AMPRO

COMPUTERS, INCORPORATED

LITTLE BOARD/186

TECHNICAL MANUAL

P/N: A74011-C

67 East Evelyn Ave. ● Mountain View, CA 94041 ● (415) 962-0230

PREFACE

This manual is for integrators of systems based on the AMPRO Little Board/186 single board computer. It contains information on hardware requirements and interconnection, and details of how to use the system. There are five chapters, organized as follows:

Chapter 1 — INTRODUCTION - General information pertaining to the Little Board/186, its major features, and a brief functional description.

Chapter 2 — INTEGRATING A SYSTEM - Descriptions of the external components necessary to construct systems based on the Little Board/186 with floppy and hard disk drives. Included are tables listing the pinouts of each of the board's connectors, as well as special considerations and specifications concerning peripheral devices.

Chapter 3 — OPERATION WITH PC-DOS - Discussion of PC and PC-DOS compatibility. Information on system customization options, including use with various types of printers, modems, and floppy and hard disk configurations. Also includes brief descriptions of the AMPRO-specific DOS utilities and drivers.

Chapter 4 — THEORY OF OPERATION - Detailed technical information on Little Board/186 hardware.

Chapter 5 — PROGRAMMERS' REFERENCE - Port addresses and other programming considerations for custom programming of Little Board/186, including information pertaining to use of the ROM-BIOS.

PLEASE NOTE

Specifications are subject to change without notice. The contents of this document are believed to be accurate. If errors are found, please notify AMPRO at the address shown on the title page of this document.

The following is a list of trademarks used within this document:

IBM: International Business Machines
MS-DOS: Microsoft Corp.
LOTUS 1-2-3: Lotus Development Corp.
LITTLE BOARD, LITTLE BOARD/186, SCSI/PLUS: AMPRO Computers, Inc.

TABLE OF CONTENTS

CHAPTER 4 - THEORY OF OPERATION

CHAPTER 5 - PROGRAMMER'S REFERENCE

APPENDIX A - SCSI/PLUS ARCHITECTURE OVERVIEW

APPENDIX B - TYPICAL INTERFACE CABLES

APPENDIX C - BOARD DOCUMENTATION

APPENDIX D - COMPONENT DATA SHEETS

CHAPTER 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The Little Board/186 is a compact, high performance 16-bit, PC-DOS based single board computer capable of replacing an entire computer system in many embedded microcomputer applications.

Included on the 5.75 X 7.75 inch SBC are an 8 mHz 80186 16-bit CPU, 512K bytes of RAM memory, 16K to 128K bytes of EPROM memory, two RS232C serial ports, a Centronics printer port, a floppy disk controller, and a multi-master SCSI I/O expansion bus. The SCSI bus serves multiple functions, including interface to hard disk controllers, inter-board networking, and I/O port expansion.

Typical Little Board/186 applications include network file servers, robotics, data logging, protocol conversion, data base processing, point-of-sale terminals, telecommunications, and industrial process control. In such applications, the board's 8 MHz 80186 16-bit microprocessor provides up to a four-fold performance advantage over the 4.77 MHz 8088 CPU found in standard PC's.

AMPRO's proprietary ROM-BIOS (included on the board) allows the Little Board/186 to directly "boot" and run standard IBM DOS versions 2.x and DOS 3.x, using a standard ASCII terminal connected to one of the board's serial ports as the console (keyboard and display) device. "MS-DOS generic" software normally runs without modification in this environment. However, IBM PC video compatibility (e.g. programs which write directly to "video RAM") is not supported, except by means of the Video RAM Emulator option (described below).

1.2 FEATURES

- Boots and runs IBM PC-DOS 2.x and 3.x
  - Data and file compatible with IBM PC
  - Runs "MS-DOS generic" programs
- 3X the computing power of a PC
- Mounts directly to a 5-1/4" disk drive
- Uses less than 7 watts of power
- Complete 16-bit high performance Single-Board Computer System:
  - 8 MHz 80186 CPU with DMA and counter/timers
  - 512K RAM (0 wait states) -- 16K-128K EPROM
  - Mini/Micro Floppy controller (4 drives, DSDD, 40/80 track)
  - