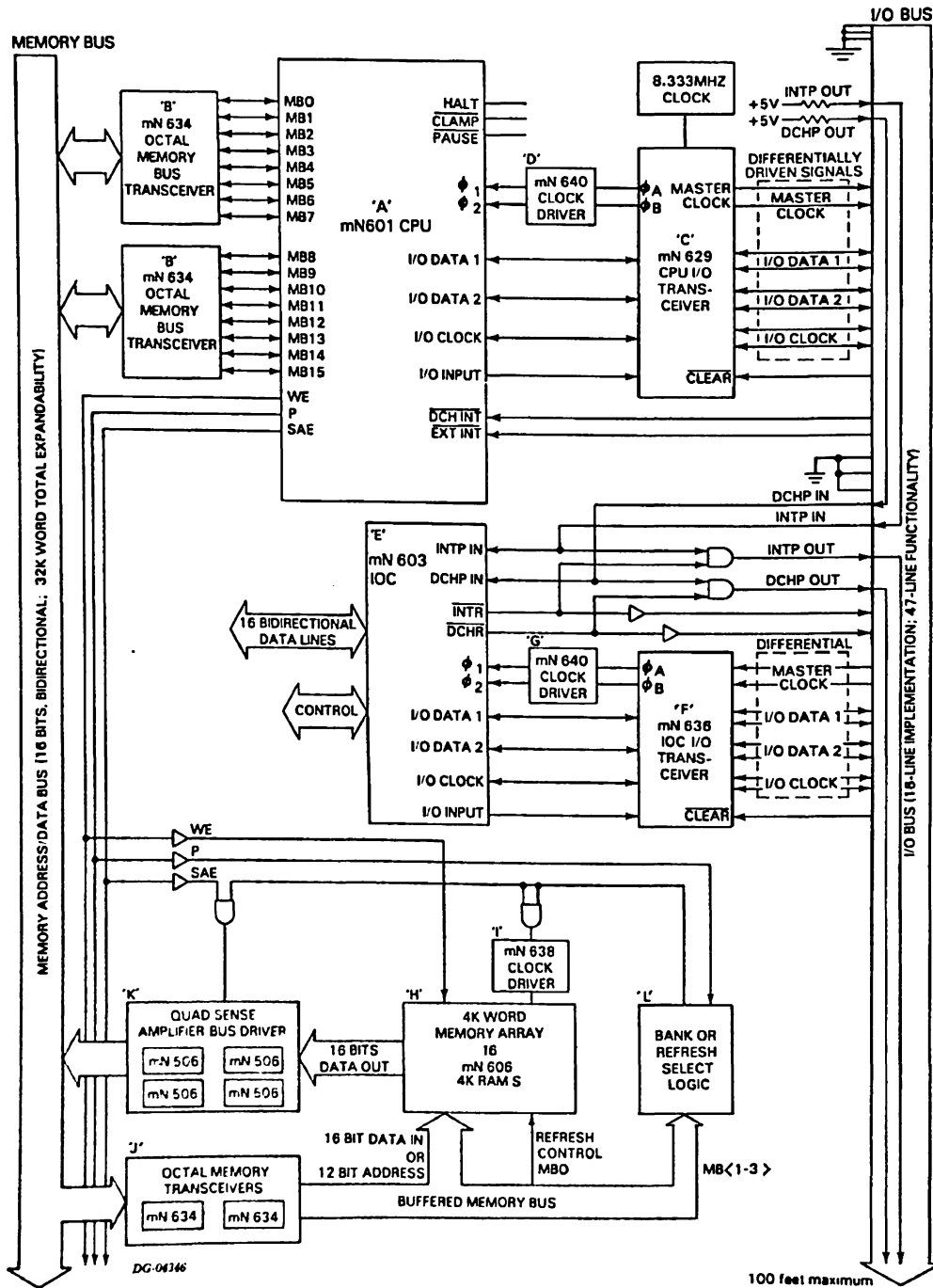
5

# **mN506**

## **CONTENTS**

<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>3</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>3</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>3</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>3</b>	<b>DC CHARACTERISTICS</b>
<b>5</b>	<b>AC CHARACTERISTICS</b>
<b>5</b>	<b>Switching Diagrams</b>
<b>5</b>	<b>Switching Timing</b>
<b>6</b>	<b>PACKAGE SPECIFICATIONS</b>

# mN506 QUAD SENSE AMP/BUS DRIVER



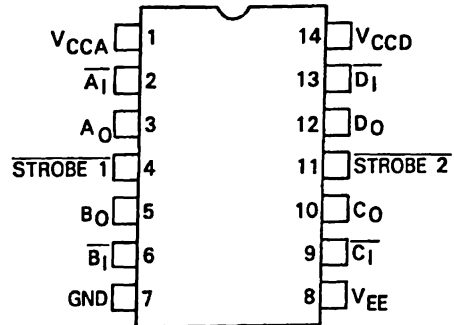
# mN506

## QUAD SENSE AMP/BUS DRIVER

### FEATURES

- FOUR SENSE AMPLIFIERS/BUS DRIVERS IN A 14-PIN CERDIP PACKAGE.
- COMPATIBLE WITH mN606 RANDOM ACCESS MEMORIES.
- TWO INDEPENDENT STROBE LINES.
- INVERTED OPEN COLLECTOR OUTPUTS WITH 35mA CURRENT SINKING CAPABILITIES.
- DRIVERS MAY BE WIRE ORed TO SIMILAR OUTPUT CIRCUITS.

### PACKAGE



DG-01132

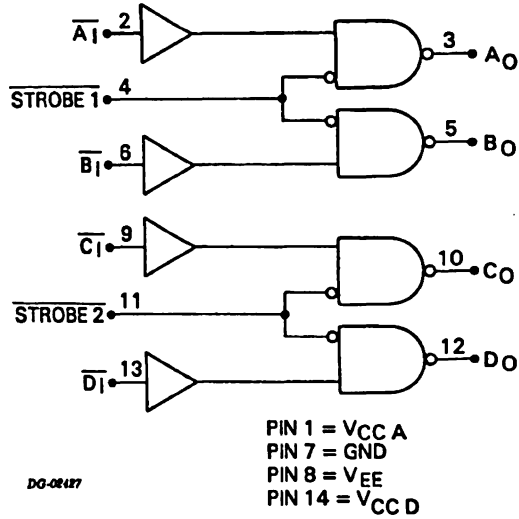
### GENERAL DESCRIPTION

- † The mN506 Quad Sense Amplifier/Bus Driver is compatible with the mN606 RAM chip. The output drivers may be wire ORed allowing direct connection of multiple memory arrays to the microNOVA memory bus.
- † The mN506 contains four analog sense amplifiers and four digital bus drivers. Independent analog and digital supply voltage sources allow power conservation in systems with battery back-up. The analog voltage supply (for the sense amplifiers) may be shut off while the digital voltage supply provides power to maintain the memory bus integrity.
- † The analog inputs of the mN506 are designed to sense the current drain caused by the outputs of the mN606 Random Access Memory. The bus drivers provide open-collector TTL compatible outputs capable of sinking up to 35mA. This allows the wire ORing of drivers on the Memory Bus. The bus drivers are enabled in pairs with the assertion of the STROBE1 and STROBE2 signals. In addition, their outputs are inverted. Since the mN606 RAM outputs are also inverted, a low level output from the mN506 indicates that the accessed memory location contains a logical 0.

# mN506

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



DC-08127

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
$\overline{A}_1$	2	IN	Sense amplifier inputs.
$\overline{B}_1$	6	IN	
$\overline{C}_1$	9	IN	
$\overline{D}_1$	13	IN	
$A_O$	3	OUT	Inverted open collector outputs. $A_O$ and $B_O$ enabled when $\overline{STROBE1}$ asserted low. $C_O$ and $D_O$ enabled when $\overline{STROBE2}$ asserted low.
$B_O$	5	OUT	
$C_O$	10	OUT	
$D_O$	12	OUT	
$\overline{STROBE1}$	4	IN	When asserted low, enables $A_O$ and $B_O$
$\overline{STROBE2}$	11	IN	When asserted low, enables $C_O$ and $D_O$
$V_{CCA}$	1		+5 ± 0.25 volts, Analog circuit power supply.
$V_{CCD}$	14		+5 ± 0.25 volts, Digital circuit power supply.
$V_{EE}$	8		-5 ± 0.25 volts, Analog circuit power supply.
GND	7		GROUND

DC-04382

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{CCA}$	.....	.7V
Supply voltage, $V_{CCD}$	.....	.7V
Supply voltage, $V_{EE}$	.....	-7V
Input voltage, $V_I$	.....	+5.5V
Output current, $I_O$	.....	60mA
Operating free-air temperature range, $T_A$	.....	0 deg. to 70 deg. C
Storage temperature range, $T_{STG}$	.....	-25 deg. to 155 deg. C

**NOTE:** Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{CCA}$	.....	$5.0 \pm 0.25V$
Supply voltage, $V_{CCD}$	.....	$5.0 \pm 0.25V$
Supply voltage, $V_{EE}$	.....	$-5.0 \pm 0.25V$
Average power dissipation	.....	0.8 W
Operating free-air temperature range, $T_A$	.....	0 deg to 70 deg. C

1 DG-0434

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{IH}$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$		2.0		V
$V_{IL}$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$			0.8	V
$V_{IHA}^*$	$\overline{A_1}, \overline{B_1}, \overline{C_1}, \overline{D_1}$	$V_{CCA} = V_{CCD} = 4.75V$ $V_{EE} = -5.25V$ $I_{IAMP} = -9.0mA$	1.0		V
$V_{ILA}^*$	$\overline{A_1}, \overline{B_1}, \overline{C_1}, \overline{D_1}$	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -4.75V$ $I_{IAMP} = 0mA$		3.0	V
$V_I$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$ (INPUT CLAMP VOLTAGE)	$V_{CCA} = V_{CCD} = 4.75V$ $V_{EE} = -4.75V$ $I_I = -18mA$	-1.5		V
$I_{IH}$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $V_I = 2.7V$		200	$\mu A$

DG-0434

\* NOTE: SEE SENSE AMP V-I CHARACTERISTICS ON PAGE 5



# mN506

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (Cont.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$I_{IL}$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $V_I = 0.5V$	-4		mA
$I_I$	$\overline{\text{STROBE1}}$ $\overline{\text{STROBE2}}$	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $V_I = 5.5V$		1	mA
$V_{OL}$	A <sub>0</sub> ,B <sub>0</sub> ,C <sub>0</sub> ,D <sub>0</sub>	$V_{CCA} = V_{CCD} = 4.75V$ $V_{EE} = -5.25V$ $V_{ISTROBE} = 0.8V$ $I_{IAMP} = -400\mu A$ $I_O = 35mA$		0.5	V
$I_{OH}$	A <sub>0</sub> ,B <sub>0</sub> ,C <sub>0</sub> ,D <sub>0</sub>	$V_{CCA} = V_{CCD} = 4.75V$ $V_{EE} = -4.75V$ $V_O = 5.5V$			
		$I_{IAMP} = -1.8mA$ $V_{ISTROBE} = 0.8V$		50	$\mu A$
		$I_{IAMP} = -400\mu A$ $V_{ISTROBE} = 2.0V$		50	$\mu A$
$I_{CCA}$	ANALOG POWER SUPPLY CURRENT ( $V_{CCA}$ )	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -1.8mA$ $V_{ISTROBE} = 2.0V$		50	mA
		$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -400\mu A$ $V_{ISTROBE} = 0.8V$		50	mA
$I_{CCD}$	DIGITAL POWER SUPPLY CURRENT ( $V_{CCD}$ )	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -1.8mA$ $V_{ISTROBE} = 2.0V$		31	mA
		$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -400\mu A$ $V_{ISTROBE} = 0.8V$		51	mA
$I_{EE}$	ANALOG POWER SUPPLY CURRENT ( $V_{EE}$ )	$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -1.8mA$ $V_{ISTROBE} = 2.0V$		-65	mA
		$V_{CCA} = V_{CCD} = 5.25V$ $V_{EE} = -5.25V$ $I_{IAMP} = -400\mu A$ $V_{ISTROBE} = 0.8V$		-65	mA

NOTE: POSITIVE CURRENT IS INTO THE PIN.

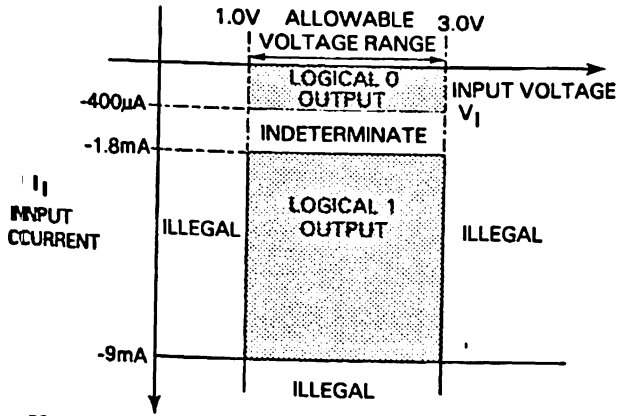
# mN506

## ELECTRICAL SPECIFICATIONS

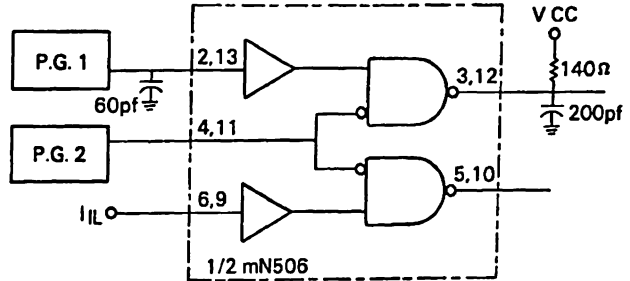
### AC CHARACTERISTICS

#### SENSE-AMP INPUT V-I CHARACTERISTICS

TEST CIRCUIT:



DG-04133

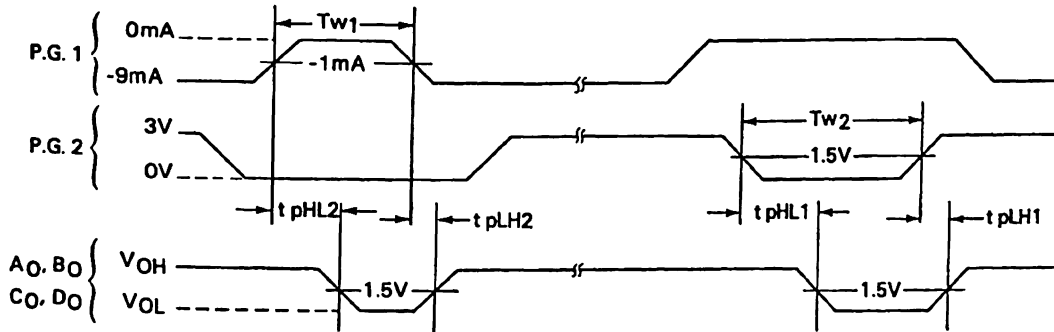


NOTE: PULSE GENERATORS 1 AND 2 NOT INTERNALLY TERMINATED.

$0 < t_r = t_f < 6\text{nS}$   
 PRR = 1MHz  
 $t_{w1} = 100\text{nS}$   
 $t_{w2} = 300\text{nS}$

DG-04135

### Switching Diagram



DG-04134

### Transition Timing

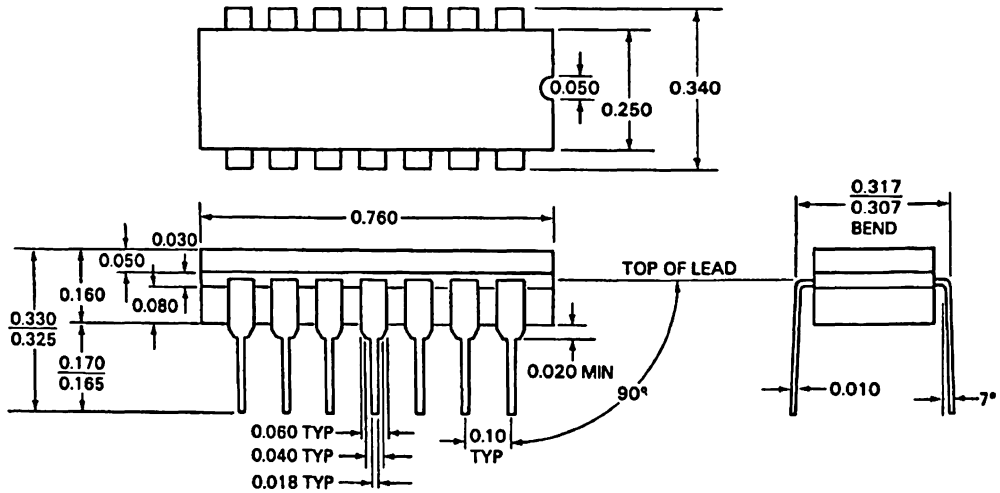
SYMBOL	CHARACTERISTIC	CONDITION*	MIN.	MAX.	UNITS
$t_{pHL1}$	PROPAGATION DELAY (HIGH TO LOW LEVEL)	$V_{CCA} = V_{CCD} = 5.0\text{V}$ $V_{EE} = -5.0\text{V}$ $I_{IL} = -1.8\text{mA}$	7	18	nS
$t_{pLH1}$	PROPAGATION DELAY (LOW TO HIGH LEVEL)	$V_{CCA} = V_{CCD} = 5.0\text{V}$ $V_{EE} = -5.0\text{V}$ $I_{IL} = -1.8\text{mA}$	7	18	nS
$t_{pHL2}$	PROPAGATION DELAY (HIGH TO LOW LEVEL)	$V_{CCA} = V_{CCD} = 5.0\text{V}$ $V_{EE} = -5.0\text{V}$ $I_{IL} = -1.8\text{mA}$	10	25	nS
$t_{pLH2}$	PROPAGATION DELAY (LOW TO HIGH LEVEL)	$V_{CCA} = V_{CCD} = 5.0\text{V}$ $V_{EE} = -5.0\text{V}$ $I_{IL} = -1.8\text{mA}$	25	38	nS

\* See Test Circuit

# mN506

## PACKAGE SPECIFICATIONS

### PACKAGE SPECIFICATIONS



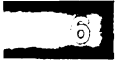
This page intentionally left blank.

5

---

---

**mN629**  
**CPU I/O TRANSCEIVER**  
**CONTENTS**



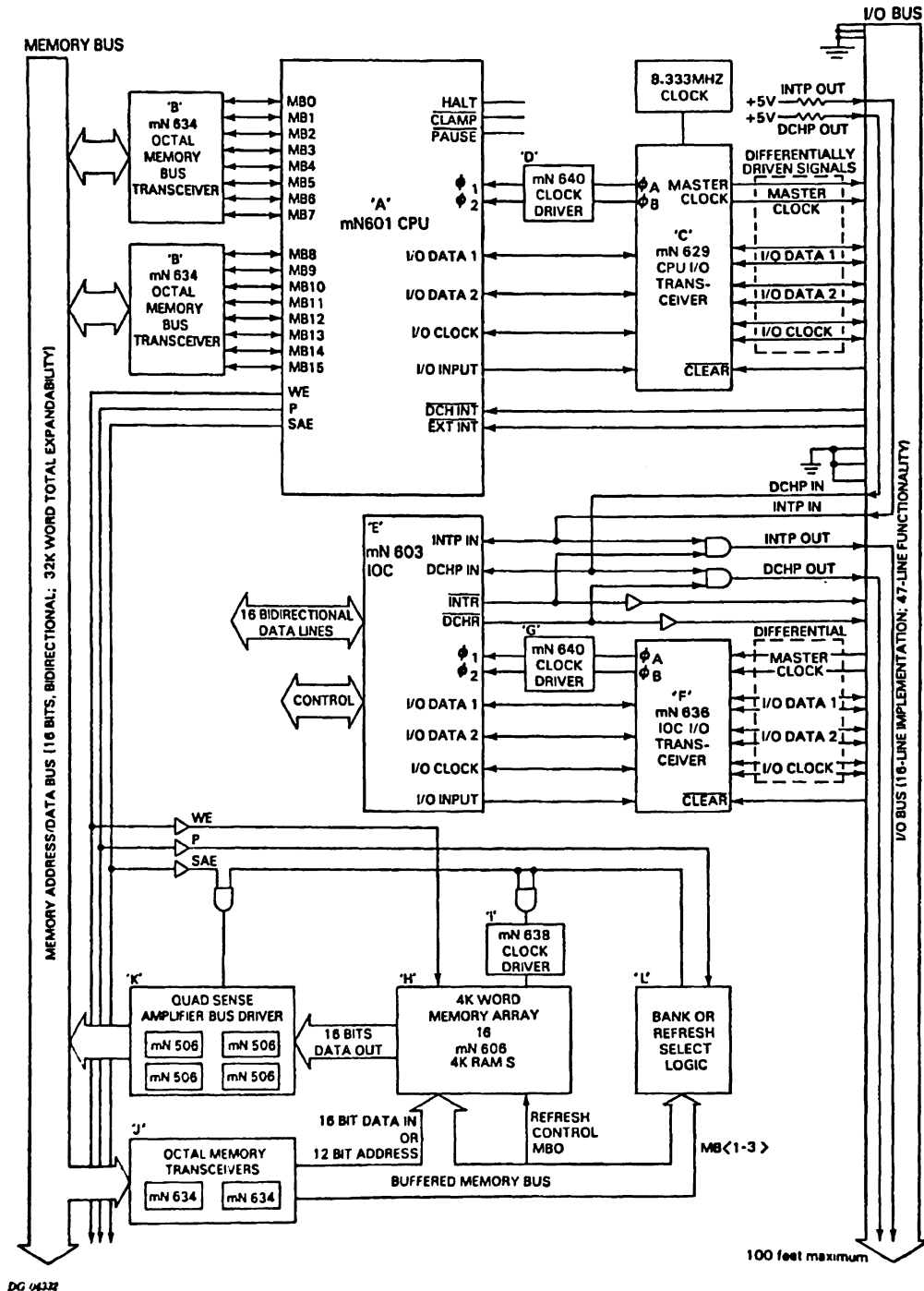
# mN629

## CONTENTS

<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>4</b>	<b>INTERNAL STRUCTURE</b>
<b>5</b>	<b>DATA TRANSMISSION AND RECEPTION</b>
<b>6</b>	<b>Transmit Mode</b>
<b>7</b>	<b>Receive Mode</b>
<b>8</b>	<b>REGISTER RESET CIRCUIT</b>
<b>8</b>	<b>CLOCKS</b>
<b>8</b>	<b>CLEAR</b>
<b>9</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>9</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>9</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>9</b>	<b>DC CHARACTERISTICS</b>
<b>11</b>	<b>AC CHARACTERISTICS</b>
<b>11</b>	<b>Switching Diagrams</b>
<b>12</b>	<b>Transition Timing</b>
<b>13</b>	<b>PACKAGE SPECIFICATIONS</b>



# mN629 CPU I/O TRANSCEIVER



DC 0422



Data General Corporation, Westboro, Massachusetts 01581

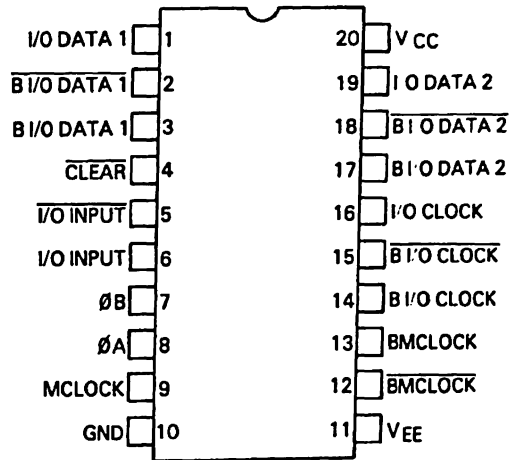
# mN629

## CPU I/O TRANSCEIVER

### FEATURES

- I/O BUS RECEIVER AND TRANSMITTER IN A SINGLE 20-PIN CERAMIC PACKAGE.
- INTERFACES mN601 CPU to microNOVA DIFFERENTIAL I/O BUS
- DIFFERENTIAL DRIVE CAPABILITIES:
  1. PROVIDE HIGH I/O BUS NOISE IMMUNITY.
  2. ALLOW LONG DISTANCE APPLICATIONS. (UP TO 100 FEET)
  3. ALLOW THE USE OF INEXPENSIVE 16-CONDUCTOR RIBBON CABLE.
- PROVIDES ALL SYSTEM TIMING FROM ONE MASTER CLOCK INPUT.

### PACKAGE



DG-01122

### GENERAL DESCRIPTION

The CPU I/O Transceiver chip interfaces the microNOVA CPU with the bi-directional, differential I/O bus. It communicates with peripheral devices via peripheral I/O Transceivers (mN636s). When the mN629 is in receive mode, it passes information to the CPU. When it is in transmit mode, the CPU can send information to I/O Controllers.

The mN629 uses differential drivers and receivers to communicate with up to 20 IOC I/O Transceivers via the I/O bus. This allows high noise immunity over a long bus. An I/O bus length of up to 100 feet is possible using inexpensive 16-conductor ribbon cable.

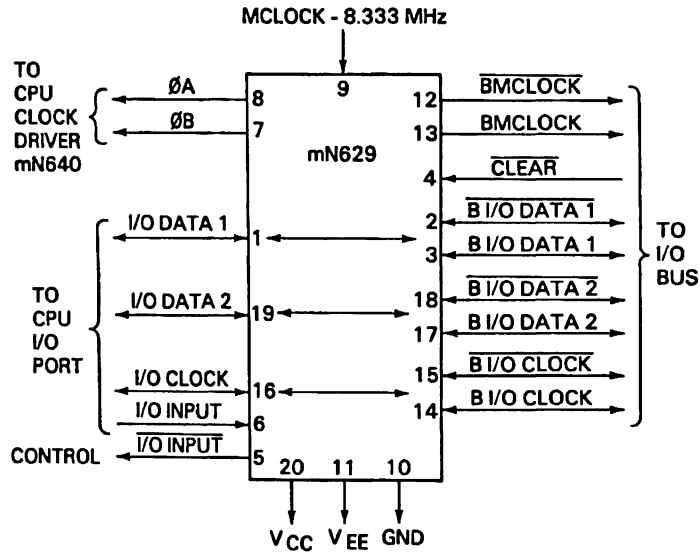
In addition to passing data between the CPU and the I/O bus, the CPU I/O Transceiver provides timing for the entire system. The system master clock is an input to the CPU I/O Transceiver. A two phase clock (ØA and ØB) is generated from this input to provide timing for the high voltage clock drivers required by the CPU. In addition, a differential version of master clock is transmitted to all the mN636 I/O Transceivers to provide synchronized timing for the entire system.



# mN629

## CPU I/O TRANSCEIVER

### FUNCTIONAL PIN CONNECTION DIAGRAM



DC-01183

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
I/O DATA1 I/O DATA2	1 19	I/O I/O	Tri-state, bi-directional data port pins. Transfer serial data between Transceiver and CPU I/O port data pins.
I/O CLOCK	16	I/O	Tri-state, bi-directional clock line used to synchronize transmission and reception of data on I/O DATA1 and I/O DATA2 lines. Note: CLEAR asserted low pulls I/O CLOCK to its low state if Transceiver is in Receive Mode.
I/O INPUT	6	IN	High = mN629 in receive mode. Low = mN629 in transmit mode.
I/O INPUT	5	OUT	Inverted output of I/O INPUT.
CLEAR	4	IN	When asserted, clears all internal flip/flops and pulls I/O CLOCK (pin 16) to the low state if the Transceiver is in the Receive Mode (used to reset mN601).
MCLOCK	9	IN	MicroNOVA system master clock input.
ØA ØB	8 7	OUT OUT	Two-phase clocks operating at one-half the MCLOCK frequency. TTL level open collector outputs.
BMCLOCK BMCLOCK	12 13	OUT OUT	Open emitter, open collector differential pair. Transmits system clocks to all peripheral Transceivers (mN636).

# mN 629

## PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
I/O DATA1	2	I/O	Open emitter, open collector differential pair. Transmit and receive serial data on I/O bus. RECEIVE MODE (I/O INPUT high) - receive differential signals from I/O bus. Clocked by BI/O CLOCK. TTL level signal output on I/O DATA1 pin. TRANSMIT MODE (I/O INPUT low) - TTL level input from I/O DATA1 transmitted differentially on I/O bus. Clocked by MCLOCK.
I/O DATA1	3	I/O	
CO DATA2	18	I/O	Open emitter, open collector differential pair. Transmit and receive serial data on I/O bus. RECEIVE MODE (I/O INPUT high) - receives differential signals from I/O bus. Clocked by BI/O CLOCK. TTL level signal output on I/O DATA2 pin. TRANSMIT MODE (I/O INPUT low) - TTL level input from I/O DATA2 transmitted differentilly on I/O bus. Clocked by MCLOCK.
I/O DATA2	17	I/O	
I/O CLOCK	15	I/O	Open emitter, open collector differential pair. Used to synchronize transmission and reception of BI/O DATA <1,2> pairs. RECEIVE MODE (I/O INPUT high) - receive differential synchronizing clock from I/O bus. Clocks data in on BI/O DATA <1,2> differential pairs. TTL level signal output on I/O CLOCK pin. TRANSMIT MODE (I/O INPUT low) - TTL level input from I/O CLOCK pin used to generate a double frequency clock from MCLOCK; is transmitted differentially to the I/O bus.
I/O CLOCK	14	I/O	
VCC	20		POWER +5.0 ± 0.25 volts
VEE	11		-5.0 ± 0.25 volts
GND	10		Ground

16

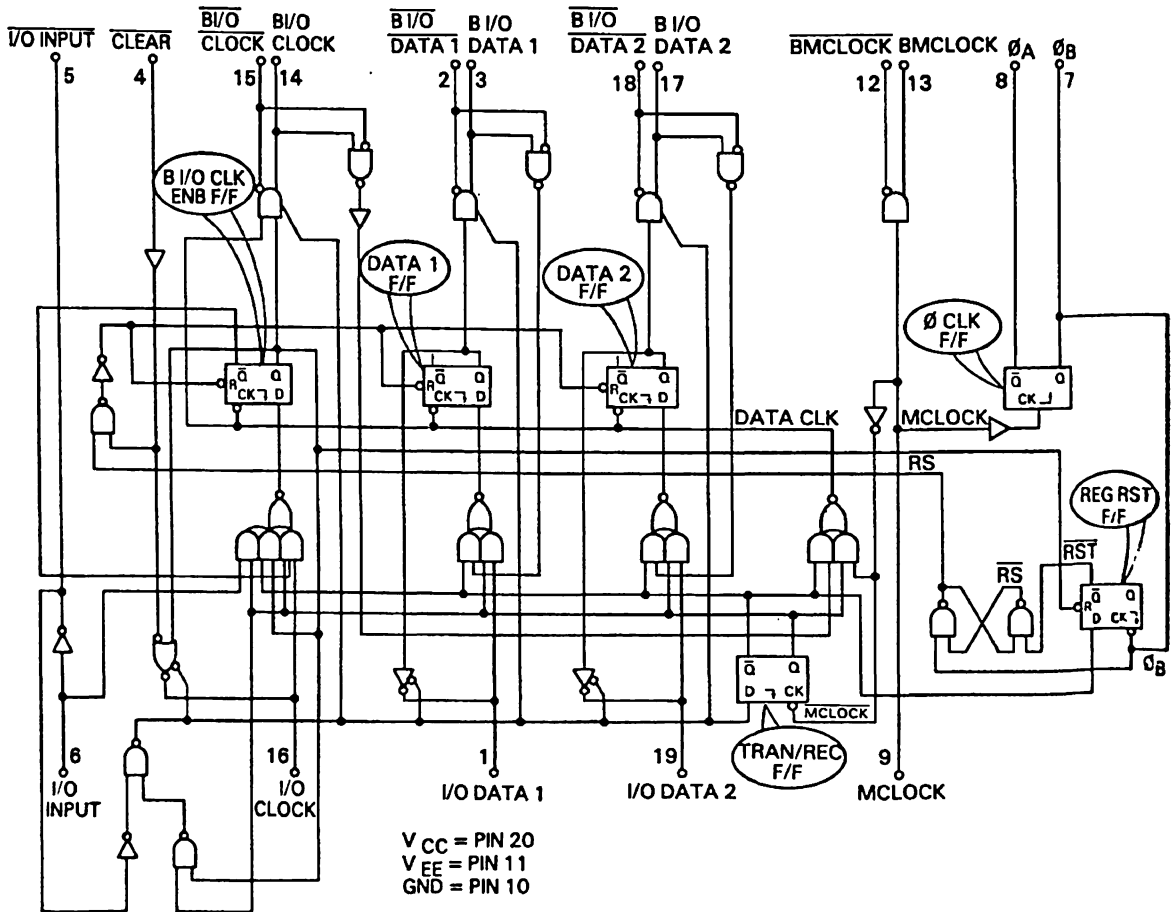
# mN629

## PIN DESCRIPTIONS

### INTERNAL STRUCTURE

The mN629 consists of six flip/flops (F/F), a number of gates, four differential drivers, and three differential receivers. The I/O INPUT pin determines whether the Transceiver is in the Receive mode or the Transmit mode. When I/O INPUT is high, the Transceiver is in the Receive mode. In this case, the I/O DATA1, I/O DATA2, and I/O CLOCK outputs are enabled. When I/O INPUT is asserted low, these output drivers are disabled while the differential drivers are enabled. In this mode, the I/O DATA1, I/O DATA2, and I/O CLOCK inputs provide data to the DATA1 F/F, DATA2 F/F, and B/I/O CLK ENB F/F.

### INTERNAL BLOCK DIAGRAM



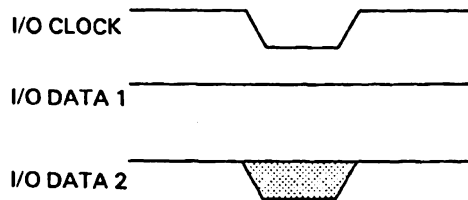
DC-0243

# mN629 DATA TRANSMISSION

## DATA TRANSMISSION AND RECEPTION

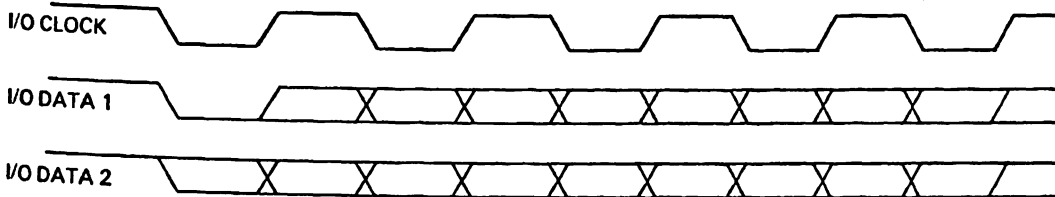
When connected to the microNOVA I/O bus, the mN629 performs two general types of data transfers; short and long as shown below.

### SHORT TRANSFER



DG-04184

### LONG TRANSFER



DG-04185

The short type consists of one I/O CLOCK pulse and transfers two bits. The long type consists of five I/O CLOCK pulses and transfers 18 bits. For the purpose of illustrating the operation of the Transceiver, the long transfer is used as an example. The one pulse transfer is simply a shortened version of the long transfer as far as the Transceiver's operation is concerned. See the Introduction and mN601, Operations Protocol, I/O Operations sections for further explanation of the types of transfers.

The transceiver operates under two modes; transmit and receive. The input waveforms, some internal signals, output waveforms, and a brief discussion of each mode are given below.

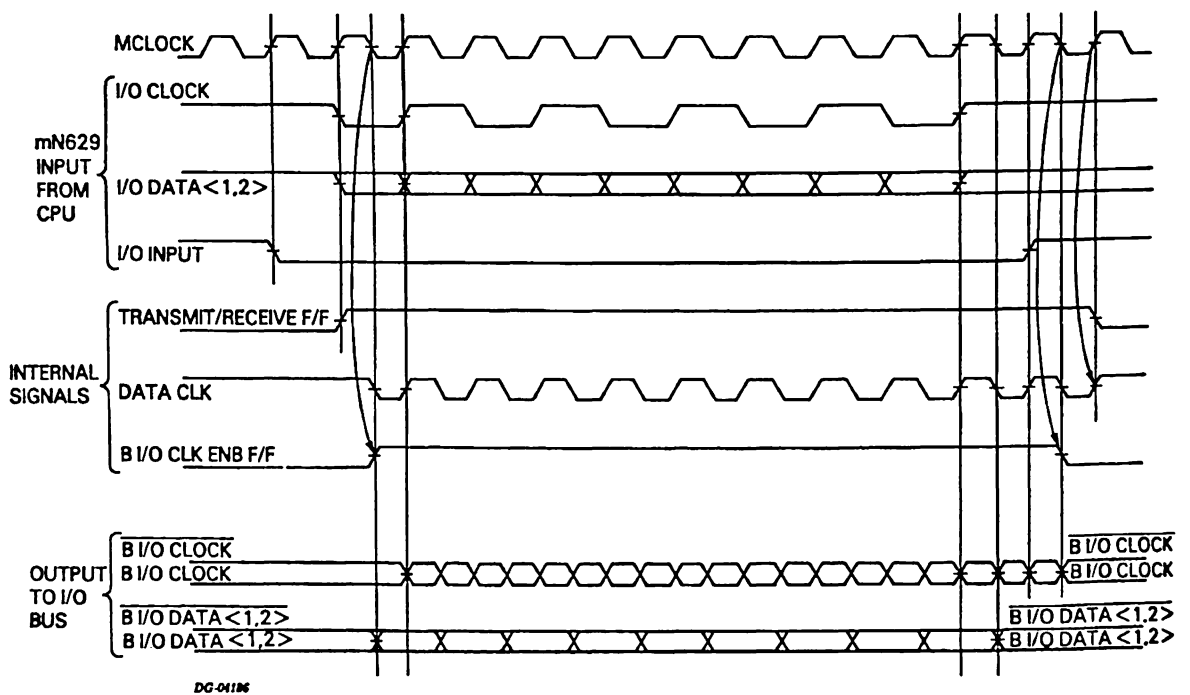
# mN629

## DATA TRANSMISSION

### Transmit Mode

Transmit mode begins when I/O INPUT is asserted low (by the CPU). The next rising edge of MCLOCK sets the Transmit/Receive F/F allowing MCLOCK to clock the DATA CLK line. As shown in the block diagram, DATA CLK clocks data into the DATA1 F/F and DATA2 F/F. These flip/flops supply the data to the differential drivers. In addition, DATA CLK sets the BI/O CLK ENB F/F for the duration of the transmission. The high output of the BI/O CLK ENB F/F allows DATA CLK to be transmitted on the I/O bus via the BI/O CLOCK differential driver. A long transfer from the CPU to the I/O bus is shown in the following timing diagram.

### TRANSMIT WAVEFORMS



**NOTE:** MCLOCK and DATA CLK operate at twice the frequency of I/O CLOCK. This provides a transition of BI/O CLOCK near the middle of every BI/O DATA < 1,2 > pulse allowing the receiving transceiver to sample near the middle of each data bit.

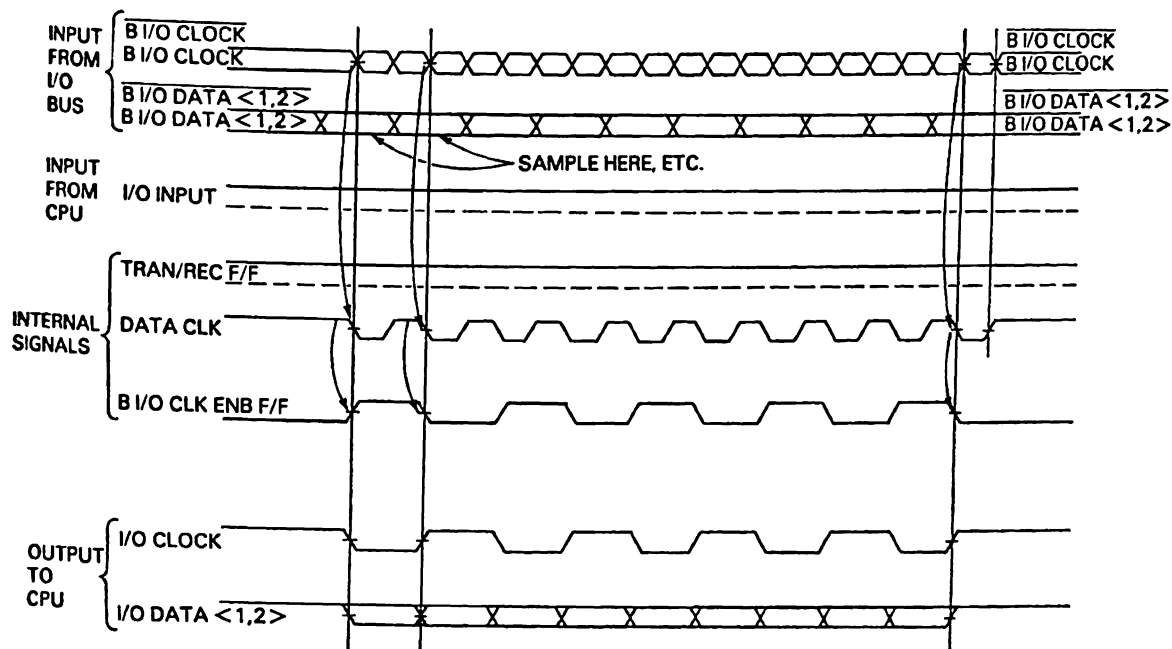
# mN629

## DATA RECEPTION

### Receive Mode

Receive mode begins when I/O INPUT is asserted high. The first BI/O CLOCK transition from the I/O bus initiates data reception. The signals received from the I/O bus are in the same form as those transmitted by the Transceiver. The Transmit/Receive F/F is in the reset state allowing BI/O CLOCK to generate DATA CLK. Data is clocked into the DATA1 F/F and DATA2 F/F on every falling edge of DATA CLK. In addition, the BI/O CLK ENB F/F is toggled on each falling edge of DATA CLK. An inverted output of this flip/flop drives the I/O CLOCK pin. A long transfer from the I/O bus to the CPU is shown in the following timing diagram.

RECEIVE WAVEFORMS



DG-04187

**NOTE:** The last BI/O CLOCK transition causes the Transceiver to sample the differential data lines after the last bit of data has been received. This ensures that the DATA1 F/F and DATA2 F/F are reset to their normal state at the completion of the transfer.

# mN629

## CONTROL

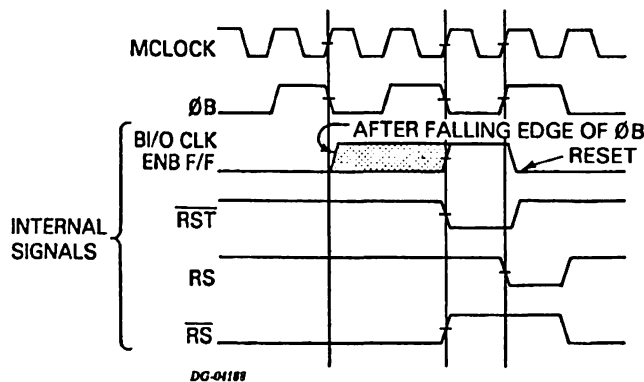
### REGISTER RESET CIRCUIT

This circuit consists of the Register Reset Flip/Flop (REG RST F/F), two NAND gates, and the associated control lines (see lower right corner of Internal Block Diagram). While in the receive mode, the Register Reset Circuit ensures that the BI/O CLK ENB F/F will be reset if set by a spurious transition of BI/O CLOCK.

Under normal operation, the BI/O CLK ENB F/F is clocked either twice or ten times depending on whether the Transceiver receives a short or a long type of transfer. Thus, there should always be an even number of BI/O CLOCK transitions. If for some reason a single transition of BI/O CLOCK is received, the BI/O CLK ENB F/F must be reset to avoid holding I/O CLOCK low and resetting the CPU (see mN601, Operations Protocol, mN601 Status).

If the BI/O CLK ENB F/F is set anywhere during the period shaded in the following timing diagram, it is reset after the rising edge of  $\emptyset B$  as described below.

B I/O CLK ENB F/F RESET TIMING



While the BI/O CLK ENB F/F is in the reset state, its Q output is 0. This forces the REG RST F/F to its reset state. If the BI/O CLK ENB F/F is clocked by a fraudulent transition of BI/O CLOCK, its Q output is no longer 0 and the REG RST F/F is no longer forced to the reset state. On the next falling edge of  $\emptyset B$ , the REG RST F/F is set placing  $\overline{RST}$  low. Though  $\overline{RST}$  is now low, RS remains high since  $\emptyset B$  is low. When  $\emptyset B$  goes high, RS goes low clearing the BI/O CLK ENB F/F. Its Q output goes low immediately resetting the REG RST F/F and placing  $\overline{RST}$  back in its normal state. RS and  $\overline{RS}$  return to their normal state on the next falling edge of  $\emptyset B$ .

The BI/O CLK ENB F/F is not reset during normal reception because the reset circuit has a built-in delay of one MCLOCK period. Even if a transition occurs at the end of the period shaded in the above timing diagram, there is enough time for the second transition to occur before the rising edge of  $\emptyset B$ .

### CLOCKS

The system master clock is applied to MCLOCK. This clock toggles the  $\emptyset CLK$  F/F which provides the two-phase (half-frequency, TTL level) clock for the CPU Clock Driver (mN640). MCLOCK is also differentially driven on the I/O bus (BM CLOCK) to provide synchronized timing for the peripheral Transceivers and Controllers.

### CLEAR

The CLEAR pin is used during system reset. When asserted low, all the mN629's internal flip/flops are reset. If the Transceiver is in the Receive Mode, the I/O CLOCK pin is pulled low (resets the CPU, see mN601). If the Transceiver is in the transmit mode, the I/O CLOCK pin is not affected by the assertion of CLEAR until the transmission is over and I/O INPUT is pulled high.

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{CC}$ .....	7V
Supply voltage, $V_{EE}$ .....	-7V
Input voltage, $V_I$ .....	5.5V
Transceiver output current, $I_{OT}$ .....	80mA
All other output current, $I_O$ .....	50mA
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg.C

**NOTE:** Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply Voltage $V_{CC}$ .....	+5.0 $\pm$ 0.25V
Supply Voltage $V_{EE}$ .....	-5.0 $\pm$ 0.25V
Average Power Dissipation .....	0.95 W
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg. C

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	TYP.	MAX.	UNITS
$V_{IH}$	MCLOCK I/O INPUT I/O DATA1, I/O DATA2 CLEAR, I/O CLOCK		2.0			V
	BI/O CLOCK, $\overline{BI/O\ CLOCK}$ BI/O DATA1, $\overline{BI/O\ DATA1}$ BI/O DATA2, $\overline{BI/O\ DATA2}$				3.2	V
$V_{IL}$	MCLOCK I/O CLOCK I/O INPUT I/O DATA1 I/O DATA2 CLEAR				0.8	V
	BI/O CLOCK, $\overline{BI/O\ CLOCK}$ BI/O DATA1, $\overline{BI/O\ DATA1}$ BI/O DATA2, $\overline{BI/O\ DATA2}$		-1.5			V



# mN629

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (CONT.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	TYP.	MAX.	UNITS
$V_{ID} =  V_{IL} - V_{IH} $	(BI/O CLOCK, $\overline{\text{BI/O CLOCK}}$ ) (BI/O DATA1, $\overline{\text{BI/O DATA1}}$ ) (BI/O DATA2, $\overline{\text{BI/O DATA2}}$ )	REQUIRED VALUE OF $ V_{IL} - V_{IH} $ FOR VALID SIGNAL DETECTION	0.50			V
$I_{IH}$	MCLOCK	$V_I = 2.5 \text{ V}$			3.2	mA
	I/O INPUT	$V_I = 2.5 \text{ V}$			80	$\mu\text{A}$
	I/O DATA1 I/O DATA2 $\overline{\text{I/O CLOCK}}$ CLEAR	$V_I = 2.5 \text{ V}$			40	$\mu\text{A}$
$I_{IL}$	MCLOCK	$V_I = 0.5 \text{ V}$			-3.8	mA
	I/O CLOCK I/O INPUT I/O DATA1 I/O DATA2 CLEAR	$V_I = 0.5 \text{ V}$			-1.6	mA
$I_{ID}$	$\overline{\text{BI/O CLOCK}}$ , $\overline{\text{BI/O CLOCK}}$ $\overline{\text{BI/O DATA1}}$ , $\overline{\text{BI/O DATA1}}$ $\overline{\text{BI/O DATA2}}$ , $\overline{\text{BI/O DATA2}}$	$0.55 \text{ V} \leq V_I \leq 2.0 \text{ V}$			500	$\mu\text{A}$
$V_{OH}$	$\overline{\text{I/O INPUT}}$	$I_O = -400 \mu\text{A}$ $V_{CC} = 4.75 \text{ V}$	2.5			V
	I/O CLOCK I/O DATA1 I/O DATA2	$I_O = -20 \mu\text{A}$ $V_{CC} = 4.75 \text{ V}$	3.0			V
	BI/O CLOCK (OPEN BI/O DATA1 EMITTER	$I_O = -35 \text{ mA}$ $V_{CC} = 5.25 \text{ V}$			3.2	V
	BI/O DATA2 OUTPUTS) BMCLOCK	$I_O = -35 \text{ mA}$ $V_{CC} = 4.75 \text{ V}$	2.0			V
	$\emptyset A, \emptyset B$	$V_{CC} = 4.75 \text{ V}$			5.5	V
$V_{OL}$	$\emptyset A, \emptyset B$	$I_O = 10 \text{ mA}$ $V_{CC} = 4.75 \text{ V}$			0.5	
	I/O INPUT I/O DATA1 I/O DATA2 I/O CLOCK	$I_O = 5 \text{ mA}$ $V_{CC} = 4.75 \text{ V}$			0.5	V
	BI/O CLOCK (OPEN BI/O DATA1 COLLECTOR BI/O DATA2 OUTPUTS) BMCLOCK	$I_O = 35 \text{ mA}$ $V_{CC} = 4.75 \text{ V}$			0.55	V
$I_{OH}$	$\emptyset A, \emptyset B$	$V_O = V_{OH} \text{ MAX.}$			0.25	mA
$I_{CC}$	TRANSMIT MODE	$V_{CC} = 5.25 \text{ V}$		213		mA
	RECEIVE MODE	$V_{EE} = -5.25 \text{ V}$		147		mA
$I_{EE}$		$V_{CC} = 5.25 \text{ V}$ $V_{EE} = -5.25 \text{ V}$		23		mA
$C_I$	ALL INPUTS				7	pF

NOTE: Positive current is into the pin.

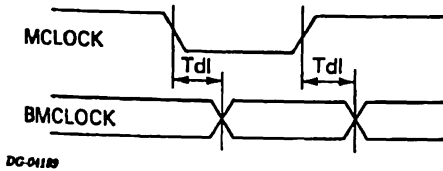
# mN629

## ELECTRICAL SPECIFICATIONS

### A/C CHARACTERISTICS

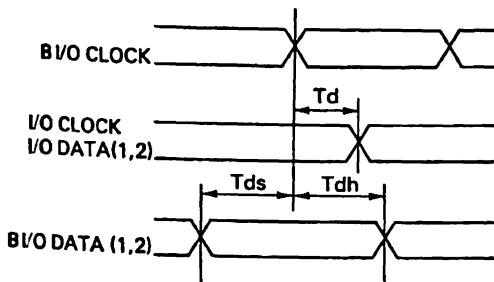
#### Switching Diagrams

##### MCLOCK TRANSMITTER TIMING



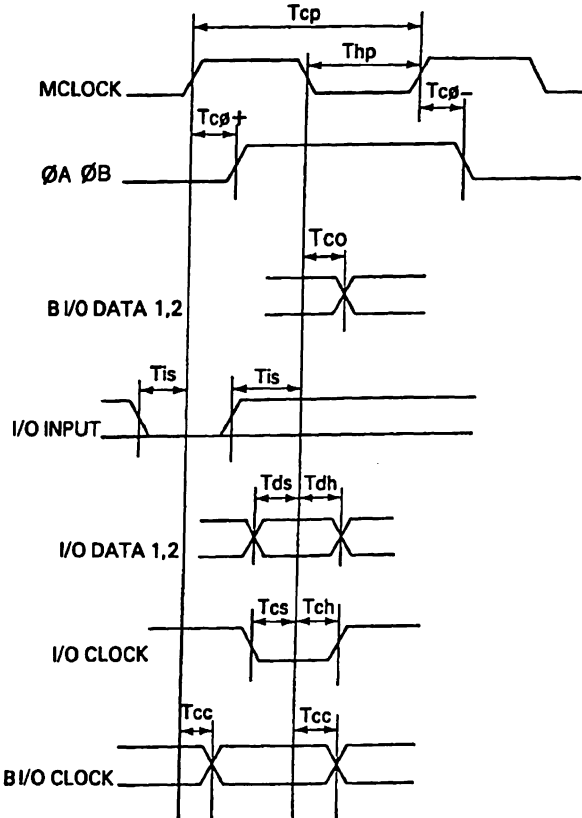
DG-04180

##### RECEIVER TIMING



DG-04130

#### TRANSMITTER TIMING

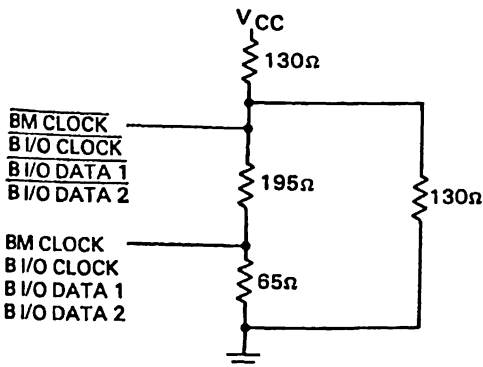


#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED FROM A 1.5 VOLT LEVEL.
- SEE TIMING TABLE ON PAGE 12.

DG-04191

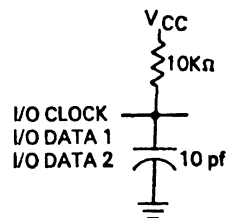
#### TEST CIRCUIT NO. 1



NOTE: ALL RESISTORS ARE  $\pm 1\%$   
ALL CAPACITORS ARE  $\pm 5\%$   
LOAD SHOULD BE CONNECTED TO  
OUTPUT USING LESS THAN  
2 INCHES OF WIRE.

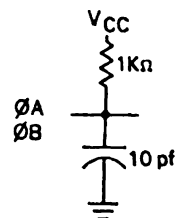
DG-04192

#### TEST CIRCUIT NO. 2



DG-04343

#### TEST CIRCUIT NO. 3



DG-04344

# mN629

## ELECTRICAL SPECIFICATIONS

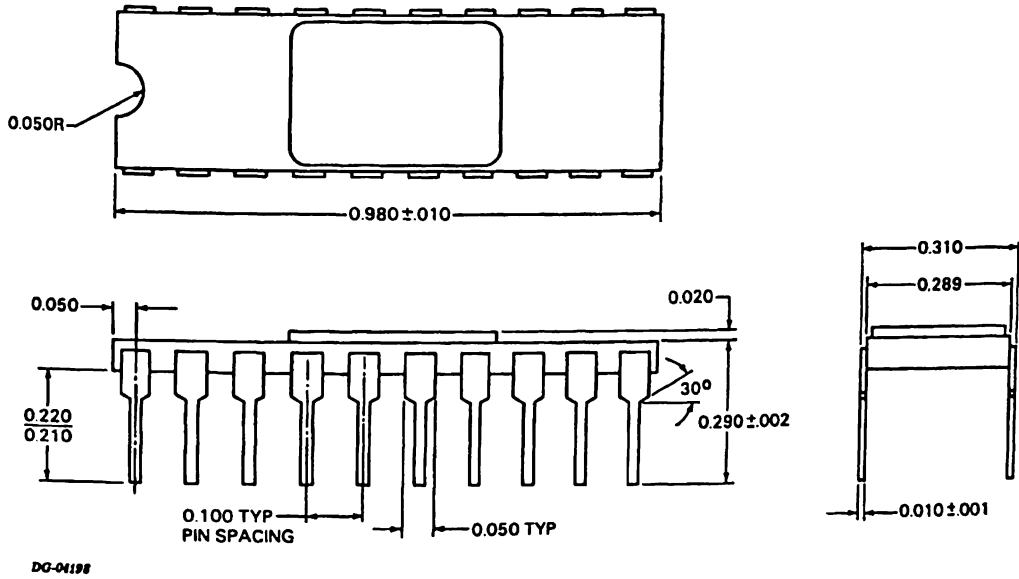
### Transition Timing

SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNIT	TEST CIRCUIT
$T_{cp}$	MASTER CLOCK PERIOD	99	125	ns	
$T_{hp}$	HALF CYCLE PERIOD	40	60	ns	
$T_{c\phi+}$	DELAY FROM MCLOCK TO $\phi A, \phi B$ (RISING EDGE)	30	50	ns	NO. 3
$T_{c\phi-}$	DELAY FROM MCLOCK TO $\phi A, \phi B$ (FALLING EDGE)	$T_{c\phi+}$	$(T_{c\phi+})+10$	ns	NO. 3
$T_{co}$	DELAY FROM MCLOCK TO BI/O DATA	25	35	ns	NO 1
$T_{is-}$	SET UP TIME FOR I/O INPUT	25	---	ns	
$T_{is+}$	SET UP TIME FOR I/O INPUT	10	---	ns	
$T_{ds}$	DATA SETUP TIME	30	---	ns	NO. 1.2
$T_{dh}$	DATA HOLD TIME	0	---	ns	NO. 1.2
$T_{cs}$	I/O CLOCK SETUP TIME	10	---	ns	NO. 1
$T_{ch}$	I/O CLOCK HOLD TIME	10	---	ns	NO 1
$T_{cc}$	DELAY FROM MCLOCK TO BI/O CLOCK	15	60	ns	NO. 1
$T_d$	DELAY FROM BI/O CLOCK TO I/O CLOCK AND I/O DATA(1,2)	30	50	ns	NO 2
$T_{dl}$	DELAY FROM MCLOCK TRANSITION TO BMCLOCK TRANSITION	10	45	ns	NO 1

# mN629

## PACKAGE SPECIFICATION

### PACKAGE SPECIFICATION



---

---

**mN634**  
**OCTAL MEMORY TRANSCEIVER**  
**CONTENTS**

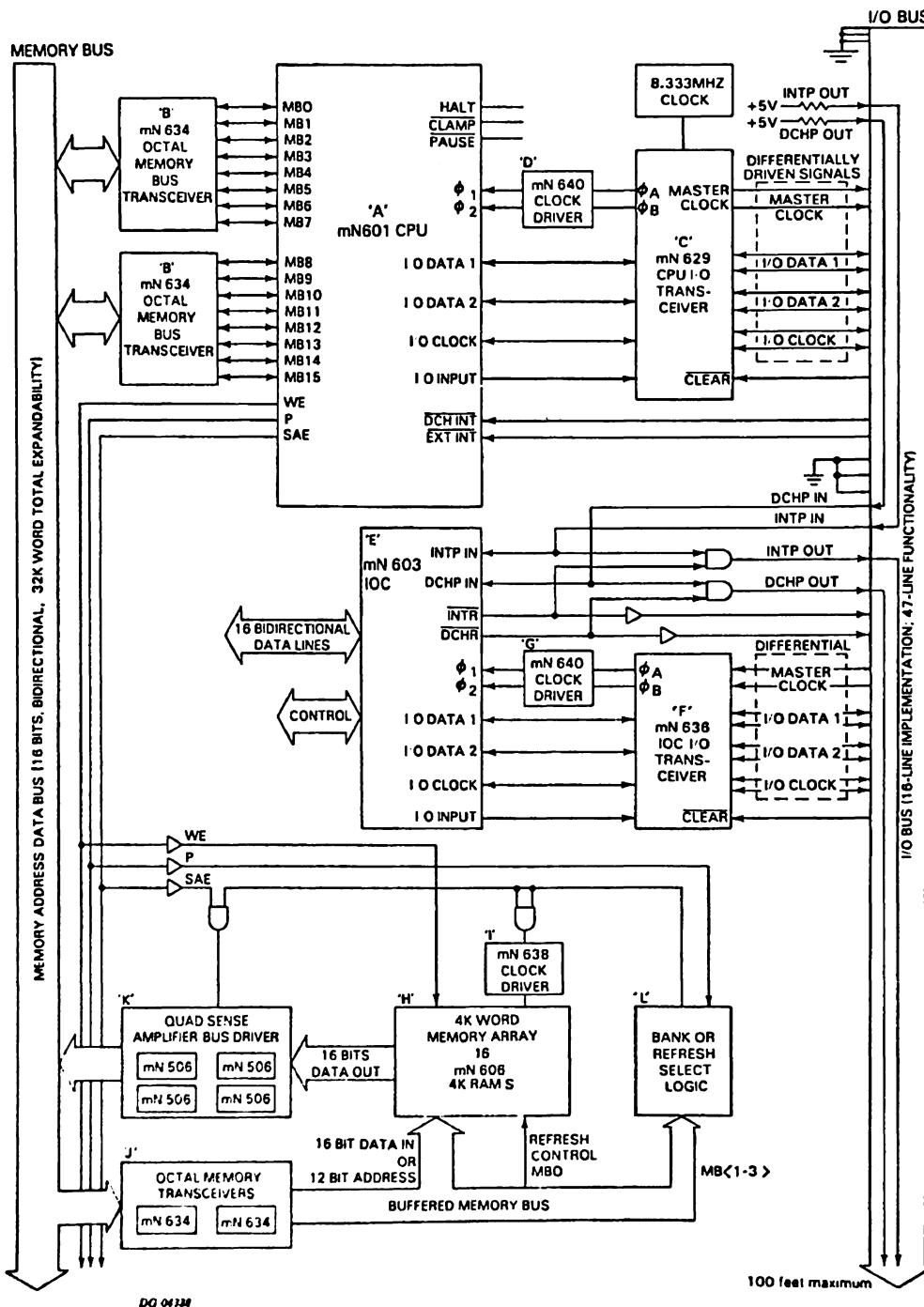


# **mN634**

## **CONTENTS**

<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>3</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>3</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>3</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>3</b>	<b>DC CHARACTERISTICS</b>
<b>4</b>	<b>AC CHARACTERISTICS</b>
<b>4</b>	<b>Switching Diagram</b>
<b>4</b>	<b>Transition Timing</b>
<b>5</b>	<b>PACKAGE SPECIFICATIONS</b>

# mN634 OCTAL MEMORY TRANSCEIVER



DO 0438

100 feet maximum



Data General Corporation, Westboro, Massachusetts 01581

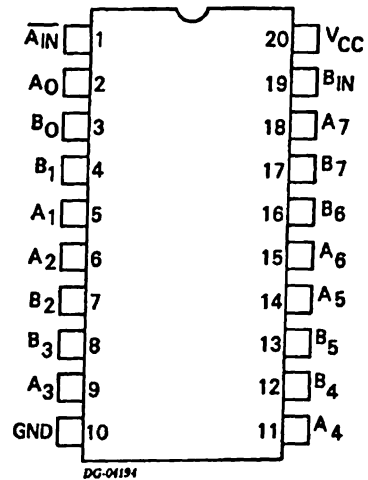
# mN634

## OCTAL MEMORY TRANSCEIVER

### FEATURES

- 8 DRIVERS/RECEIVERS IN A 20-PIN CERAMIC DUAL-IN-LINE PACKAGE.
- NON-INVERTING OPEN-COLLECTOR OUTPUTS.
- SINKS 35mA per OUTPUT.

### PACKAGE



### GENERAL DESCRIPTION

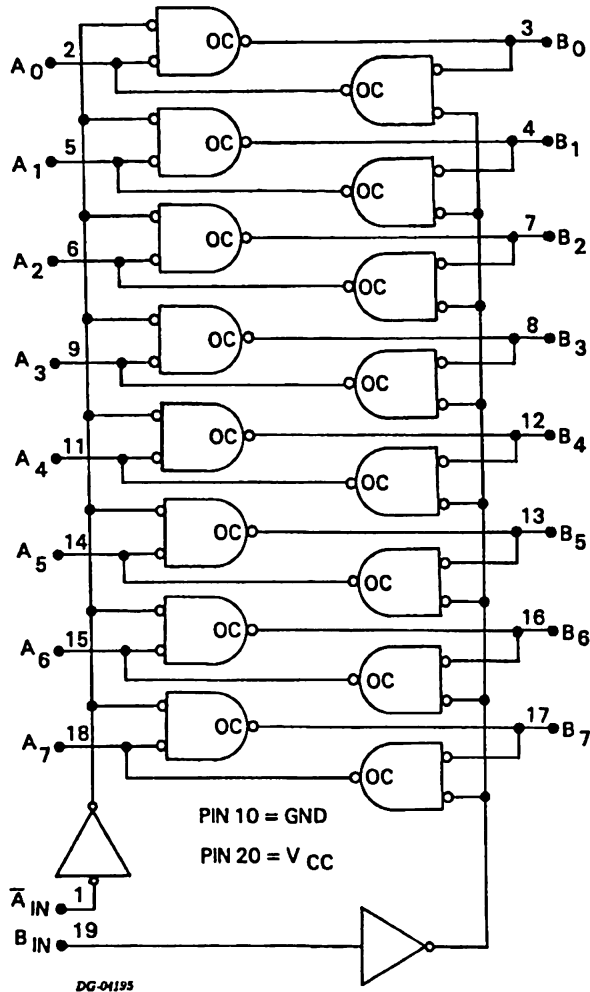
The mN634 Octal Memory Transceiver drives data to and from the microNOVA memory bus. Eight bits may be driven in either direction by the chip. It is used to interface the mN601 CPU to the memory bus as well as to drive the address and data lines for a memory array. See Application Notes.



# mN634

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



### TRUTH TABLE

$\overline{A_{IN}}$	$B_{IN}$	OPERATION
L	L	A data $\rightarrow$ B data
L	H	Low catching latch
H	L	Isolation
H	H	B data $\rightarrow$ A data

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
$A_{0-7}$	2,5,6,9, 11,14, 15,18	I/O	If $A_{in}$ asserted low, A pins are inputs with corresponding B pins as outputs. If $B_{in}$ asserted high, B pins are inputs with corresponding A pins as outputs. If neither $A_{in}$ nor $B_{in}$ is asserted, the A and B pins are open. If both $A_{in}$ and $B_{in}$ are asserted, the A and B outputs can be both high or latched low.
$B_{0-7}$	3,4,7,8, 12,13 16,17	I/O	
$\overline{A_{in}}$	1	In	When asserted low, enables A pins as inputs.
$B_{in}$	19	In	When asserted high, enables B pins as inputs.
$V_{CC}$	20		+5.0 $\pm$ 0.25 volts
GND	10		Ground

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{CC}$ .....	+7V
Input voltage, $V_I$ .....	5.5V
Output Current, $I_O$ .....	60mA
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg.C

**NOTE:** Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{CC}$ .....	5.0 + 0.25V
Average power dissipation .....	1.25 W
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg.C

#### DCC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	TYP.	MAX.	UNITS
$V_{IH}$	$A_{0-7}, B_{0-7}$ $\overline{A}_{IN}, \overline{B}_{IN}$		2.0			V
$V_{IL}$	$A_{0-7}, B_{0-7}$ $\overline{A}_{IN}, \overline{B}_{IN}$				.8	V
$I_{IH}$	$\overline{A}_{IN}, \overline{B}_{IN}$	$V_I = 2.4V$			0.06	mA
		$V_I = 5.5V$			1	mA
	$A_{0-7}, B_{0-7}$	$V_O = 4V$			0.06	mA
		$V_O = 5.5V$			0.25	mA
$I_{IL}$	$A_{0-7}, B_{0-7}$ $\overline{A}_{IN}, \overline{B}_{IN}$	$V_I = 0.5V$			-2	mA
$V_{OL}$	$A_{0-7}, B_{0-7}$	$I_O = 35mA$			0.65	V
$I_{CC}^*$		$V_{CC} = 5.25V$ (INPUTS LOW)		217	280	mA
		$V_{CC} = 5.25V$ (INPUTS HIGH)		115	170	mA
$C_O$		5 V OUTPUT, (OUTPUT OFF)		5		pF
		0 V OUTPUT, (OUTPUT OFF)		25		pF

**NOTE:** Positive current is into the pin.

\*It is recommended not to exceed a 50% duty cycle for all outputs low.

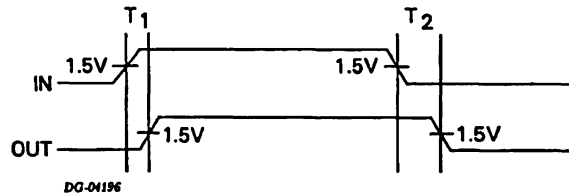
# mN634

## ELECTRICAL SPECIFICATIONS

### AC CHARACTERISTICS

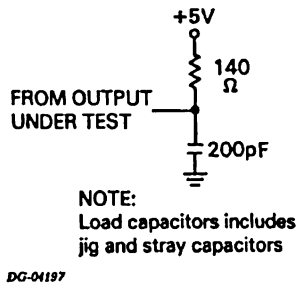
#### Switching Diagram

#### INPUT TO OUTPUT SWITCHING

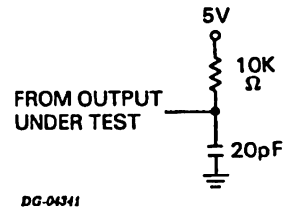


#### TEST CONDITIONS

##### Test Circuit No.1



##### Test Circuit No.2



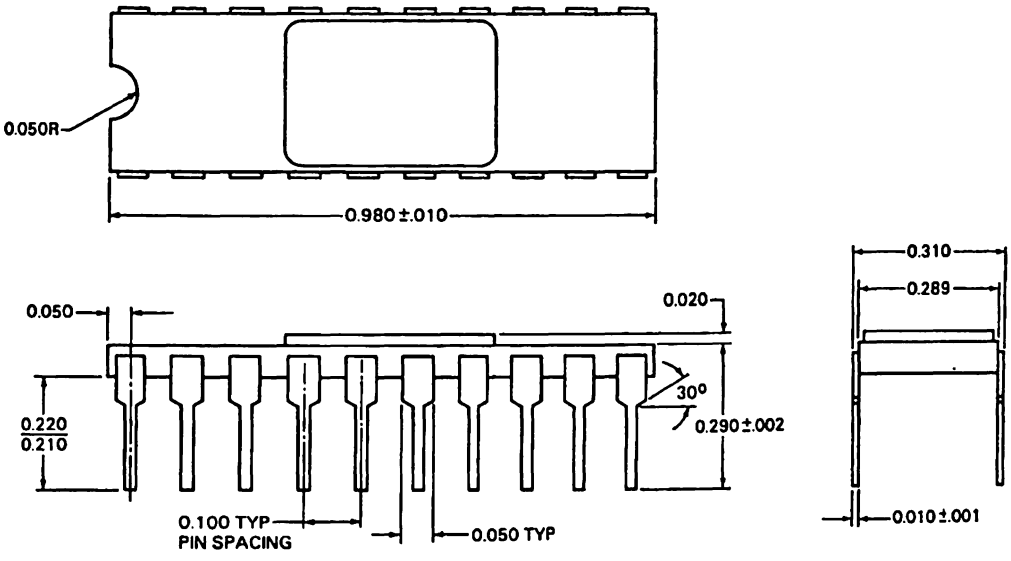
#### Transition Timing

SYMBOL	CHARACTERISTIC	CONDITIONS	MAX.	UNITS
T <sub>1</sub>	INPUT TO OUTPUT DELAY FOR RISING TRANSITION	TEST CIRCUIT 1	35	ns
T <sub>2</sub>	INPUT TO OUTPUT DELAY FOR FALLING TRANSITION	TEST CIRCUIT 1	25	ns
		TEST CIRCUIT 2	17	ns

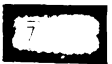
# mN634

## PACKAGE SPECIFICATIONS

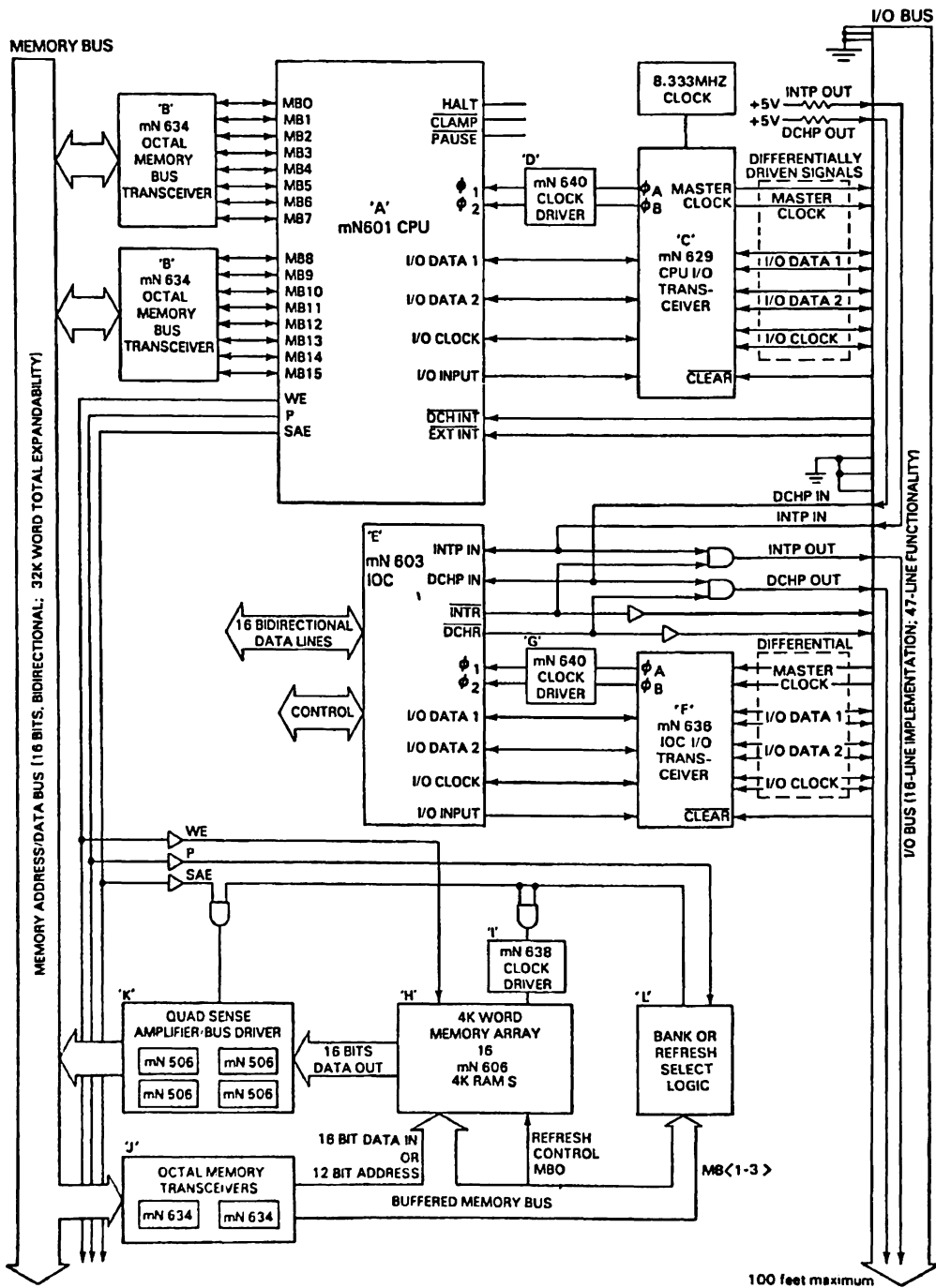
### PACKAGE SPECIFICATIONS



DG-04196



# mN636 IOC I/O TRANSCEIVER



DG-04328



Data General Corporation, Westboro, Massachusetts 01581

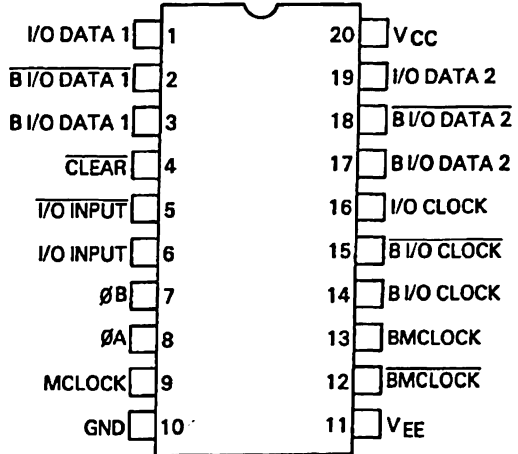
# mN636

## IOC I/O TRANSCEIVER

### FEATURES

- I/O BUS RECEIVER AND TRANSMITTER IN A SINGLE 20-PIN CERAMIC PACKAGE.
- INTERFACES mN603 I/O CONTROLLER TO THE microNOVA DIFFERENTIAL I/O BUS.
- DIFFERENTIAL DRIVE CAPABILITIES:
  1. PROVIDES HIGH I/O BUS NOISE IMMUNITY.
  2. ALLOWS LONG DISTANCE APPLICATIONS (UP TO 100 FEET)
  3. ALLOWS THE USE OF INEXPENSIVE 16-CONDUCTOR RIBBON CABLE.
- SYNCHRONIZES THE IOC (mN603) TO THE CPU (mN601).

### PACKAGE



DG-04199

### GENERAL DESCRIPTION

The IOC I/O Transceiver chip interfaces the microNOVA I/O Controller chip (mN603) with the bi-directional, differential I/O bus. It communicates with the CPU via a CPU I/O Transceiver. When the mN636 is in receive mode, it passes information to the I/O Controller. When it is in transmit mode, the IOC can send information to the CPU.

The mN636 uses differential drivers and receivers to communicate with the CPU Transceiver via the I/O bus. This allows high noise immunity over a long bus. An I/O bus length of up to 100 feet is possible using inexpensive 16-conductor ribbon cable.

In addition to passing data between an I/O Controller and the I/O bus, the mN636 receives the system master clock from the CPU Transceiver. This differential signal generates  $\phi A$  and  $\phi B$  which are used by the high voltage clock driver (mN640) to generate  $\phi 1$  and  $\phi 2$ . These clocks provide synchronized timing for the associated I/O Controller.

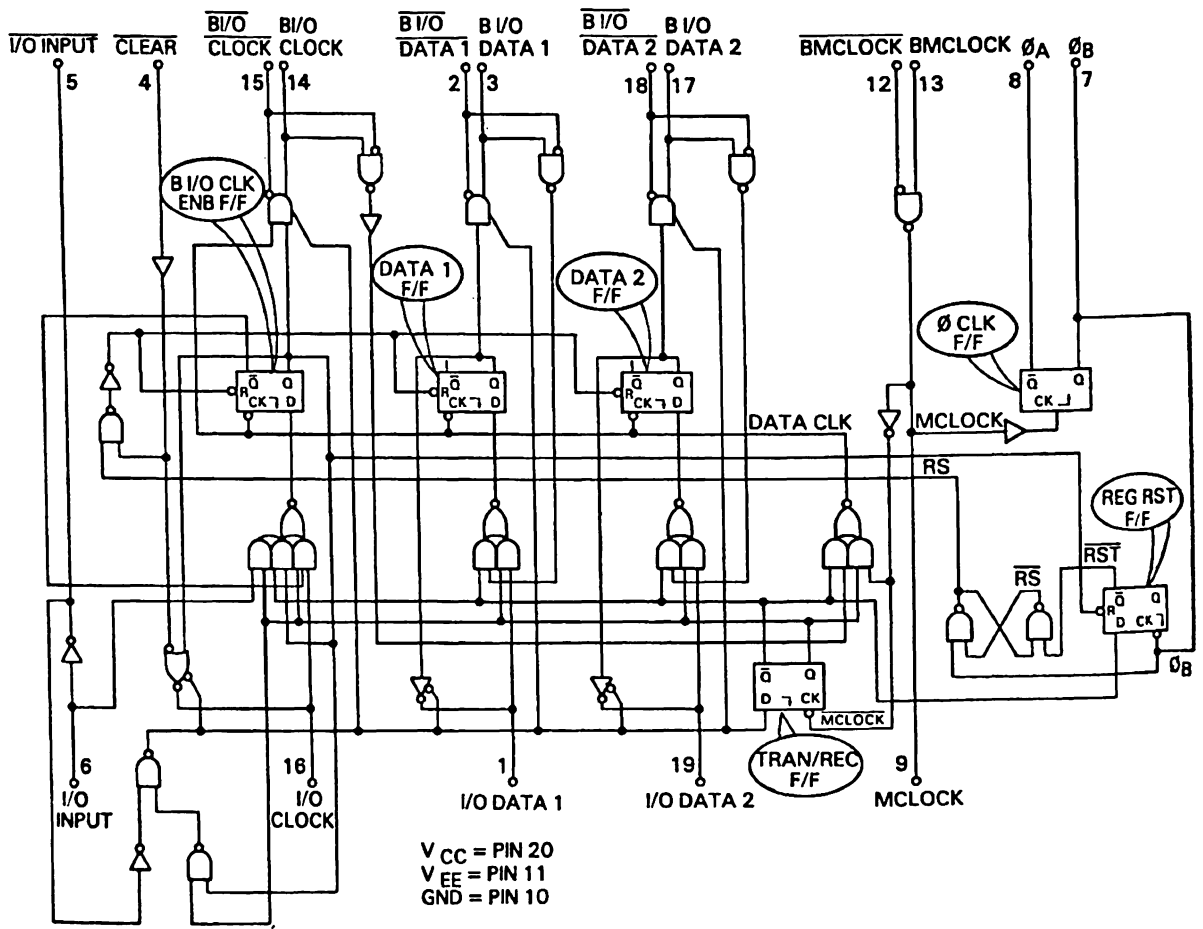
# mN636

## INTERNAL STRUCTURE

### INTERNAL STRUCTURE

The mN636 consists of six flip/flops (F/F), a number of gates, three differential drivers, and four differential receivers. The I/O INPUT pin determines whether the Transceiver is in the Receive mode or the Transmit mode. When I/O INPUT is high, the Transceiver is in the receive mode. In this case, the I/O DATA1, I/O DATA2, and I/O CLOCK outputs are enabled. When I/O INPUT is low, these output drivers are disabled while the differential drivers are enabled. In this mode, the I/O DATA1, I/O DATA2, and I/O CLOCK inputs provide data to the DATA1 F/F, DATA2 F/F, and B/I/O CLK ENB F/F.

### INTERNAL BLOCK DIAGRAM



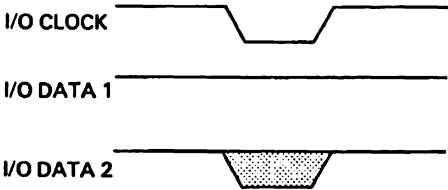
DG-04201

# mN636 DATA TRANSMISSION

## DATA TRANSMISSION AND RECEPTION

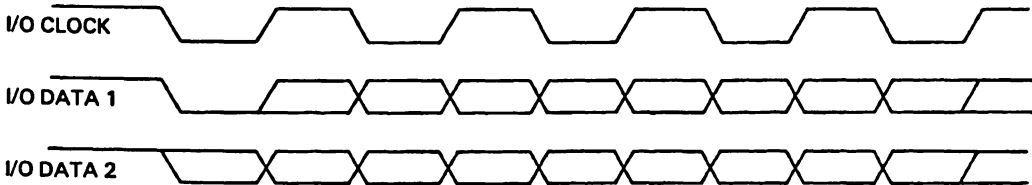
When connected to the microNOVA I/O bus, the mN636 performs two general types of data transfers; short and long as shown below:

### SHORT TRANSFER



DG-04184

### LONG TRANSFER



DG-04185

The short type consists of one I/O CLOCK pulse and transfers two bits. The long type consists of five I/O CLOCK pulses and transfers 18 bits. For the purpose of illustrating the operation of the Transceiver, the long transfer is used as an example. The one pulse transfer is simply a shortened version of the long transfer as far as the Transceiver's operation is concerned. See the Introduction and mN603, I/O Data Port for further explanation of the I/O protocols.

The Transceiver operates under two modes; transmit and receive. The input waveforms, some internal signals, output waveforms and a brief discussion of each mode are given below.



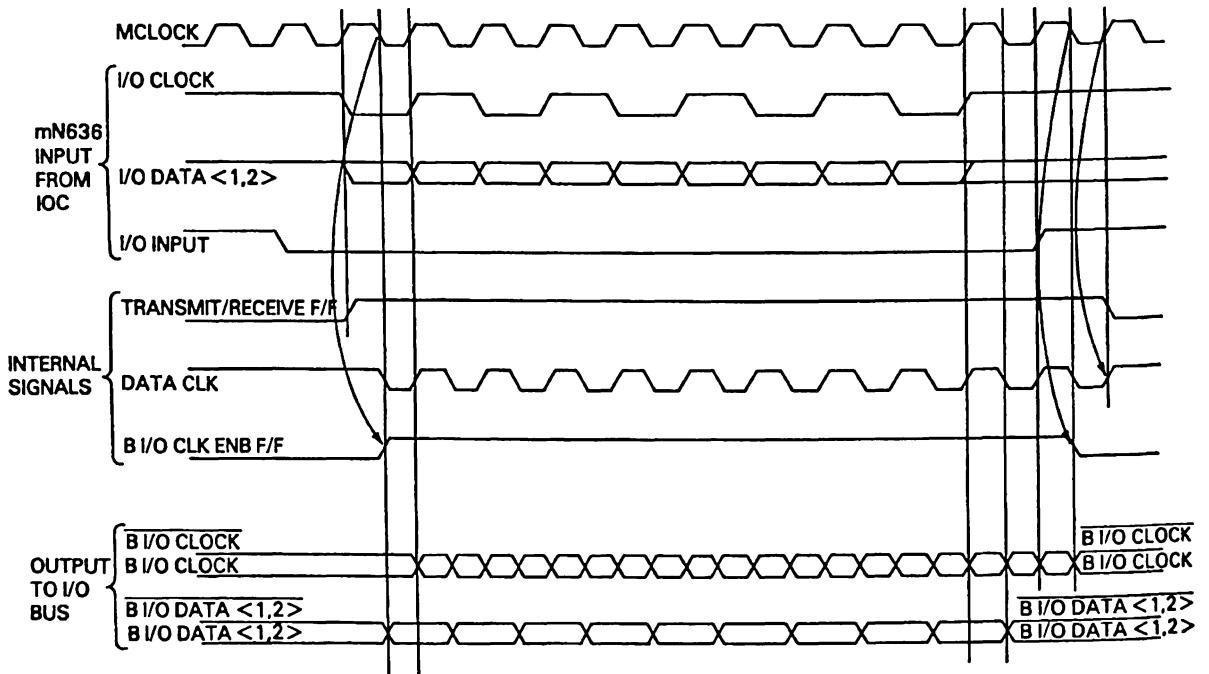
# mN636

## DATA TRANSMISSION

### Transmit Mode

Transmit mode begins when I/O INPUT is asserted low (by an IOC). The next rising edge of MCLOCK sets the Transmit/Receive F/F allowing MCLOCK to clock the DATA CLK line. As shown in the block diagram, DATA CLK clocks data into the DATA1 F/F and DATA2 F/F. These flip/flops supply the data to the differential drivers. In addition, DATA CLK sets the B I/O CLK ENB F/F for the duration of the transmission. The high output of the B I/O CLK ENB F/F allows DATA CLK to be transmitted on the I/O bus via the B I/O CLOCK differential driver. A long transfer from the IOC to the I/O bus is shown in the following timing diagram.

TRANSMIT WAVEFORMS



DG-04204

**NOTE:** MCLOCK and DATA CLK operate at twice the frequency of I/O CLOCK. This provides a transition of B I/O CLOCK near the middle of every B I/O DATA <1,2 > pulse allowing the receiving transceiver to sample near the middle of each data bit.

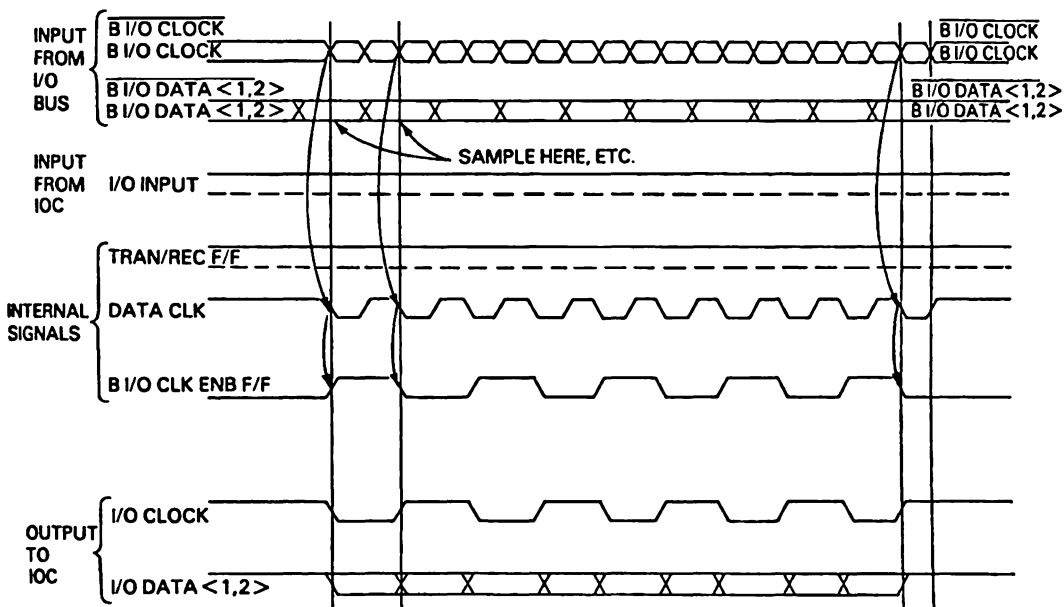
# mN636

## DATA RECEPTION

### Receive Mode

Receive mode begins when I/O INPUT is asserted high. The first BI/O CLOCK transition from the I/O bus initiates data reception. The signals received from the I/O bus are in the same form as those transmitted. The Transmit/Receive F/F is in the reset state allowing BI/O CLOCK to generate DATA CLK. Data is clocked into the DATA1 F/F and DATA2 F/F on every falling edge of DATA CLK. In addition, the BI/O CLK ENB F/F is toggled on each falling edge of DATA CLK. An inverted output of this flip/flop drives the I/O CLOCK pin. A long transfer from the I/O bus to the IOC is shown in the following timing diagram.

### RECEIVE WAVEFORMS



DG 04205

**NOTE:** The last BI/O CLOCK transition causes the Transceiver to sample the differential data lines after the last bit of data has been received. This ensures that the DATA1 and DATA2 F/Fs are reset to their normal state at the completion of the transfer.

# mN636 CONTROL

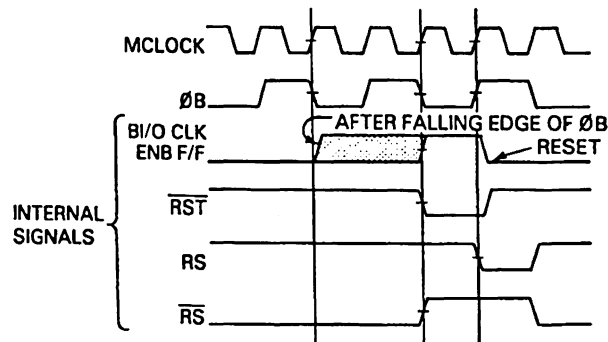
## REGISTER RESET CIRCUIT

This circuit consists of the REG RST F/F, two NAND gates, and the associated control lines (see lower right corner of Internal Block Diagram). While in the receive mode, the Register Reset Circuit ensures that the BI/O CLK ENB F/F will be reset if set by a spurious transition of BI/O CLOCK.

Under normal operation, the BI/O CLK ENB F/F is clocked either twice or ten times depending on whether the Transceiver receives a short or a long type of transfer. Therefore, there should always be an even number of BI/O CLOCK transitions. If for some reason a single transition of BI/O CLOCK is received, the BI/O CLK ENB F/F must be reset to avoid holding I/O CLOCK low and resetting the associated IOC (see mN603, System Requirements).

If the BI/O CLK ENB F/F is set anywhere during the period shaded in the following timing diagram, it is reset after the rising edge of  $\phi B$  as described below.

## B I/O CLK ENB F/F RESET TIMING



DG-01184

While the BI/O CLK ENB F/F is in the reset state, its Q output is 0. This forces the REG RST F/F to its reset state. If the BI/O CLK ENB F/F is clocked by a fraudulent transition of BI/O CLOCK, its Q output is no longer 0 and the REG RST F/F is no longer forced to the reset state. On the next falling edge of  $\phi B$ , the REG RST F/F is set placing RST low. Though RST is now low, RS remains high since  $\phi B$  is low. When  $\phi B$  goes high, RS goes low clearing the BI/O CLK ENB F/F. Its Q output goes low immediately resetting the REG RST F/F and placing RST back in its normal state. RS and RS-bar return to their normal state on the next falling edge of  $\phi B$ .

The BI/O CLK ENB F/F is not reset during normal reception because the reset circuit has a built-in delay of one MCLOCK period. Even if a transition occurs at the end of the period shaded in the above timing diagram, there is enough time for the second transition to occur before the rising edge of  $\phi B$ .

## CLOCKS

The system master clock is received via the BM CLOCK differential pair. This differential receiver clocks the  $\phi$ CLK F/F which provides the two-phase (half-frequency, TTL level) clock for the I/O Controller Clock Driver (mN640). A TTL level version of the system Master Clock is provided on the MCLOCK pin.

## CLEAR

The CLEAR pin is used during system reset. When asserted low, all the mN636's internal flip/flops are reset. If the Transceiver is in the receive mode, the I/O CLOCK pin is pulled low (this resets the associated I/O Controller, see mN603). If the Transceiver is in the transmit mode, the I/O CLOCK pin is not affected by the assertion of CLEAR until the transmission is over and I/O INPUT is pulled high.

# mN636

## ELECTRICAL SPECIFICATIONS

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{CC}$	7V
Supply voltage, $V_{EE}$	-7V
Input voltage, $V_I$	5.5V
Transceiver output current, $I_{OT}$	80mA
All other output current, $I_O$	50mA
Operating free-air temperature range, $T_A$	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$	-55 deg. to 125 deg.C

NOTE: Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{CC}$	$+5.0 \pm 0.25V$
Supply voltage, $V_{EE}$	$-5.0 \pm 0.25V$
Average power dissipation	0.95 W
Operating free-air temperature range, $T_A$	0 deg to 70 deg.C

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	TYP.	MAX.	UNITS
$V_{IH}$	I/O INPUT I/O DATA1, I/O DATA2 CLEAR, I/O CLOCK		2.0			V
	BMCLOCK, BMCLOCK BI/O CLOCK, BI/O CLOCK BI/O DATA1, BI/O DATA1 BI/O DATA2, BI/O DATA2				3.2	V
$V_{IL}$	I/O CLOCK I/O INPUT I/O DATA1 I/O DATA2 CLEAR				0.8	V
	BI/O CLOCK, BI/O CLOCK BI/O DATA1, BI/O DATA1 BI/O DATA2, BI/O DATA2 BMCLOCK, BMCLOCK		-1.5			V

# mN636

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (CONT.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	TYP.	MAX.	UNITS
$V_{ID}$ = $ V_{IL} - V_{IH} $	( <u>B/I/O CLOCK</u> , <u>B/I/O CLOCK</u> ) ( <u>B/I/O DATA1</u> , <u>B/I/O DATA1</u> ) ( <u>B/I/O DATA2</u> , <u>B/I/O DATA2</u> ) ( <u>BMCLOCK</u> , <u>BMCLOCK</u> )	REQUIRED VALUE OF $ V_{IL} V_{IH} $ FOR VALID SIGNAL DETECTION	0.50			V
$I_{IH}$	I/O INPUT	$V_I = 2.5V$			80	$\mu A$
	I/O DATA1 I/O DATA2 I/O CLOCK CLEAR	$V_I = 2.5V$			40	$\mu A$
$I_{IL}$	I/O CLOCK I/O INPUT I/O DATA1 I/O DATA2 CLEAR	$V_I = 0.5V$			-1.6	mA
$I_{ID}$	<u>B/I/O CLOCK</u> , <u>B/I/O CLOCK</u> <u>B/I/O DATA1</u> , <u>B/I/O DATA1</u> <u>B/I/O DATA2</u> , <u>B/I/O DATA2</u> <u>BMCLOCK</u> , <u>BMCLOCK</u>	$0.55V \leq V_I \leq 2.0V$			500	$\mu A$
$V_{OH}$	I/O INPUT	$I_O = -400\mu A$ $V_{CC} = 4.75V$	2.5			V
	I/O CLOCK I/O DATA1 I/O DATA2	$I_O = -20\mu A$ $V_{CC} = 4.75V$	3.0			V
	B/I/O CLOCK (OPEN B/I/O DATA1 EMITTER B/I/O DATA2 OUTPUTS)	$I_O = -35mA$ $V_{CC} = 5.25V$			3.2	V
		$I_O = -35mA$ $V_{CC} = 4.75V$	2.0			V
	MCLOCK	$V_{CC} = 4.75V$ $I_{OH} = 400\mu A$	2.4			V
	$\phi A \phi B$	$V_{CC} = 4.75V$			5.5	V
$V_{OL}$	$\phi 1, \phi 2$	$I_O = 10mA$ $V_{CC} = 4.75V$			0.5	V
	I/O INPUT I/O DATA1 I/O DATA2 I/O CLOCK, MCLOCK	$I_O = 5mA$ $V_{CC} = 4.75V$			0.5	V
	<u>B/I/O CLOCK</u> <u>B/I/O DATA1</u> <u>B/I/O DATA2</u>	$I_O = 35mA$ $V_{CC} = 4.75V$			0.55	V
$I_{OH}$	$\phi A, \phi B$	$V_O = V_{OH} MAX$			0.25	mA
$I_{CC}$	TRANSMIT MODE	$V_{CC} = 5.25V$		213		mA
	RECEIVE MODE	$V_{EE} = -5.25V$		147		
$I_{EE}$		$V_{CC} = 5.25V$ $V_{EE} = -5.25V$		23		mA
$C_I$	ALL INPUTS			7		pf

NOTE: Positive current is into the pin.

DG 04101

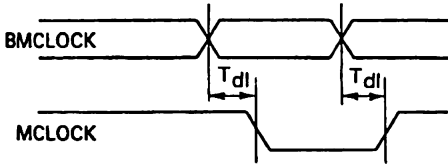
# mN636

## ELECTRICAL SPECIFICATIONS

### AAC CHARACTERISTICS

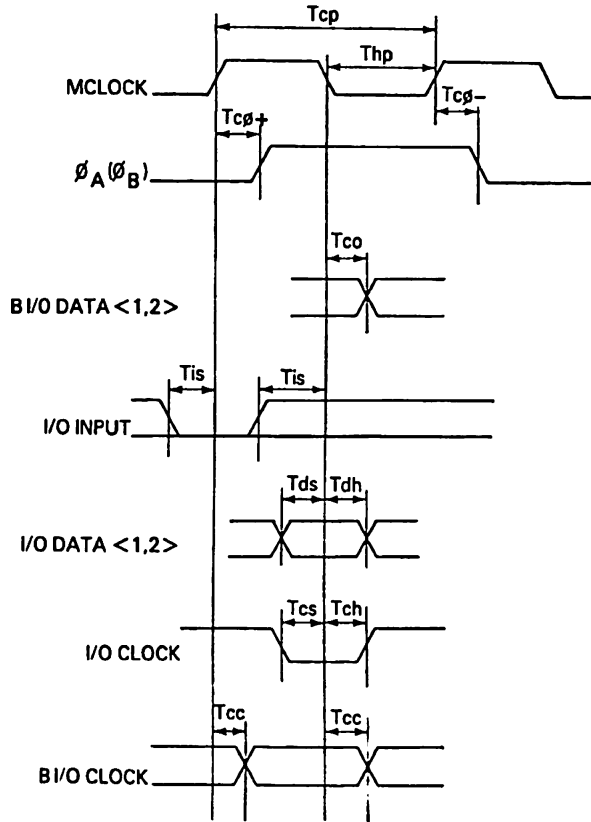
#### Switching Diagrams

#### BMCLOCK RECEIVER TIMING

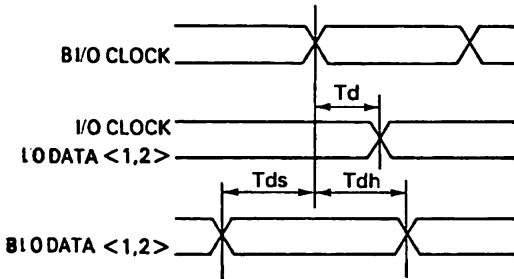


DG-04189

#### TRANSMITTER TIMING

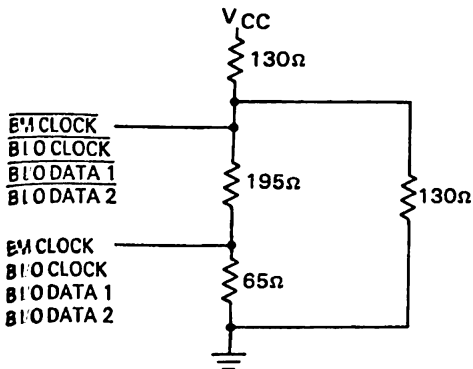


#### RECEIVER TIMING



DG-04190

#### TEST CIRCUIT NO. 1



NOTE: ALL RESISTORS ARE  $\pm 1\%$   
 ALL CAPACITORS ARE  $\pm 5\%$   
 LOAD SHOULD BE CONNECTED TO  
 OUTPUT USING LESS THAN  
 2 INCHES OF WIRE.

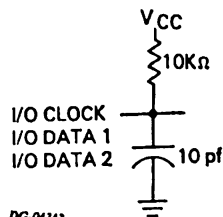
DG-04192

#### NOTES:

- UNLESS OTHERWISE NOTED, TIME INTERVALS ARE MEASURED FROM A 1.5 VOLT LEVEL.
- SEE TIMING TABLE ON PAGE 12.

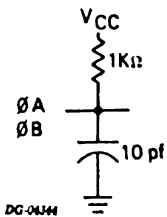
DG-04191

#### TEST CIRCUIT NO. 2



DG-04342

#### TEST CIRCUIT NO. 3



DG-04344

# mN636

## ELECTRICAL SPECIFICATIONS

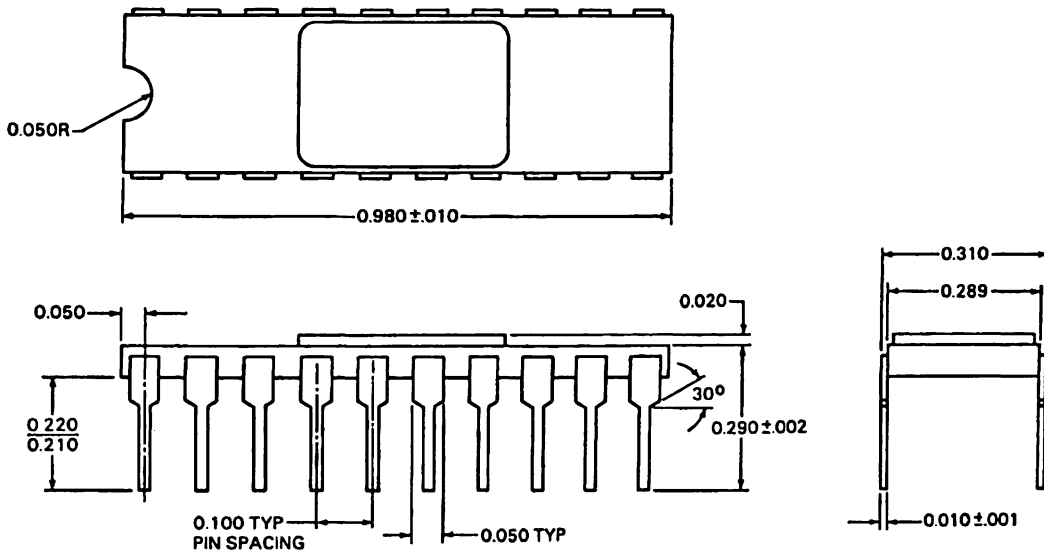
### Transition Timing

SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNIT	TEST CIRCUIT
$T_{cp}$	MASTER CLOCK PERIOD	99	125	ns	
$T_{hp}$	HALF CYCLE PERIOD	40	60	ns	
$T_{c\phi+}$	DELAY FROM MCLOCK TO $\phi_A, \phi_B$ (RISING EDGE)	30	50	ns	NO.3
$T_{c\phi-}$	DELAY FROM MCLOCK TO $\phi_A, \phi_B$ (FALLING EDGE)	$T_{c\phi+}$	$(T_{c\phi+})+10$	ns	NO.3
$T_{co}$	DELAY FROM MCLOCK TO BI/O DATA	25	35	ns	NO.1
$T_{is-}$	SET UP TIME FOR I/O INPUT	25	---	ns	
$T_{is+}$	SET UP TIME FOR I/O INPUT	10	---	ns	
$T_{ds}$	DATA SETUP TIME	30	---	ns	NO.1,2
$T_{dh}$	DATA HOLD TIME	0	---	ns	NO.1,2
$T_{cs}$	I/O CLOCK SETUP TIME	10	---	ns	NO.1
$T_{ch}$	I/O CLOCK HOLD TIME	10	---	ns	NO.1
$T_{cc}$	DELAY FROM MCLOCK TO BI/O CLOCK	15	60	ns	NO.1
$T_d$	DELAY FROM BI/O CLOCK TO I/O CLOCK AND I/O DATA(1,2)	30	50	ns	NO.2
$T_{dl}$	DELAY FROM BMCLOCK TRANSITION TO MCLOCK TRANSITION	5	45	ns	

# mN636

## PACKAGE SPECIFICATION

### PACKAGE SPECIFICATION



DG-04198





---

---

**mN638**  
**MEMORY CLOCK DRIVER**  
**CONTENTS**

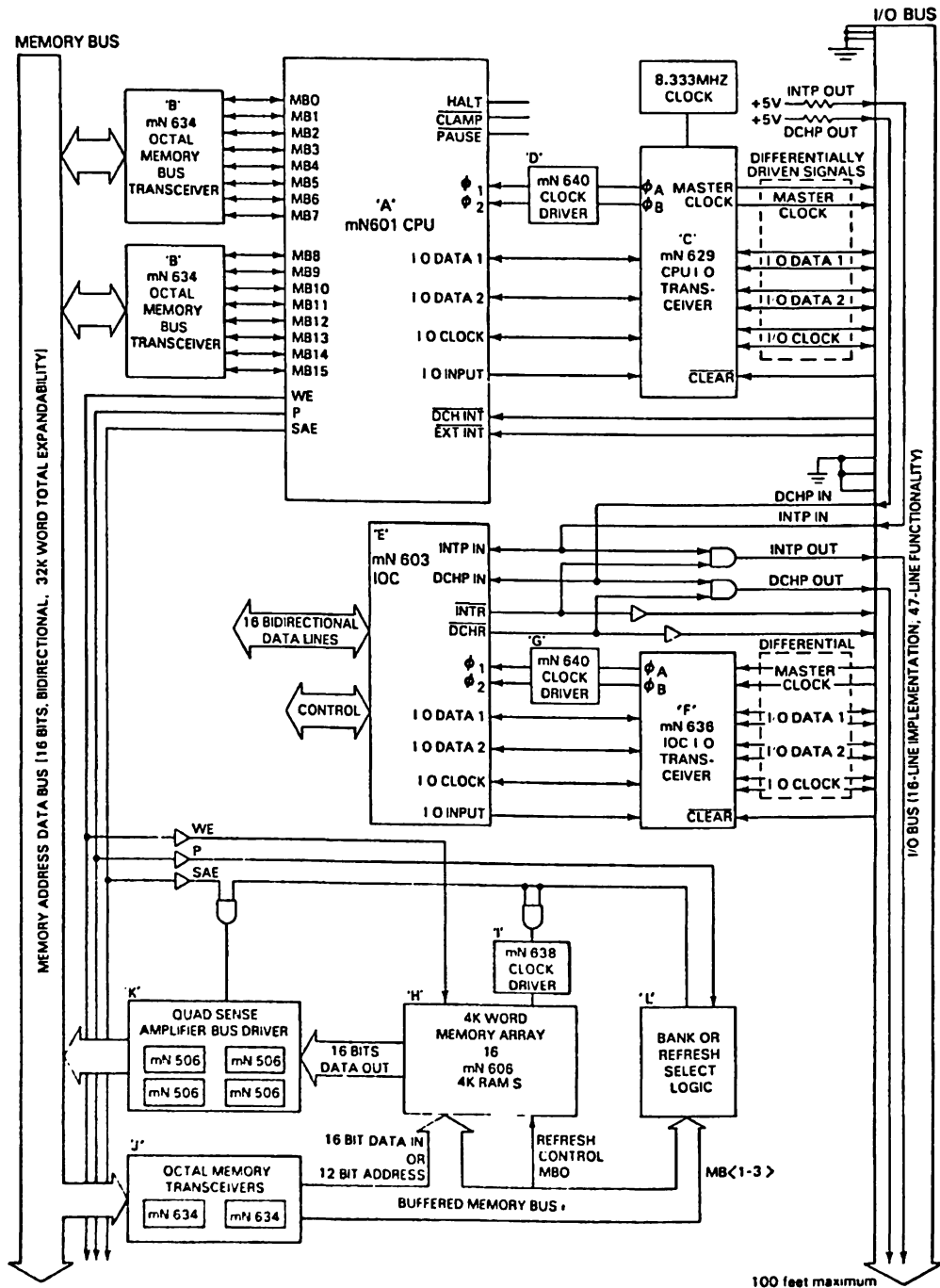


# **mN638**

## **CONTENTS**

- 1      FEATURES**
- 1      GENERAL DESCRIPTION**
- 2      PIN DESCRIPTIONS**
  
- 3      ELECTRICAL SPECIFICATIONS**
- 3      ABSOLUTE MAXIMUM RATINGS**
- 3      RECOMMENDED OPERATING CONDITIONS**
- 3      DC CHARACTERISTICS**
- 5      AC CHARACTERISTICS**
- 5      Switching Diagram**
- 5      Transition Timing**
  
- 6      PACKAGE SPECIFICATIONS**

# mN638 MEMORY CLOCK DRIVER



DG-04342

100 feet maximum



Data General Corporation, Westboro, Massachusetts 01581

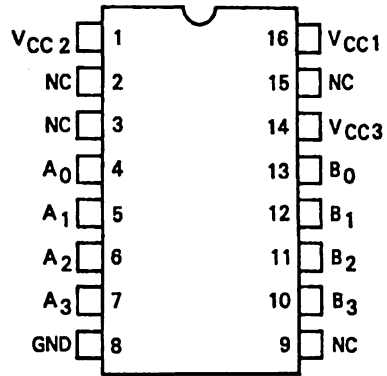
# mN638

## MEMORY CLOCK DRIVER

### FEATURES

- DUAL 3 INPUT BIPOLAR to MOS DRIVER IN A 16-PIN CERDIP PACKAGE.
- TTL COMPATIBLE INPUTS, HIGH VOLTAGE OUTPUTS COMPATIBLE WITH microNOVA mN606 MOS RANDOM ACCESS MEMORY CLOCK INPUTS.

### PACKAGE



NC = no connection

DG-04211

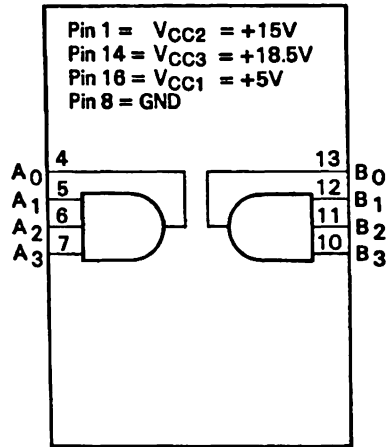
### GENERAL DESCRIPTION

The mN638 Clock Driver chip contains two TTL level to high voltage converters. These may be used to drive the clock input of the mN606 Random Access Memory chip.

# mN638

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



DC-04212

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
$A_1, A_2, A_3$	5,6,7	In	TTL inputs to 3-input AND gate. $A_0$ is high voltage output.
$A_0$	4	Out	High Voltage output of 3-input ( $A_1, A_2, A_3$ ) AND gate.
$B_1, B_2, B_3$	12, 11, 10	In	TTL inputs to 3-input AND gate. $B_0$ is high voltage output.
$B_0$	13	Out	High voltage output of 3-input ( $B_1, B_2, B_3$ ) AND gate.
$V_{CC1}$	16		$+5 \pm 0.25$ volts
$V_{CC2}$	1		$+15 \pm 1.0$ volts
$V_{CC3}$	14		$+18.5 \pm 0.75$ volts
GND	8		Ground

### ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS

Supply voltage, $V_{CC1}$ .....	7V
Supply voltage, $V_{CC2}$ .....	18V
Supply voltage, $V_{CC3}$ .....	21V
Input voltage, $V_I$ .....	5.5V
Output current, $I_O$ .....	60mA
Operating ambient temperature range, $T_A$ .....	0 deg. to 70 deg. C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg. C

**NOTE:** Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### RECOMMENDED OPERATING CONDITIONS

Supply voltage, $V_{CC1}$ .....	$5.0 \pm 0.25V$
Supply voltage, $V_{CC2}$ .....	$15.0 \pm 1V$
Supply voltage, $V_{CC3}$ .....	$18.5 \pm 0.75V$
Average power dissipation .....	1.0 W
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg. C

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{IH}$	A 1-3, B 1-3		2.0		V
$V_{IL}$	A 1-3, B 1-3			0.8	V
$I_{IH}$	A 1-3, B 1-3	$V_{CC1} = 5.25V$ $V_I = 2.4V$		40	$\mu A$
$I_{IL}$	A 1-3, B 1-3	$V_{CC1} = 5.25V$ $V_I = 0.5V$		-1.6	mA

# mN638

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (Cont.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{OH}$	$A_O, B_O$	$V_{CC1} = 4.75V$ $V_{CC2} = 14V$ $V_{CC3} = 17.75V$ $I_O = -20mA$	13		V
$V_{OL}$	$A_O, B_O$	$V_{CC1} = 4.75V$ $I_O = 25mA$	0	0.5	V
$I_{CC1H}$	SUPPLY CURRENT	$V_{CC1} = 5.5V$		9	mA
$I_{CC2H}$	SUPPLY CURRENT	$V_{CC2} = 16V$	-2.7	.25	mA
$I_{CC3H}$	SUPPLY CURRENT	$V_{CC3} = 19.25V$		4	mA
$I_{CC1L}$	SUPPLY CURRENT	$V_{CC1} = 5.5V$		80	mA
$I_{CC2L}$	SUPPLY CURRENT	$V_{CC2} = 16V$		250	$\mu A$
$I_{CC3H}$	SUPPLY CURRENT	$V_{CC3} = 19.25V$		20	mA

**NOTE:** Positive current is into the pin.

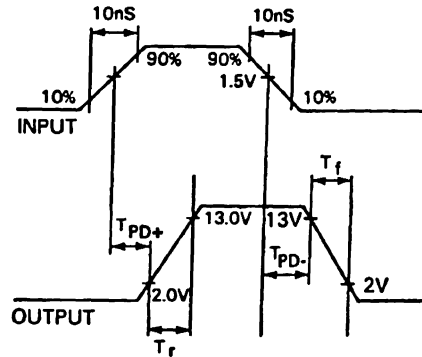


# mN638

## ELECTRICAL SPECIFICATIONS

### AC CHARACTERISTICS

#### Switching Diagram

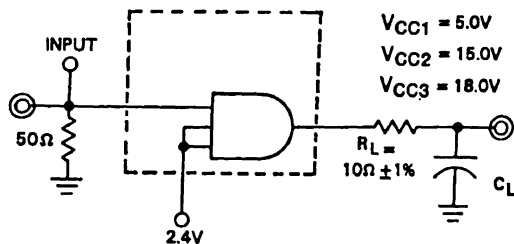


DG-04213

#### Transition Timing

SYMBOL	CHARACTERISTIC	CONDITIONS	MIN.	MAX.	UNITS
$T_r$	OUTPUT RISE TIME (A <sub>O</sub> , B <sub>O</sub> )	$C_L = 15\text{pF}$	5		ns
		$C_L = 470\text{pF}$		25	ns
$T_f$	OUTPUT FALL TIME (A <sub>O</sub> , B <sub>O</sub> )	$C_L = 15\text{pF}$	5		ns
		$C_L = 470\text{pF}$		25	ns
$T_{PD+}$	PROPAGATION DELAY	$C_L = 15\text{pF}$	3		ns
$T_{PD-}$		$C_L = 470\text{pF}$		20	ns

#### TEST CONDITIONS



1. Input Pulse: Amplitude = 0 to .3v  
Width = 400nS (50% to 50%)  
 $t_r = t_f = 10\text{nS}$   
Freq. = 1.67MHz

NOTE: LOAD CAPACITOR INCLUDES  
JIG AND WIRING CAPACITANCE.

DG-04214



**mN640**  
**CPU/IOC CLOCK DRIVER**  
**CONTENTS**



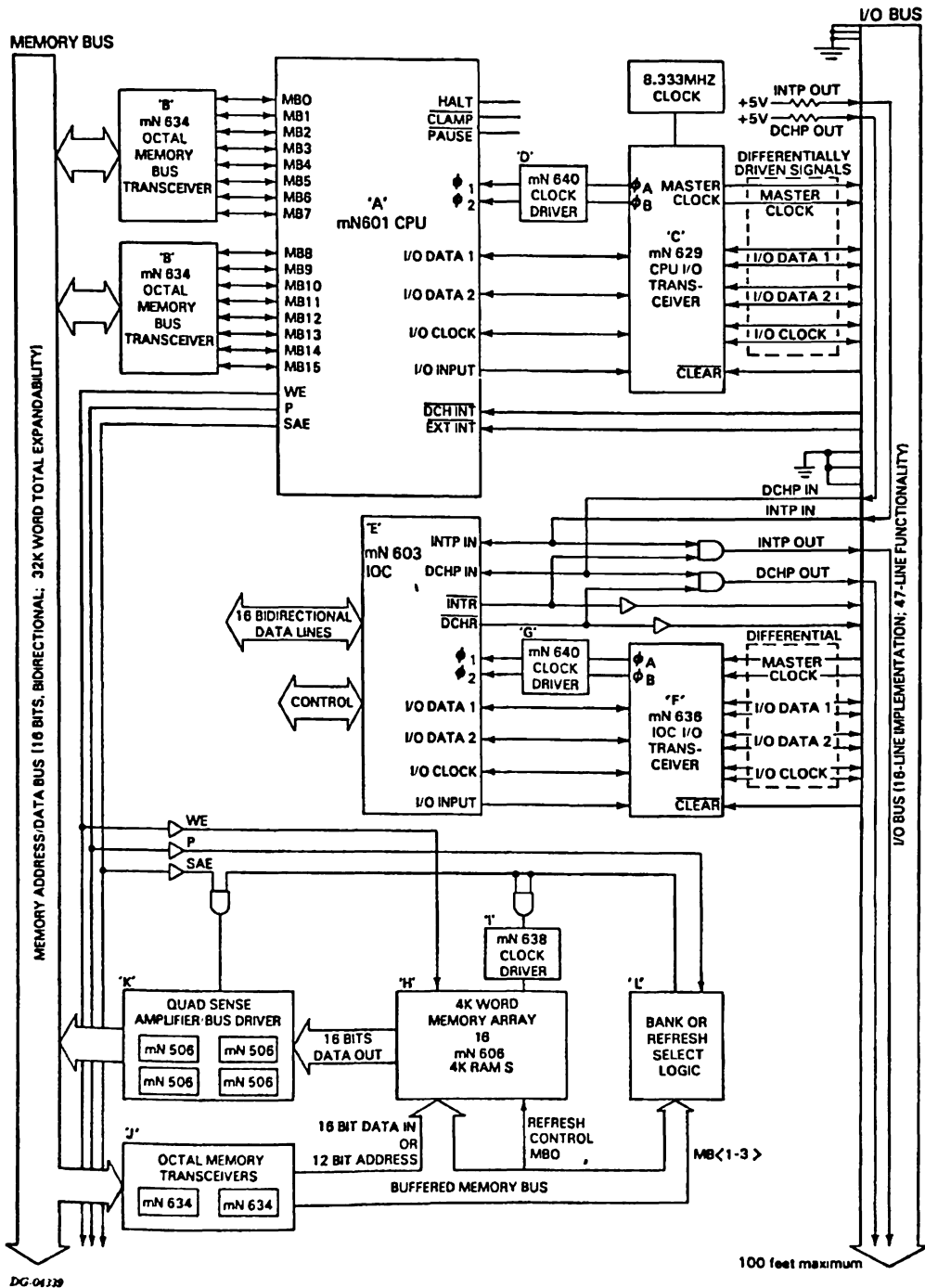
# mN640

## CONTENTS

<b>1</b>	<b>FEATURES</b>
<b>1</b>	<b>GENERAL DESCRIPTION</b>
<b>2</b>	<b>PIN DESCRIPTIONS</b>
<b>3</b>	<b>ELECTRICAL SPECIFICATIONS</b>
<b>3</b>	<b>ABSOLUTE MAXIMUM RATINGS</b>
<b>3</b>	<b>RECOMMENDED OPERATING CONDITIONS</b>
<b>3</b>	<b>DC CHARACTERISTICS</b>
<b>5</b>	<b>AC CHARACTERISTICS</b>
<b>5</b>	<b>Switching Diagrams</b>
<b>5</b>	<b>Transition Timing</b>
<b>6</b>	<b>PACKAGE SPECIFICATIONS</b>



# mN640 CPU/IOC CLOCK DRIVER



DC-01329

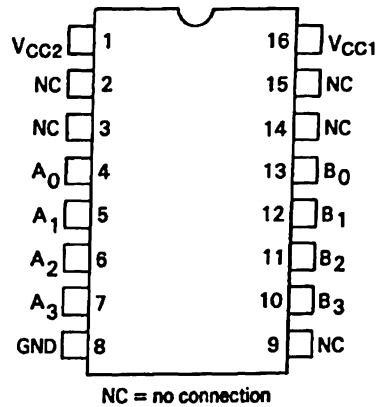
# mN640

## CPU AND I/O CLOCK DRIVER

### FEATURES

- DUAL 3 INPUT BIPOLAR TO MOS DRIVER IN A 16-PIN CERDIP PACKAGE.
- TTL COMPATIBLE INPUTS, HIGH VOLTAGE OUTPUTS COMPATIBLE WITH microNOVA mN601 CPU and mN603 I/O CONTROLLER MOS CLOCK INPUTS

### PACKAGE



DG-04215

### GENERAL DESCRIPTION

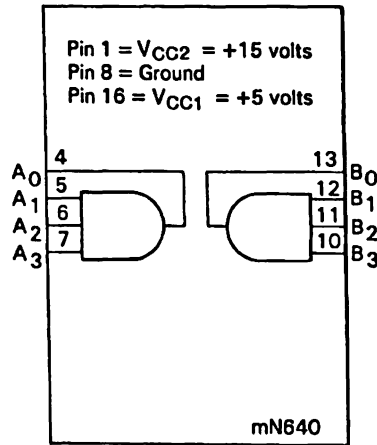
The mN640 Clock Driver chip contains two TTL level to high voltage converter gates. These are used to drive the two-phase clock inputs of the mN601 and mN603 MOS chips. Additional signal shaping circuitry, required to shape the non-overlapping clock signals, is described in the Design Notes.



# mN640

## PIN DESCRIPTIONS

### FUNCTIONAL PIN CONNECTION DIAGRAM



DG-04216

### PIN DESCRIPTIONS

MNEMONIC	PIN NO.	IN/OUT	FUNCTION
A <sub>1</sub> ,A <sub>2</sub> ,A <sub>3</sub>	5,6,7	In	TTL inputs to 3-input AND gate. A <sub>0</sub> is high voltage output.
A <sub>0</sub>	4	Out	High voltage output of 3-input (A <sub>1</sub> ,A <sub>2</sub> ,A <sub>3</sub> ) AND gate.
B <sub>1</sub> ,B <sub>2</sub> ,B <sub>3</sub>	12,11,10	In	TTL inputs to 3-input AND gate. B <sub>0</sub> is high voltage output.
B <sub>0</sub>	13	Out	High voltage output of 3-input (B <sub>1</sub> ,B <sub>2</sub> ,B <sub>3</sub> ) AND gate.
V <sub>CC1</sub>	16		5 ± 0.25 volts
V <sub>CC2</sub>	1		15 ± 1.0 volts
GND	8		Ground

# mN640

## ELECTRICAL SPECIFICATIONS

### ELECTRICAL SPECIFICATIONS

#### Absolute Maximum Ratings

Supply voltage, $V_{CC1}$ .....	7V
Supply voltage, $V_{CC2}$ .....	-18V
Input voltage, $V_I$ .....	5.5V
Output current, $I_O$ .....	60mA
Operating free-air temperature range, $T_A$ .....	0 deg. to 70 deg.C
Storage temperature range, $T_{STG}$ .....	-55 deg. to 125 deg.C

**NOTE:** Subjecting a circuit to conditions either outside these limits or at these limits for an extended period of time may cause irreparable damage to the circuit. These ratings are not intended to be used during the operation of the circuit.

#### Recommended Operating Conditions

Supply voltage, $V_{CC1}$ .....	$5.0 \pm 0.25V$
Supply voltage, $V_{CC2}$ .....	$15.0 \pm 1.0V$
Average power dissipation .....	1.0 W
Operating free-air temperature range $T_A$ .....	-0 deg.C to 70 deg.C

#### DC CHARACTERISTICS

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{IH}$	A 1-3, B 1-3		2.0		V
$V_{IL}$	A 1-3, B 1-3			0.8	V
$I_{IH}$	A 1-3, B 1-3	$V_{CC1} = 5.25V$ $V_I = 2.4V$		40	$\mu A$
$I_{IL}$	A 1-3, B 1-3	$V_{CC1} = 5.25V$ $V_I = 0.4V$		-1.6	mA

# mN640

## ELECTRICAL SPECIFICATIONS

### DC CHARACTERISTICS (CONT.)

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN.	MAX.	UNITS
$V_{OH}$	$A_O, B_O$	$V_{CC1} = 4.75V$ $V_{CC2} = 14V$ $I_O = 500 \mu A$	13		V
$V_{OL}$	$A_O, B_O$	$V_{CC1} = 4.75V$ $I_O = 25mA$	0	0.5	V
$I_{CC1H}$ + $I_{CC2H}$	SUPPLY CURRENT	$V_{CC1} = 5.5V$		9	mA
	SUPPLY CURRENT	$V_{CC2} = 16V$		250	$\mu A$
$I_{CC1L}$ + $I_{CC2L}$	SUPPLY CURRENT	$V_{CC1} = 5.5V$		60	mA
	SUPPLY CURRENT	$V_{CC2} = 16V$		42	mA
$C_I$				7	pF

NOTE: Positive current is into the pin.

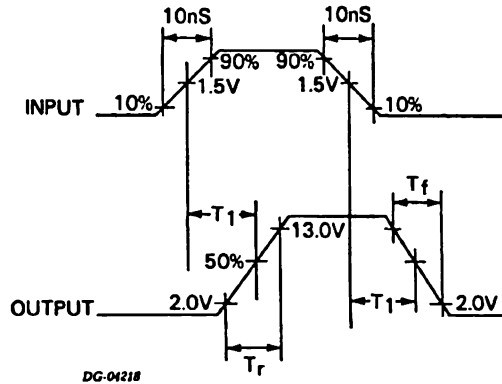


# mN640

## ELECTRICAL SPECIFICATIONS

### AC CHARACTERISTICS

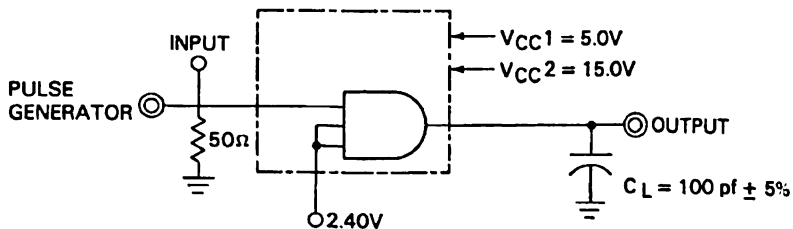
#### Switching Diagram



#### Transition Timing

SYMBOL	CHARACTERISTICS	CONDITIONS	MIN	MAX	UNITS
$T_r$	OUTPUT RISE TIME	$V_{CC1} = 5.0V$ $V_{CC2} = 15.0V$ TEST CIRCUIT	4	20	ns
$T_f$	OUTPUT FALL TIME	$V_{CC1} = 5.0V$ $V_{CC2} = 15.0V$ TEST CIRCUIT	4	20	ns
$T_1$	PROPAGATION DELAY	$V_{CC1} = 5.0V$ $V_{CC2} = 15.0V$ TEST CIRCUIT	15	35	ns

#### TEST CONDITIONS



1. Test each input separately
2. Input Pulse: Amplitude = 0 to 3V  
Width = 100nS (50% to 50%)  
 $t_r = t_f = 10nS$   
Freq. = 5MHz

NOTE:  $C_L$  includes jig and wiring capacitance.

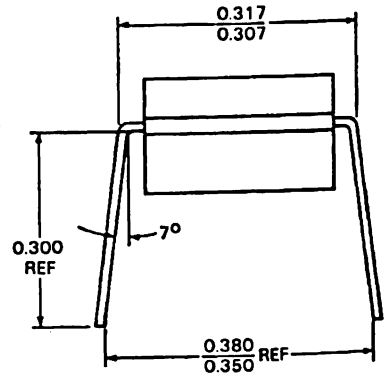
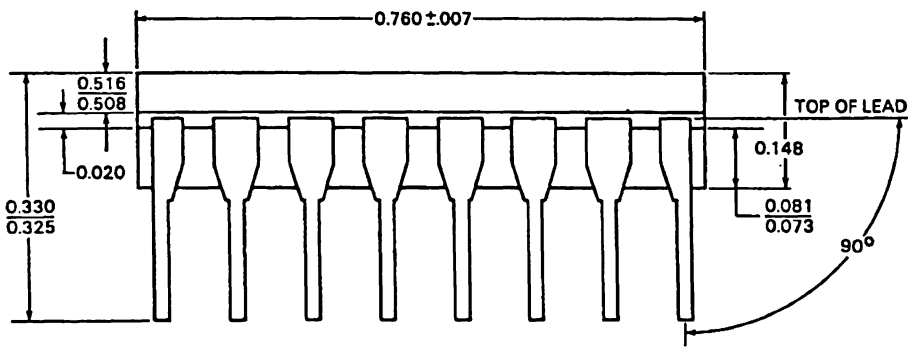
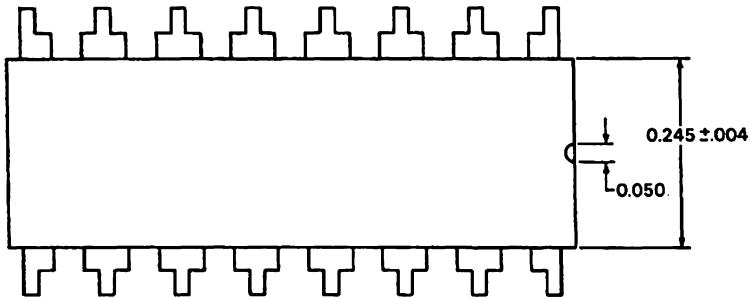
DG-04219



# mN640

## PACKAGE SPECIFICATION

### PACKAGE SPECIFICATIONS



DG-0211



This page intentionally left blank.

10

---

---

# DESIGN NOTES CONTENTS



# DESIGN NOTES

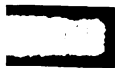
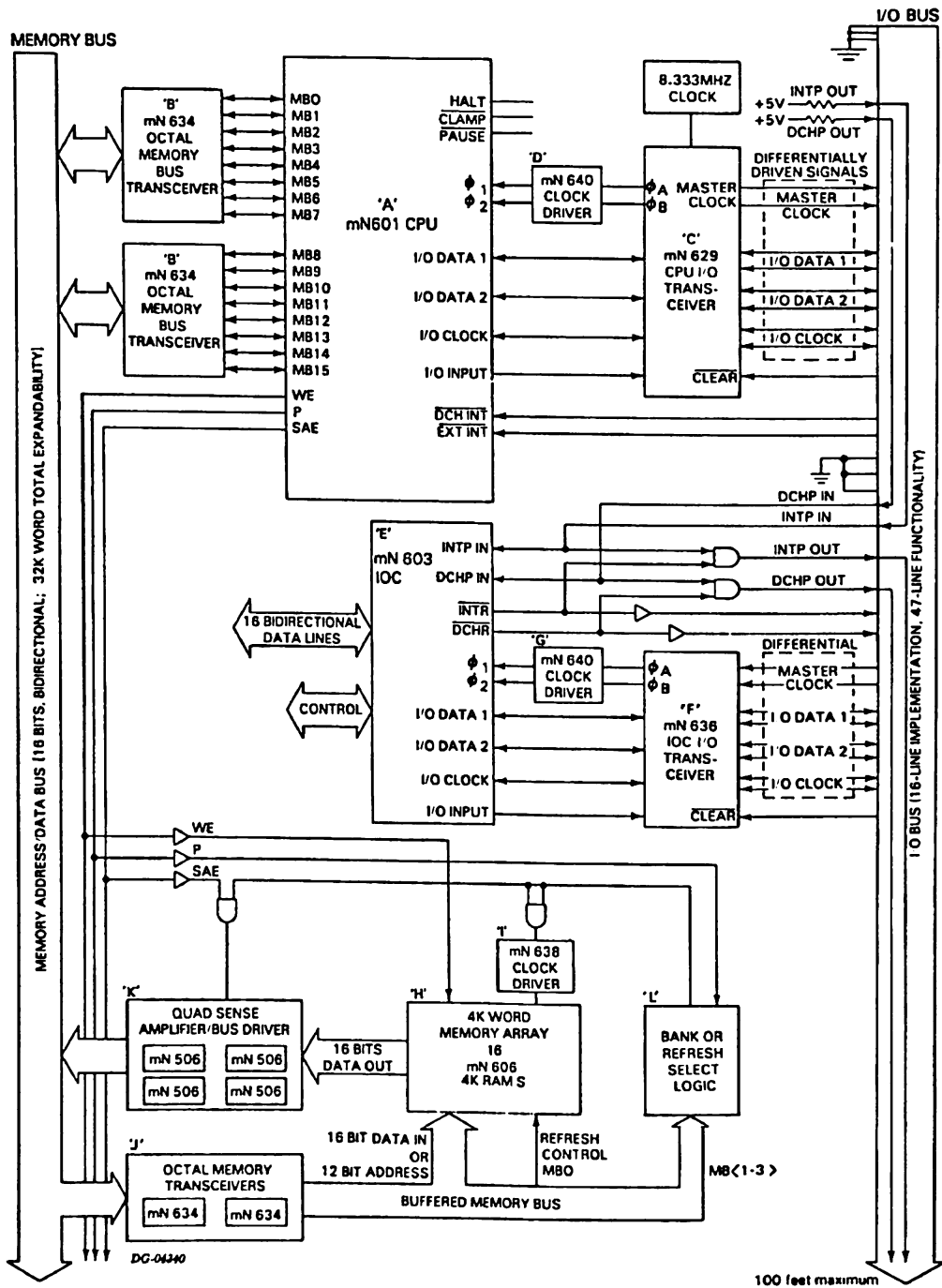
## CONTENTS

<b>1</b>	<b>INTRODUCTION TO DESIGN NOTES</b>
<b>2</b>	<b>SIMPLIFIED CPU MODULE</b>
<b>2</b>	<b>MEMORY BUS</b>
<b>4</b>	<b>mN601 SYSTEM CLOCK</b>
<b>4</b>	<b>CLOCK OVERSHOOT</b>
<b>4</b>	<b>mN601 GROUND PADS</b>
<b>4</b>	<b>mN601 DECOUPLING</b>
<b>5</b>	<b>GENERAL PURPOSE I/O INTERFACE</b>
<b>5</b>	<b>mN603 SYSTEM CLOCK</b>
<b>5</b>	<b>CLOCK OVERSHOOT</b>
<b>5</b>	<b>mN603 DECOUPLING</b>
<b>5</b>	<b>I/O BUS TERMINATION</b>
<b>8</b>	<b>mN606 RAM ARRAY DESIGN</b>
<b>8</b>	<b>BOARD CONSTRUCTION /VOLTAGE DISTRIBUTION</b>
<b>8</b>	<b>mN606 RAM DECOUPLING</b>
<b>8</b>	<b>mN638 CLOCK DRIVER DECOUPLING</b>
<b>10</b>	<b>mN506 SENSE AMP/BUS DRIVER DECOUPLING</b>
<b>10</b>	<b>MEMORY BUS RECEIVERS</b>
<b>10</b>	<b>PROM/RAM</b>
<b>12</b>	<b>ADDRESSING I/O DEVICES VIA THE MEMORY BUS</b>
<b>12</b>	<b>ADDRESSING</b>
<b>12</b>	<b>MEMORY</b>
<b>12</b>	<b>RAM DISABLE</b>
<b>14</b>	<b>mN601 AND mN603 NON-OVERLAPPING CLOCKS</b>
<b>15</b>	<b>POWER SUPPLY 12V TO 15V CONVERSION</b>



# DESIGN NOTES

## SYSTEM BLOCK DIAGRAM



# DESIGN NOTES

## INTRODUCTION

### INTRODUCTION TO DESIGN NOTES

This portion of the data manual contains engineering notes which are aids to the design engineer using the microNOVA chip set. The notes are compiled in sections, each devoted to a particular facet of design implementation. Each circuit shown has been thoroughly tested, and the text accompanying each schematic suggests proven methods for circuit layout. The six design notes are:

- Typical CPU module design
- General purpose I/O interface design
- mN606 RAM array design
- Using the memory bus to address I/O devices
- Generating the non-overlapping clocks required by the mN601 and mN603
- 12V to 15V power supply conversion

The design notes covering each of the above subjects describe conservative engineering practices suitable for a wide range of applications. The logic schematics show suggested circuitry, values of discrete components and, where applicable, manufacturer's part numbers for integrated circuits not included in Data General's microcomputer chip set.



# DESIGN NOTES

## SIMPLIFIED CPU MODULE

### SIMPLIFIED CPU MODULE

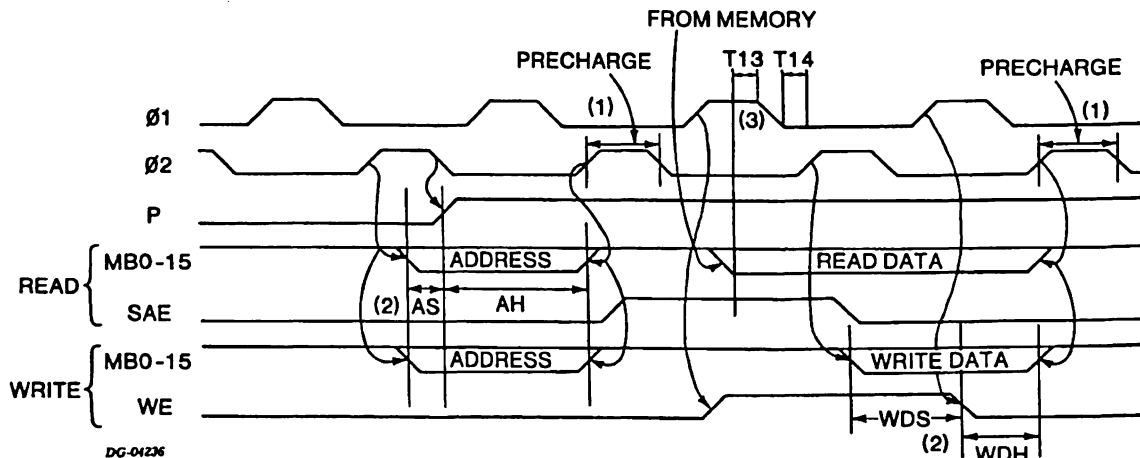
In a typical microNOVA CPU configuration, the mN601 microprocessor is interfaced to both the microNOVA I/O bus and memory bus as shown in the CPU logic schematic on the following page. When designing a CPU module, there are several practical considerations that should be taken into account; these are discussed below.

#### MEMORY BUS

The memory bus, MB <0-15>, is precharged internally by the CPU during every other  $\phi 2$  clock pulse, regardless of the operation being performed. During precharging, care should be exercised to ensure that memory bus lines are not driven low, thereby causing excess current on the lines.

After the memory bus lines are precharged, data is placed on the bus by discharging those lines which carry 0's and leaving charged those lines which carry 1's. As shown in the diagram below, the memory bus is precharged once during either a read or write cycle (and at least twice during a read-modify-write cycle - see mN601 circuit description, Memory Operations). Thus, discharging the memory bus before information is either read by the CPU during a read operation or read by memory during a write operation can result in data loss. Similarly, discharging the memory bus before the memory address is read at address time (when the memory control signal P is asserted) can result in incorrect memory addressing. The interline capacitance in the bus should be limited to 10pf to eliminate crosstalk.

#### MEMORY BUS PRECHARGING



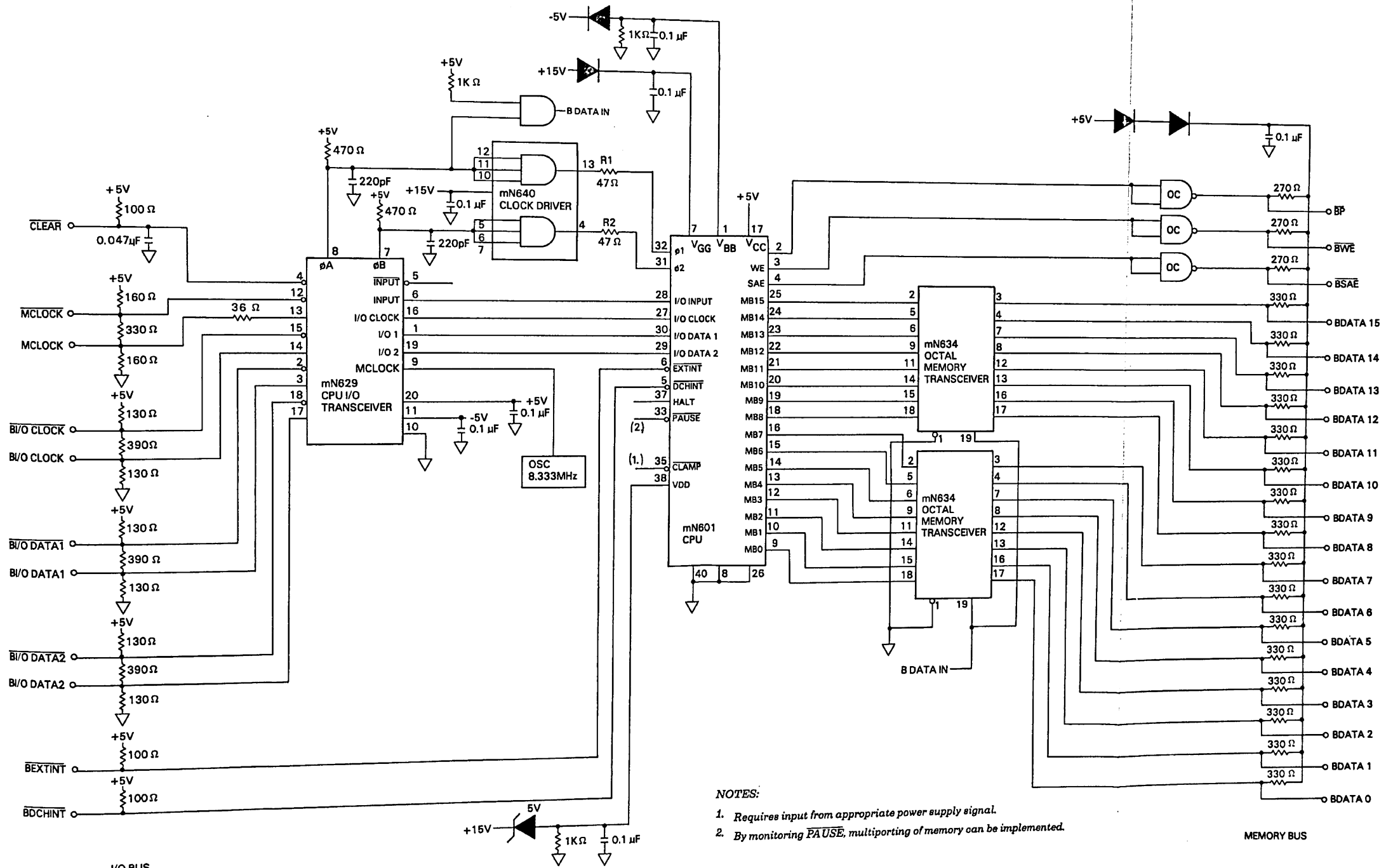
#### NOTES:

1. During precharge time, MB lines should not be driven low because high current spikes may cause damage to the mN601.
2. During memory address or data set-up time, the MB lines should not be driven low, since incorrect addressing or data loss will result. See mN601, Memory Timing Table.
3. In order to read information from the memory, MB lines carrying 0's must be discharged by T13, before the falling edge of  $\phi 1$  (see mN601, Switching Diagrams). Once discharged, the MB lines remain discharged until the next precharge.

# DESIGN NOTES

## TYPICAL microNOVA CPU MODULE

TYPICAL microNOVA CPU MODULE



- NOTES:
1. Requires input from appropriate power supply signal.
  2. By monitoring PAUSE, multiporting of memory can be implemented.



# DESIGN NOTES

## SIMPLIFIED CPU MODULE

### mN601 SYSTEM CLOCK

The non-overlapping clocks required to generate the microprocessor's internal four-phase clock are described in the Non-overlapping Clocks note. This circuitry is also shown in the CPU logic schematic.

### CLOCK OVERSHOOT

Care should be exercised in minimizing inductance on the clock lines. An overshoot on the clock waveform caused by inductance in the connections between the mN640 clock driver and the mN601 will not create a problem providing its amplitude is limited to 1V or less. This overshoot is dependent upon the clock driver rise and fall times. The value of the overshoot can be controlled by changing the values of the resistors, R1 and R2 (see schematic), in series with the outputs of the mN640 clock driver.

### mN601 GROUND PADS

Noise voltage developed between the three grounds pads (VSS, pins 8, 26, and 40) as a result of current in the etch will result in reduced CPU operating margins. For this reason, it is recommended that these pads be interconnected with short and heavy, high conductivity connections.

### mN601 DECOUPLING

It is recommended that decoupling of the mN601's supply voltage connections ( $V_{BB}$ ,  $V_{CC}$ ,  $V_{DD}$  and  $V_{GG}$ ) be made as close as possible to the mN601 chip and the nearest ground connected to it.

Suggested values for decoupling capacitors are:

Supply Connection	Capacitor Value
$V_{BB}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{CC}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{DD}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{GG}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent

Overall board decoupling at low frequency should be incorporated according to standard engineering practices.



# DESIGN NOTES

## GENERAL PURPOSE I/O INTERFACE

### GENERAL PURPOSE I/O INTERFACE

The logic schematic on the following page illustrates a typical general purpose microNOVA I/O interface. In the drawing, the mN603 I/O controller is interfaced to the I/O bus via an mN636 IOC I/O transceiver and to the device via the 16-bit data bus ( $\langle 0-15 \rangle$  H or D  $\langle 0-15 \rangle$  L) and a 4-to-16 line decoder.

#### mN603 SYSTEM CLOCK

The non-overlapping clocks required to generate the mN603's internal four-phase clock are described in the Non-overlapping Clocks note. This circuitry is also shown in the microNOVA I/O interface schematic.

#### CLOCK OVERSHOOT

Care should be exercised in minimizing inductance on the clock lines. An overshoot on the clock waveform caused by inductance in the connections between the mN640 clock driver and the mN603 will not create a problem providing its amplitude is limited to 1V or less. This overshoot is dependent upon the clock driver rise and fall times. The value of the overshoot can be controlled by changing the values of the resistors, R1 and R2 (see schematic), in series with the outputs of the mN640 clock driver.

#### mN603 DECOUPLING

It is recommended that decoupling of the mN603's supply voltage connections ( $V_{BB}$ ,  $V_{CC}$ ,  $V_{DD}$  and  $V_{GG}$ ) be made as close as possible to the mN603 chip and the nearest ground connection.

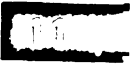
Suggested values for the decoupling capacitors are:

Supply Connection	Capacitor Value
$V_{BB}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{CC}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{DD}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent
$V_{GG}$	0.1 $\mu$ f EIA # Z5V dielectric, or equivalent

Overall board decoupling at low frequency should be incorporated according to standard engineering practices.

#### I/O BUS TERMINATION

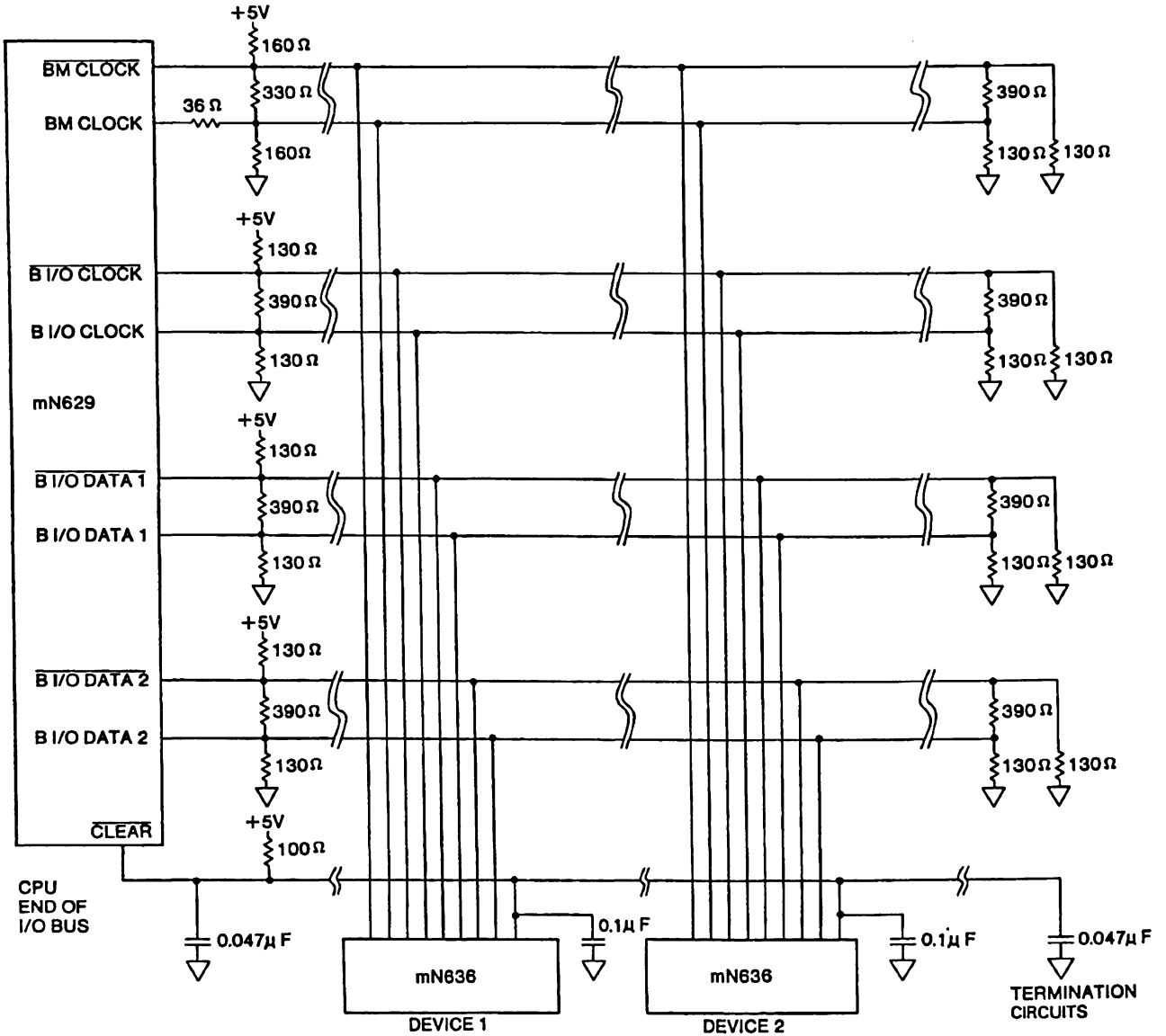
A suggested method of terminating the differentially driven I/O bus signals is shown on the following page. Non-differential I/O bus signals do not require termination; however, the CLEAR line should contain a 0.1 $\mu$ f capacitor into each device and a 0.047 $\mu$ f capacitor in the termination circuits.



# DESIGN NOTES

## GENERAL PURPOSE I/O INTERFACE

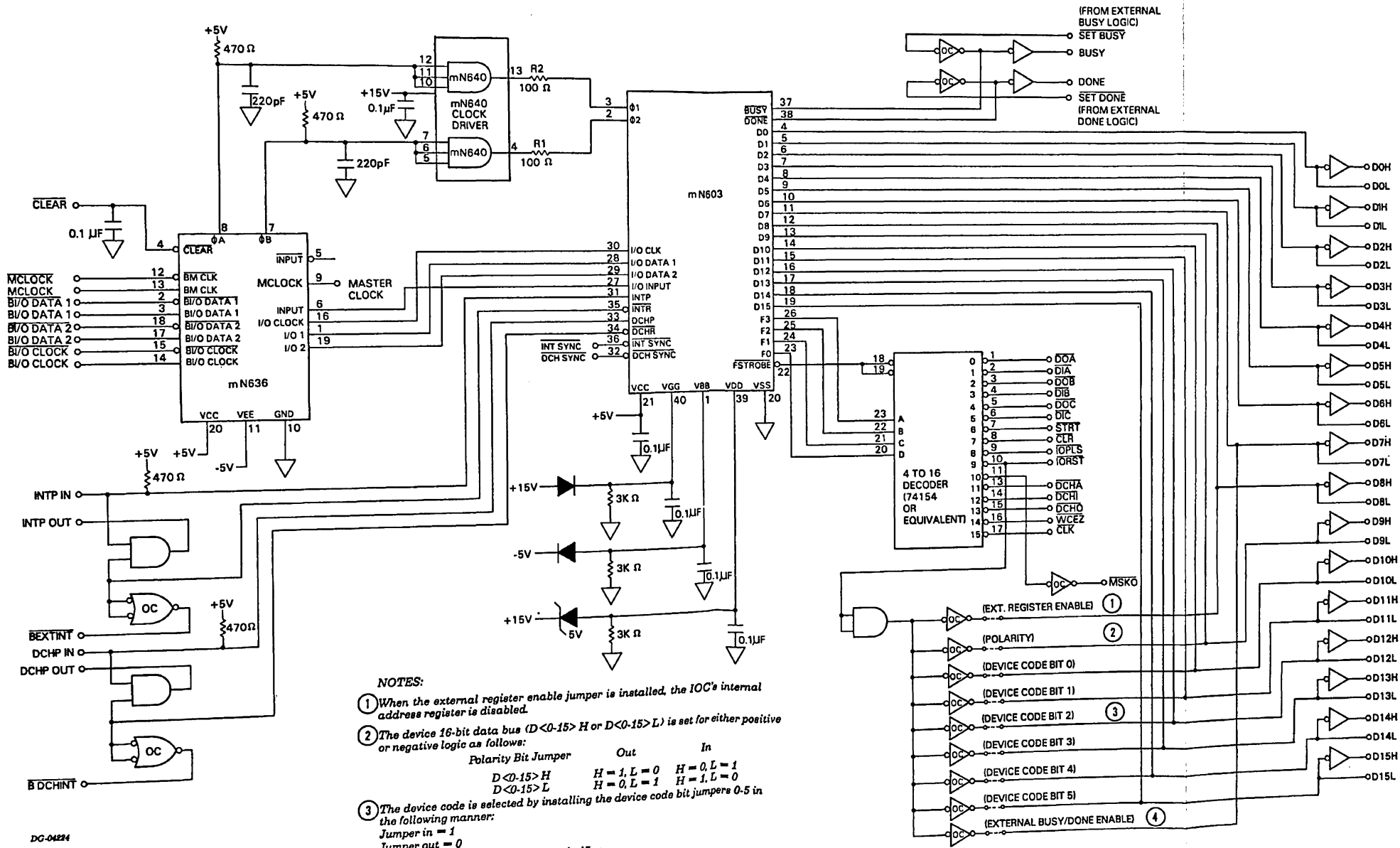
### I/O BUS TERMINATION



**NOTE:**  
 The values of the resistors shown in the diagram are applicable when Scotchflex cable, part No. 3365/16, or equivalent, is used. With other cables, the values of the resistors may vary; but, the current driving capabilities of the transceivers must not be exceeded.

DG-04274

### GENERAL PURPOSE INTERFACE



- NOTES:**
- When the external register enable jumper is installed, the IOC's internal address register is disabled.
  - The device 16-bit data bus ( $D<0-15>H$  or  $D<0-15>L$ ) is set for either positive or negative logic as follows:  

Polarity Bit Jumper	Out	In
$D<0-15>H$	H = 1, L = 0	H = 0, L = 1
$D<0-15>L$	H = 0, L = 1	H = 1, L = 0
  - The device code is selected by installing the device code bit jumpers 0-5 in the following manner:  
 Jumper in = 1  
 Jumper out = 0  
 Example of jumpering for device code 15g:  

Device code bit jumpers	0	1	2	3	4	5
Jumper in/out	0	0	1	1	0	1
	1		5		= 15g	

- When the Busy/Done jumper is installed, external Busy and Done flags are enabled.

# DESIGN NOTES

## mN606 RAM ARRAY DESIGN

### mN606 RAM ARRAY DESIGN

The following notes discuss basic concepts which should be considered when designing a printed circuit board suitable for mN606 MOS RAMs.

#### BOARD CONSTRUCTION/VOLTAGE DISTRIBUTION

Two-layer printed circuit board construction provides an effective method of distributing voltages and ground on a high-speed memory card such as a dynamic RAM board. On a two-layer board, the normal method is to grid or mesh ground and power distribution lines. While these grids may be irregular, each grid should be a rectangle, approximately 1 by 2 inches. The grid distribution is constructed by connecting the horizontal voltage or ground traces on one side of the board with their vertical counterparts on the opposite side of the board (see the printed circuit board diagram). This type of construction reduces signal return distances and equalizes current flow along the buses. Equalizing current flow reduces supply voltage drops, thereby reducing voltage and ground variations between memory devices.

#### mN606 RAM DECOUPLING

It is necessary to decouple the mN606 RAM supply connections to prevent power spikes which will force the RAM to operate out of specification. MOS memories exhibit small standby currents but, when accessed, draw relatively large, short duration transient currents from the power supply. These high transient currents can be reduced by placing decoupling capacitors a short distance from the memory device. The decoupling capacitors average the current pulse over the interval of 'on time' to 'off time', essentially decreasing the frequency of the current components presented to the main power distribution lines and supplies.

Two types of capacitors are normally used for decoupling each voltage. The first type provides the high frequency pulse currents required by the RAMs. The second type provides storage for the first type and further reduces the magnitude of the low frequency currents that are presented to the power supply.

The first type (designated as C1 and C2 on the diagram) is usually ceramic and is located close to the mN606. These capacitors must have sufficient capacity and a frequency response that is high enough to supply the pulse currents to the RAMs without discharging more than an acceptable amount. The capacitance is determined by both the voltage drop that can be tolerated at the mN606 and the peak transient current. The leading and trailing edge current spikes contain high frequency components which dictate the frequency response characteristics of the capacitor. These high frequency response capacitors should be located adjacent to each mN606; or, if the array is designed with the RAMs close together, they can be located near every other chip.

For a 64 memory device array (16K of memory), a typical value for  $V_{DD}$  decoupling (C1) is  $0.05\mu\text{f}$  per 4K RAM, or approximately thirty-two  $0.1\mu\text{f}$  capacitors distributed between alternate chips. Similarly, the typical value for  $V_{BB}$  decoupling (C2) is  $0.05\mu\text{f}$  per mN606, or approximately thirty-two  $0.1\mu\text{f}$  capacitors distributed between alternate chips.

The second type (designated as C3 and C4 on the diagram) is usually a tubular tantalum capacitor which is used to integrate the cycle current requirements. The value of capacitors C3 and C4 is related to the difference between the cycle current and the standby current. For a 64 memory device array, the typical total value for C3 ( $V_{DD}$  decoupling) is  $100\mu\text{f}$  divided between two or three capacitors located near the outer or connector edge of the array. Similarly, the typical value for C4 ( $V_{BB}$  decoupling) is  $5\mu\text{f}$ .

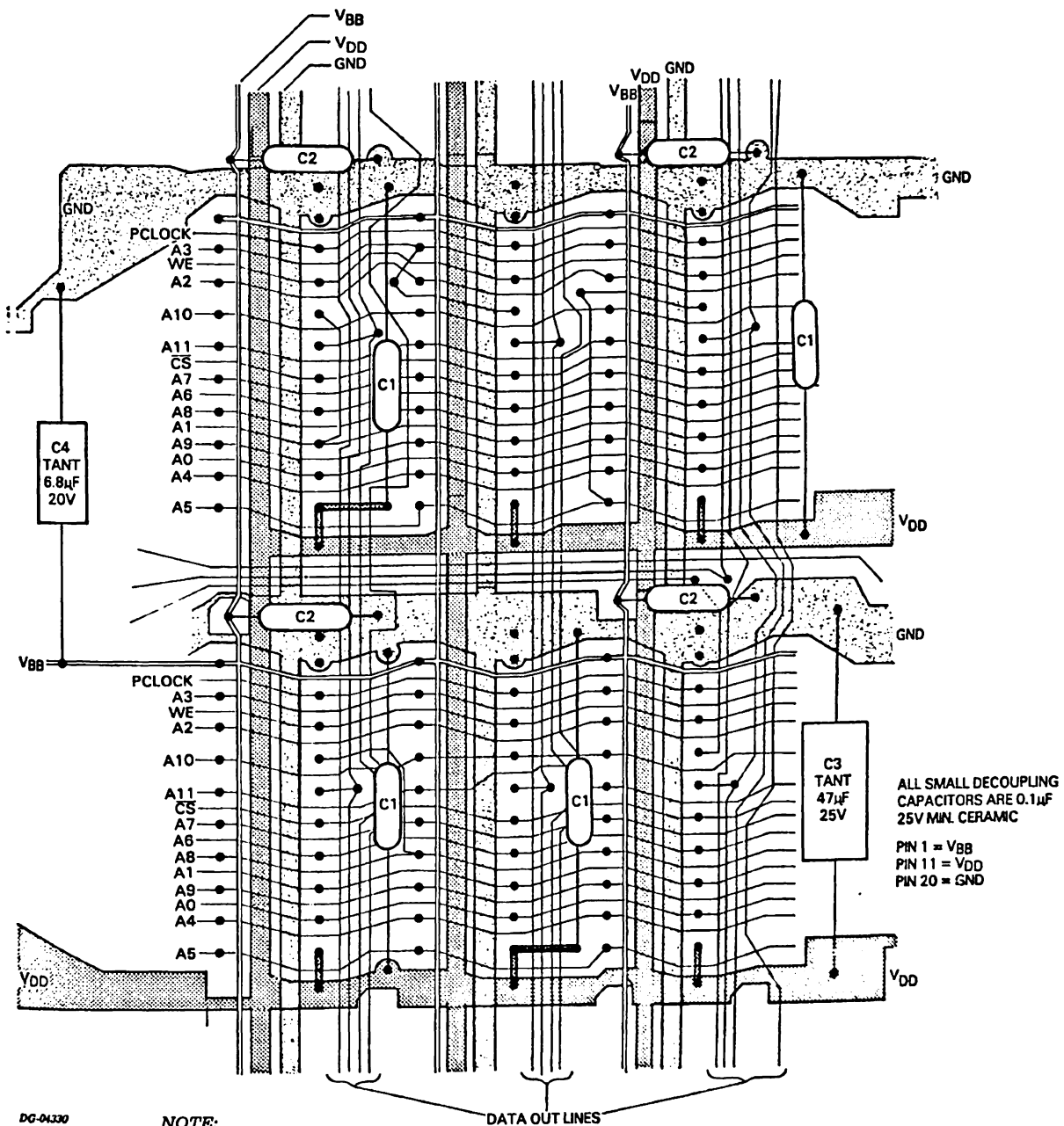
#### mN638 CLOCK DRIVER DECOUPLING

Decoupling the  $V_{CC2}$  (+15V) connection of each mN638 memory clock driver is recommended because of the relatively high driver peak currents encountered.  $V_{CC2}$  should be decoupled by both a  $0.1\mu\text{f}$  and a  $6.8\mu\text{f}$  capacitor. The  $0.1\mu\text{f}$  capacitor should be located as close to each mN638 chip as possible while one  $6.8\mu\text{f}$  capacitor can serve up to four mN638 chips.

Both the  $V_{CC3}$  and  $V_{CC1}$  connections of the clock driver should also be decoupled. In this case, a  $0.1\mu\text{f}$  capacitor should be used to decouple each connection and the two capacitors should be located as close to the chip as possible.

# DESIGN NOTES

## mN606 RAM ARRAY DESIGN



DG-04330

**NOTE:**

In a 4K memory array, the same bus is used for both address and data inputs. Thus, in most cases, the data input line of each chip is tied to one of the address lines, and one address line can serve only one chip; i.e., one chip per data bit. Since the mN606 has only 12 address lines, some chips receive their data input from a memory bus line that is not otherwise connected to the chip (see diagram). The data outputs of each chip are independent. They are fed as inputs to sense amp/bus drivers (see 4K memory schematic).

In an 8K memory array, the data inputs to the same bits can be connected in parallel. Similarly, the data outputs of the same bits can also be connected in parallel.

# DESIGN NOTES

## mN606 RAM ARRAY DESIGN

### mN506 SENSE AMP/BUS DRIVER DECOUPLING

It is recommended that the  $V_{CC1}$ ,  $V_{CC2}$  and  $V_{EE}$  connections of each mN506 chip be decoupled using a 0.1 $\mu$ f capacitor for each connection. These capacitors should be located as close to the mN506 chip as possible. One capacitor can serve the  $V_{CC1}$  connection of two chips, one capacitor can serve the  $V_{CC2}$  connection of two chips, and one capacitor can serve the  $V_{EE}$  connection of four chips.

### MEMORY BUS RECEIVERS

mN634 memory bus drivers can be used successfully to drive the mN606 memory array. Since the mN634 is an open-collector output device, it requires pull-up resistors to supply the high input requirement (+4V) of the RAM. The value of the pull-up resistors is computed from the timing requirements of the CPU. The memory bus is precharged by the CPU prior to the time the input lines are sampled by the RAMs. Thus, there is a period of approximately 240ns during which the input lines can be brought to a high of 4V. In the schematic of a typical 4K memory shown on the following page, the pull-up resistors have a value of 510 ohms.

Because of the inductance of the RAM array, there may be an inductive undershoot in the input lines when they are pulled low by the mN634. This can drive the low input below the recommended specifications of the mN606 RAM. To reduce the undershoot pulse, a resistor in series with the input line can be used. While the value of these resistors (one per input line) is layout dependent, the computed value of the series resistors in the typical 4K memory schematic is 33 ohms.

Devices other than mN634's can be used to drive the input lines of the RAM array. When TTL drivers with totem pole outputs are used, a pull-up resistor, as described above, is required to ensure that the input threshold requirement of the RAM (+4V) is satisfied. Schottky gates with totem pole outputs are not recommended.

### PROM/RAM

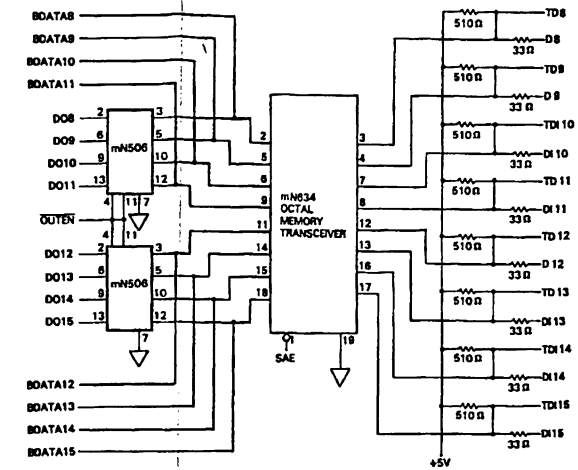
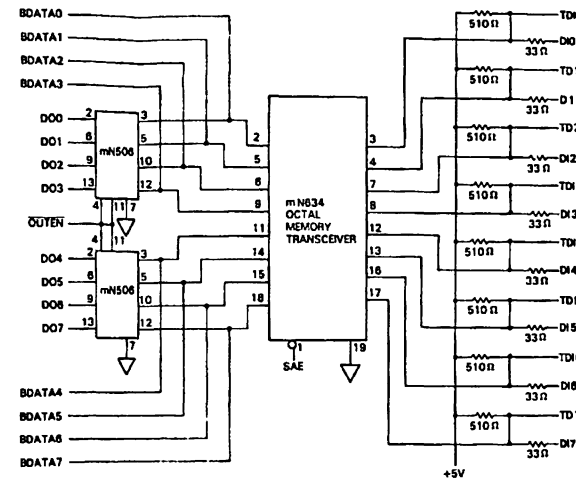
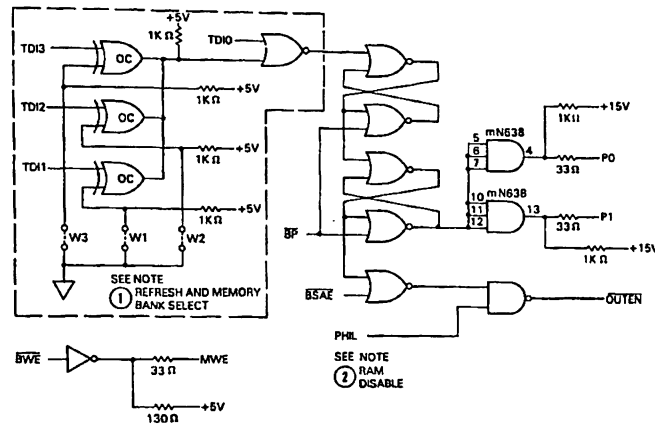
When both PROM and RAM elements are included in a microNOVA computer system, the PROM is given priority over the RAM. This is accomplished using the PHIL signal, as shown in the logic schematic on the following page.



# DESIGN NOTES

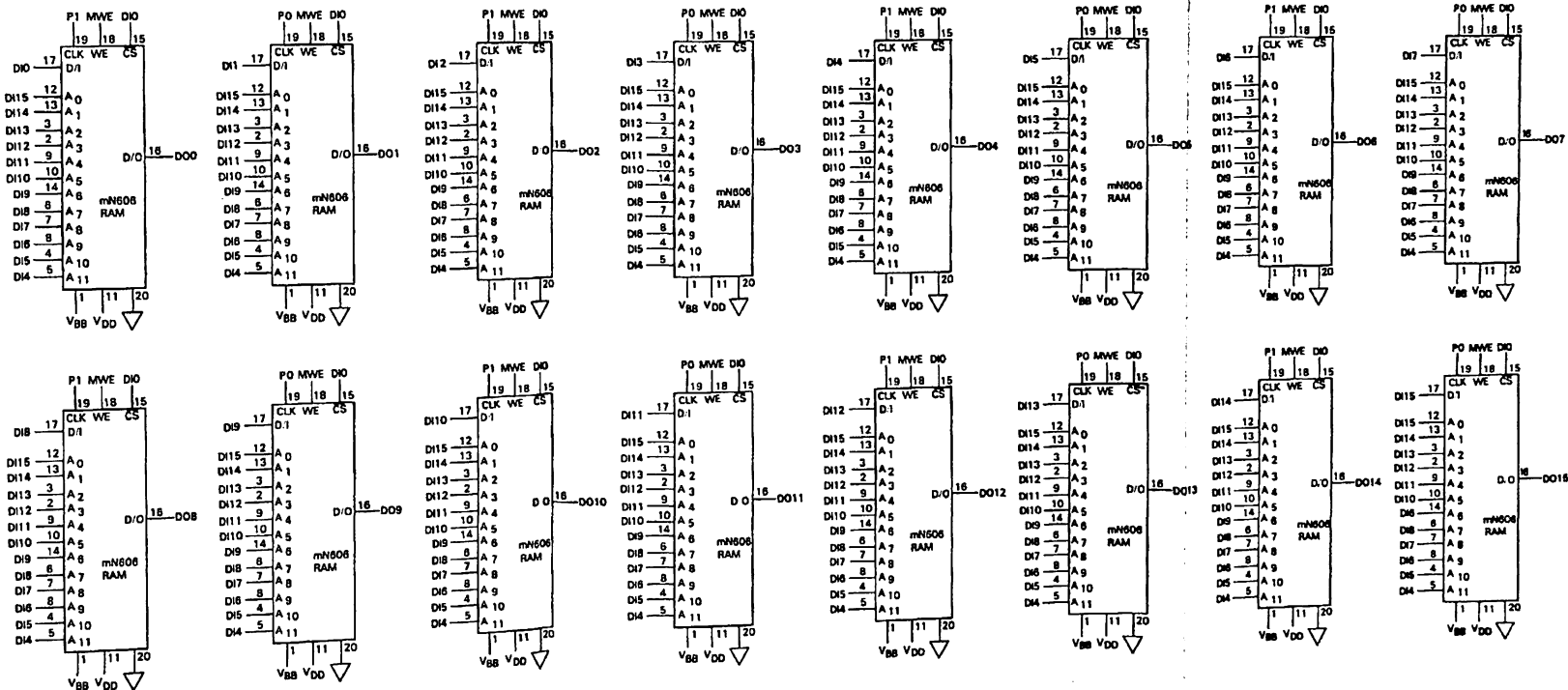
## 4K RAM MODULE

### 4K RAM MODULE



#### NOTES:

- 1 The bank select logic consists of three open-collector EXCLUSIVE-OR gates, a pull-up resistor to +5Vdc, and three jumpers, designated W1, W2, and W3. The 4Kx16-bit RAM memory array is selected when both the value of the address lines (TD11-TD13) matches the value selected by the jumper settings and TD10 is low. When this occurs, the output of the EXCLUSIVE-OR gates goes to a high logic level, and the RAM array is enabled for either read, write, or read-modify-write operations when the memory control signal BP is asserted. When a refresh cycle is initiated, TD10 is high, thereby selecting the memory, independent of TD11-TD13, when BP is asserted. Also, D10 (input to CS pin of RAMs) goes high so that a refresh operation will be performed by the mN606's.
- 2 When a PROM element is addressed, the PHIL signal is pulled low and OUTEN cannot go low. When this occurs the RAM outputs are disconnected from the memory bus.



DO-04227



# DESIGN NOTES

## ADDRESSING I/O DEVICES VIA THE MEMORY BUS

### ADDRESSING I/O DEVICES VIA THE MEMORY BUS

In microNOVA systems it is possible to make an active register behave as a memory register, thus allowing I/O interfaces to be accessed via the microNOVA memory bus.

A suggested method of implementing this technique is shown in the following logic schematic. It is assumed in the drawing that the memory bus is driven as depicted in the CPU logic schematic contained in the CPU Notes.

#### ADDRESSING

The memory register can be addressed by using jumpers on the memory input lines (MEM <0-15> , see schematic). When all the input lines to the address select NAND gates are at a high logic level, the memory register is enabled for either reading or writing.

#### MEMORY REGISTER

In the drawing, the register is comprised of sixteen edge-triggered flip-flops with independent clear and reset lines. The outputs of the register are connected to the memory output lines (MEM <0-15> ) via tri-state gates that are enabled at memory read time.

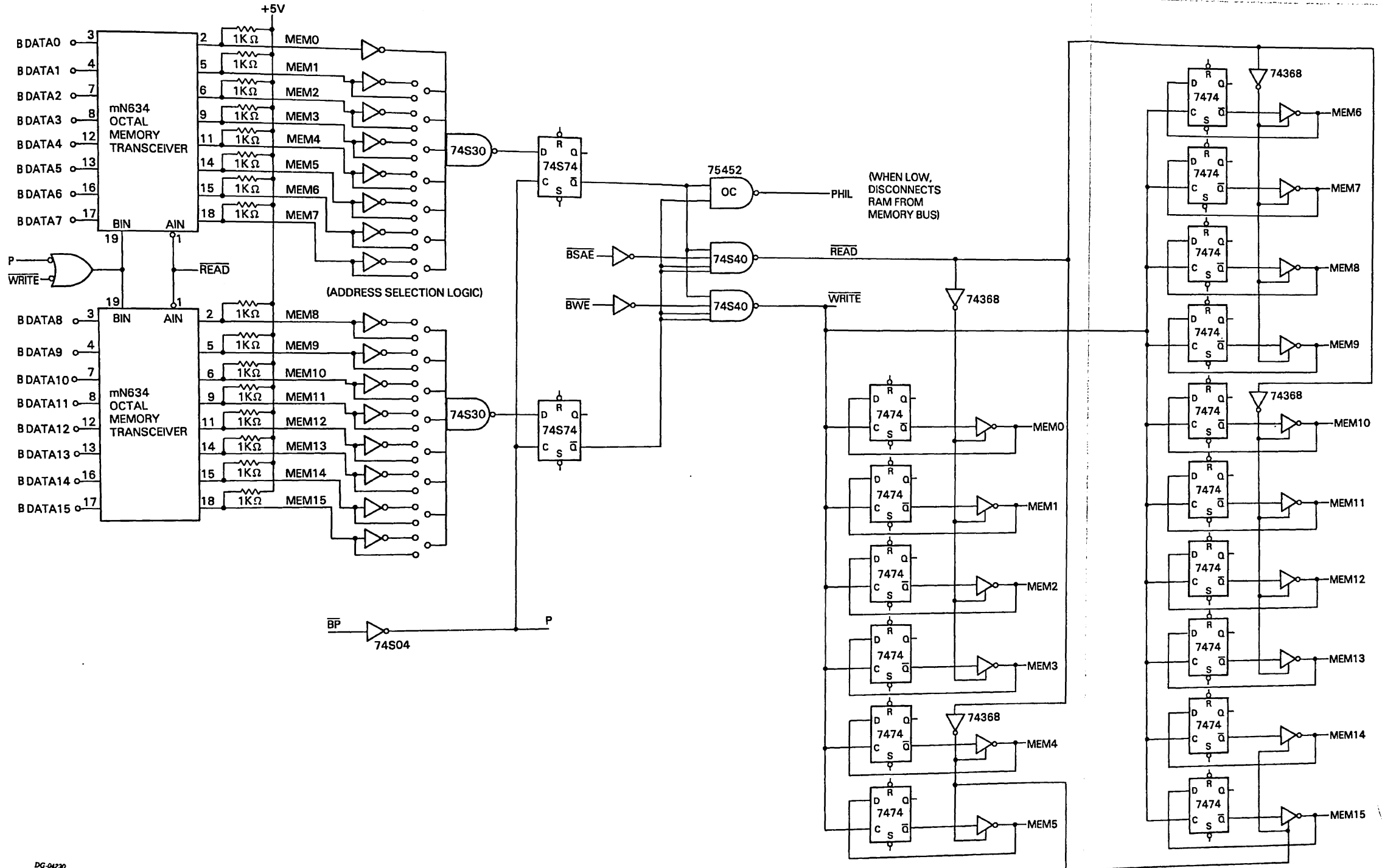
#### RAM DISABLE

The PHIL signal (see schematic) is used to disable any RAMs in the system which respond to the same address as that selected for the memory register. When this condition does not exist, the PHIL circuitry has no function.



# DESIGN NOTES

## ADDRESSING I/O DEVICES VIA THE MEMORY BUS



# DESIGN NOTES

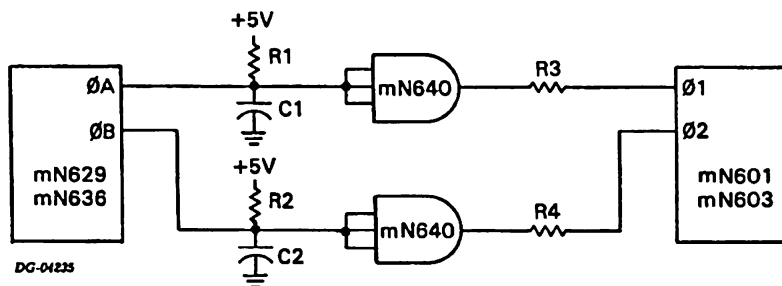
## mN601 AND mN603 NON-OVERLAPPING CLOCKS

### mN601 AND mN603 NON-OVERLAPPING CLOCKS

The mN601 microprocessor and the mN603 I/O controller require two non-overlapping clock signals to generate their internal four-phase system clock. The logic schematic below illustrates a suggested method of producing these MOS clock inputs.

As shown, external circuitry is used to shape the clock signals generated by the appropriate transceivers before they are supplied to the mN601 or mN603 via the mN640 clock driver (shown as two AND gates).

#### EXTERNAL CLOCK SHAPING



#### NOTES:

1. The recommended values for the discrete components shown in the diagram are:

Reference Designators	Values
R1, R2	470ohm, 5%
C1, C2	Typical value is 220pf, silver mica, +5%. This value can vary depending upon circuit layout.
R3, R4	Typical value for the mN601 is 47ohm; and for the mN603, 100ohm. These values can vary when required to reduce overshoot in the clock signals.

2. Ground connections between C1, C2 and the mN640 clock driver should be as short and heavy as possible.
3. It is recommended that decoupling of the mN640 clock driver supply voltage connections ( $V_{CC1}$  and  $V_{CC2}$ ) be made as close to the mN640 chip as possible.

# DESIGN NOTES

## POWER SUPPLY 12V TO 15V CONVERSION

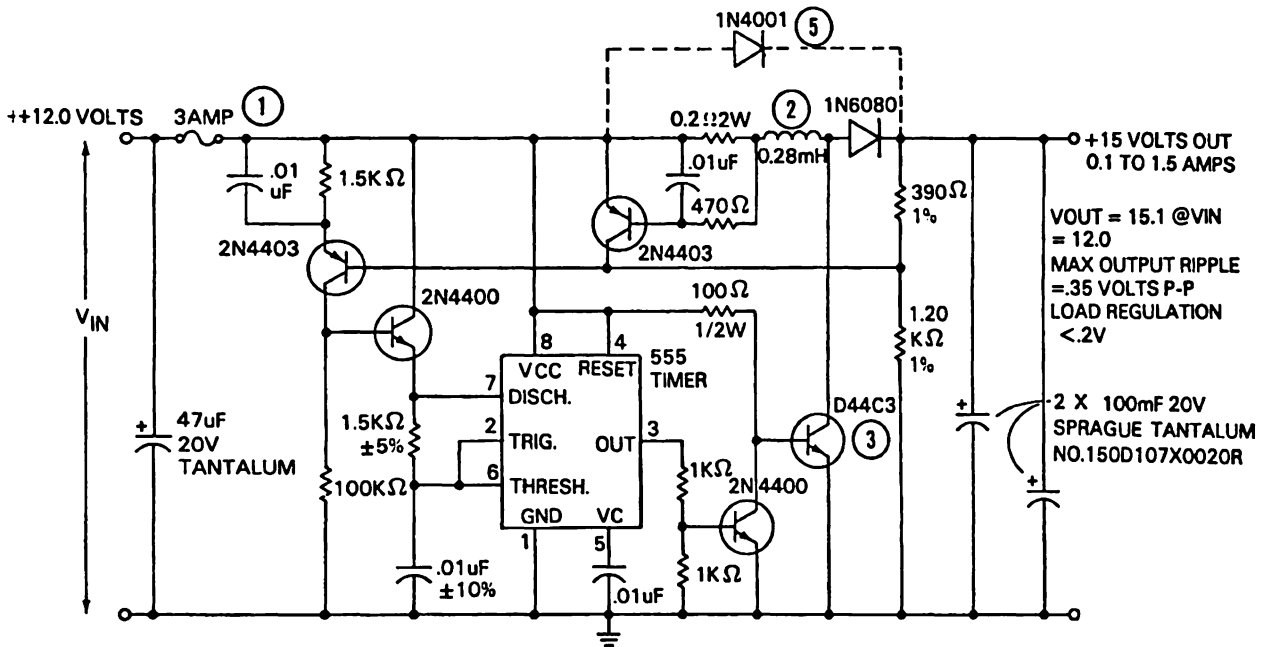
### POWER SUPPLY 12V TO 15V CONVERSION

To accommodate systems running with a 12V power supply, logic schematics for two 12V to 15V converters are shown. These converters provide the 15V supply required to operate Data General's microNOVA integrated circuits.

The first voltage converter schematic produces a supply voltage of 15V at 1.5amps. This voltage converter supports a full microNOVA chip set, including 32K of memory.

The second schematic produces a supply voltage of 15.2V at 0.15amps. This will support either an mN601 or mN603 connection.

#### +12 TO +15 VOLT CONVERTER FOR microNOVA CHIP SET



DG-04238

#### NOTES:

- ① Fuse or +12volt current limit needed to avoid overheating 0.2Ω resistor and 1N6080 diode with permanent output short.
- ② 0.28mH Inductor: 41 turns NO.20 wire on Arnold Magnetics toroid core NO.A-201163.
- ③ D44C3 transistor on time = 10μSEC. Off time varies with load. Max frequency is 25KHz.
- ④ Max component height 0.34" above board.
- ⑤ 1N4001 diode may be necessary to prevent overshoot of output if V<sub>IN</sub> turn on time is fast.
- ⑥ Efficiency: 80%, typical.



This page intentionally left blank.



# APPENDICES CONTENTS

122

# APPENDICES

## CONTENTS

<b>A-1</b>	<b>APPENDIX A</b>
<b>A-1</b>	<b>I/O BUS OPERATIONS</b>
<b>B-1</b>	<b>APPENDIX B</b>
<b>B-1</b>	<b>LOGIC CONVENTIONS</b>
<b>B-1</b>	<b>SIGNAL NAMES</b>

12

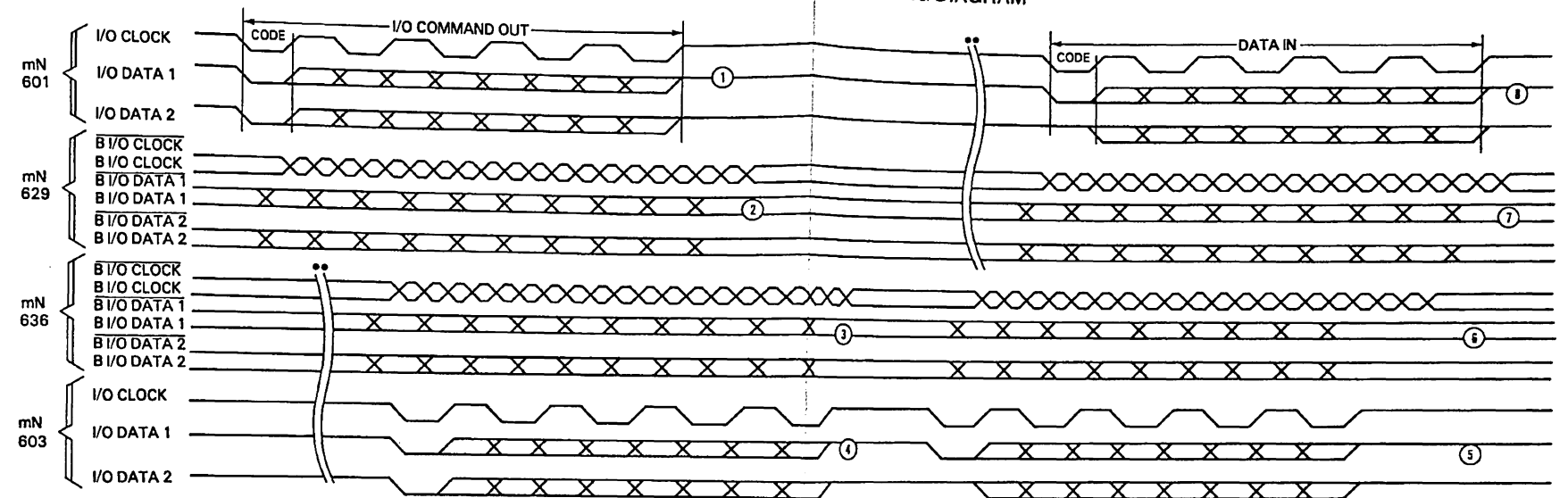


**This page intentionally left blank.**

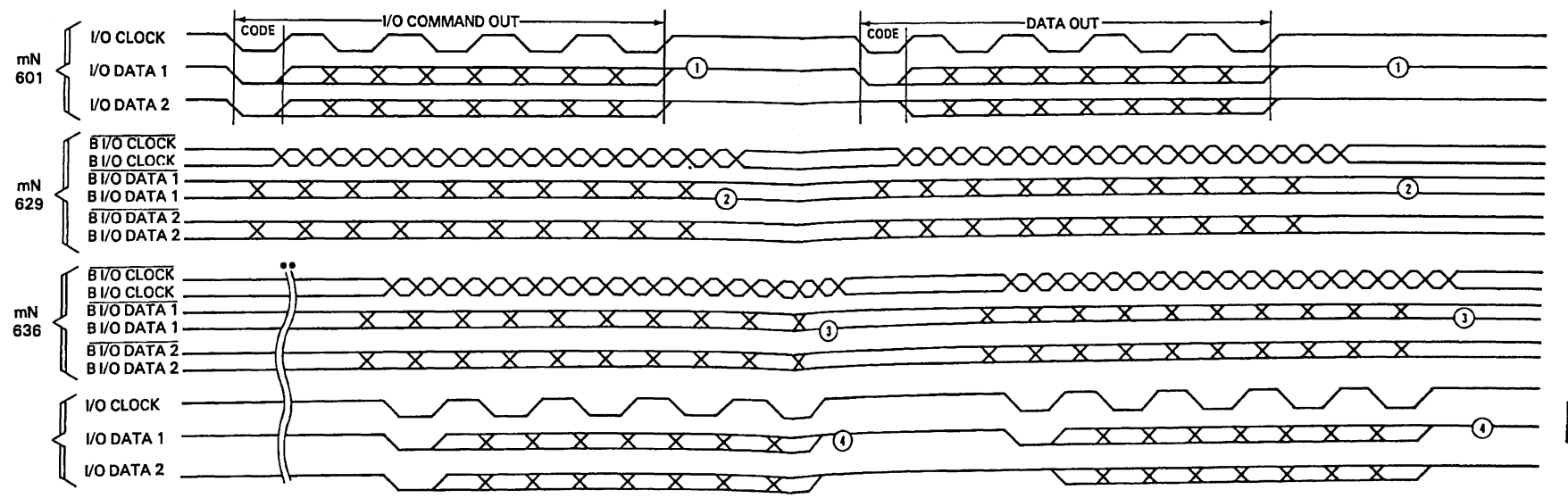
# APPENDIX A I/O BUS OPERATIONS

- NOTES:
- ① I/O COMMAND OR DATA SENT FROM mN601 MICROPROCESSOR TO THE CPU I/O TRANSCEIVER.
  - ② CPU I/O COMMAND OR DATA DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE CPU I/O TRANSCEIVER.
  - ③ DIFFERENTIALLY DRIVEN I/O COMMAND OR DATA RETRIEVED FROM THE I/O BUS BY THE IOC I/O TRANSCEIVER.
  - ④ CPU I/O COMMAND OR DATA RECEIVED BY THE mN603 I/O CONTROLLER.
  - ⑤ DATA SENT FROM THE mN603 TO THE IOC I/O TRANSCEIVER.
  - ⑥ IOC DATA DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE IOC I/O TRANSCEIVER.
  - ⑦ DIFFERENTIALLY DRIVEN DATA RETRIEVED FROM THE I/O BUS BY THE CPU I/O TRANSCEIVER.
  - ⑧ IOC DATA RECEIVED BY THE mN601.

PROGRAMMED I/O DATA IN TIMING DIAGRAM



PROGRAMMED I/O DATA OUT TIMING DIAGRAM



NOTE:  
SEE mN601 I/O OPERATIONS AND I/O TIMING TABLE FOR DETAILED INFORMATION.  
•• TIME IS DEPENDENT UPON THE LENGTH OF THE I/O BUS BETWEEN THE mN601 AND THE mN603.  
DG-04232

This page intentionally left blank.



NOTES :

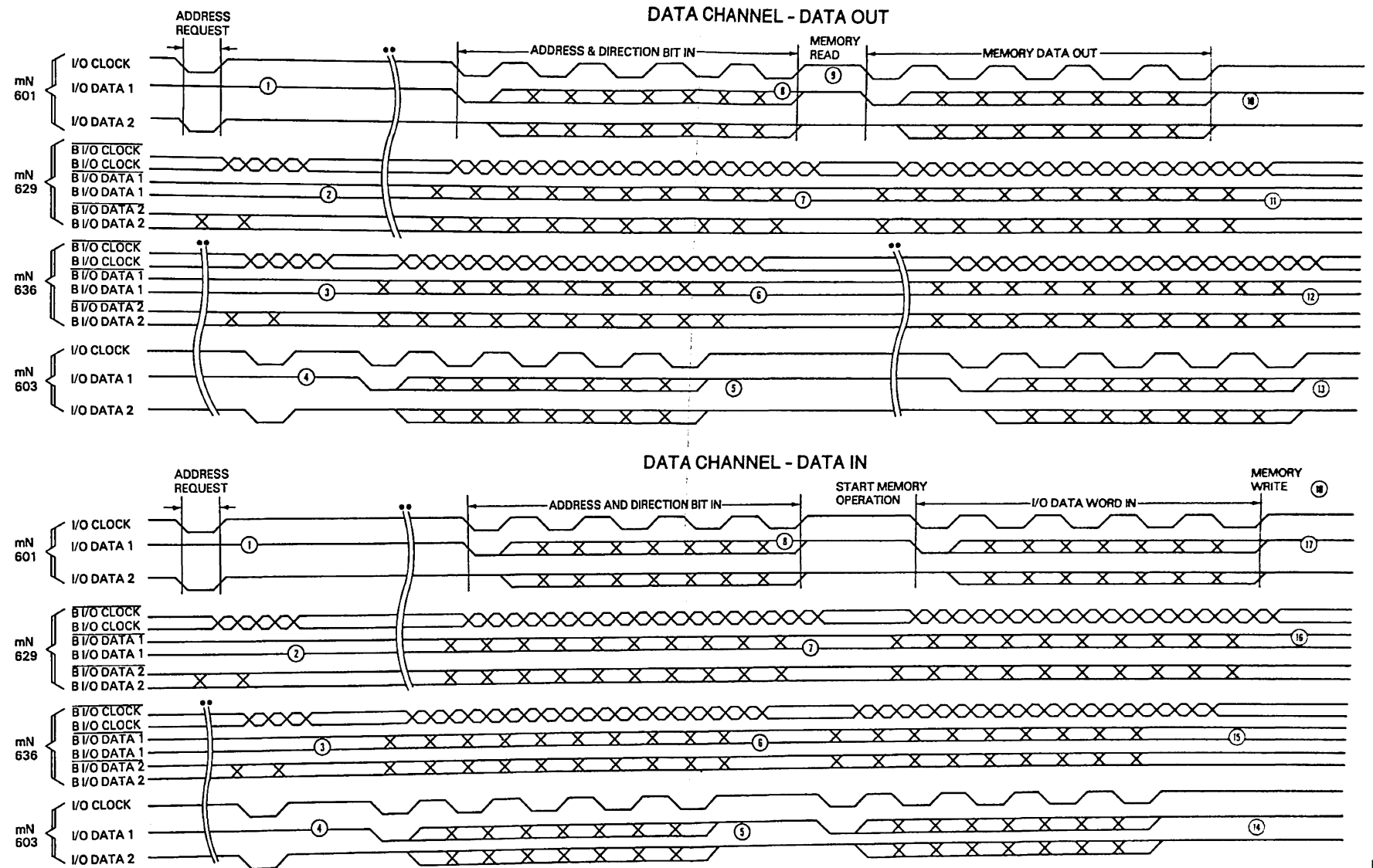
- ① DATA CHANNEL ADDRESS REQUEST SENT FROM THE mN601 TO THE CPU I/O TRANSCEIVER.
- ② DATA CHANNEL ADDRESS REQUEST DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE CPU I/O TRANSCEIVER.
- ③ DIFFERENTIALLY DRIVEN DATA CHANNEL ADDRESS REQUEST RETRIEVED FROM THE I/O BUS BY THE IOC I/O TRANSCEIVER.
- ④ DATA CHANNEL ADDRESS REQUEST RECEIVED BY THE mN603.
- ⑤ MEMORY ADDRESS AND DIRECTION MODE BIT SENT FROM THE mN603 TO THE IOC I/O TRANSCEIVER.
- ⑥ MEMORY ADDRESS AND DIRECTION MODE BIT DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE IOC I/O TRANSCEIVER.
- ⑦ DIFFERENTIALLY DRIVEN MEMORY ADDRESS AND DIRECTION MODE BIT RETRIEVED FROM THE I/O BUS BY THE CPU I/O TRANSCEIVER.
- ⑧ MEMORY ADDRESS AND MODE BIT RECEIVED BY THE mN601 AND READ-MODIFY-WRITE MEMORY OPERATION INITIATED.

DATA CHANNEL DATA OUT OPERATION - MEMORY DATA TO I/O CONTROLLER

- ⑨ DATA RETRIEVED BY CPU DURING READ PORTION OF MEMORY OPERATION IN PROGRESS. WRITE PORTION OF OPERATION NOT EXECUTED.
- ⑩ MEMORY DATA SENT FROM THE mN601 TO THE CPU I/O TRANSCEIVER.
- ⑪ MEMORY DATA DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE CPU I/O TRANSCEIVER.
- ⑫ DIFFERENTIALLY DRIVEN MEMORY DATA RETRIEVED FROM THE I/O BUS BY THE IOC I/O TRANSCEIVER.
- ⑬ MEMORY DATA RECEIVED BY THE mN603.

DATA CHANNEL DATA IN OPERATION - DATA FROM I/O CONTROLLER TO MEMORY

- ⑭ DATA SENT FROM THE mN603 TO THE IOC I/O TRANSCEIVER.
- ⑮ DATA DIFFERENTIALLY DRIVEN ON THE I/O BUS BY THE IOC I/O TRANSCEIVER.
- ⑯ DIFFERENTIALLY DRIVEN DATA RETRIEVED FROM THE I/O BUS BY THE CPU I/O TRANSCEIVER.
- ⑰ DATA RECEIVED BY THE mN601.
- ⑱ DATA WRITTEN IN MEMORY DURING WRITE PORTION OF MEMORY OPERATION IN PROGRESS.



NOTE:  
SEE mN601 I/O OPERATIONS AND I/O TIMING TABLE FOR DETAILED INFORMATION.  
●● TIME IS DEPENDENT UPON THE LENGTH OF THE I/O BUS BETWEEN THE mN601 AND THE mN603.

DO-04233

This page intentionally left blank.



### LOGIC CONVENTIONS

Data General logic schematics are drawn in close accordance with Mil-Std-806C. Using this convention, logical functions are drawn as physically implemented; i.e., where discrete gates are used to implement a function, the gates are shown. On the other hand, where a more complex integrated circuit is used, e.g., a memory transceiver, that function is shown as a rectangular box instead of the gates comprising the function.

Throughout this manual, a distinction is frequently made between electrical levels and logical values. Electrical levels are referred to as high or low; logical values, as 1 or 0.

#### Signal Names

The voltage level at which a signal is said to be 'asserted' (true) is a matter of definition. To distinguish between signals that are asserted high (0=L, 1=H) and those that are asserted low (0=H, 1=L), a naming convention has been adopted in Data General's documentation which defines the relationship between the logical value and electrical level of a signal. Signal names which include a horizontal bar over the name, e.g., WRITE, are asserted when they are at a low electrical level; signal names without the bar, e.g., WRITE, are asserted when they are at a high electrical level.

Logical functions often require more than one binary signal. For instance, three lines are required to express an octal digit. In general, these closely related signals are individually identified by effectively subscripting a common label. For example, MB0 through MB15 are all required to completely specify a function. All or part of this group of signals is identified by placing brackets around the range of subscripts included, as MB < 0-15 > . In this case, the suffix carries the information that there are sixteen MB lines under discussion, from MB0 through MB15, inclusive.







Data General Corporation, Westboro, Massachusetts 01581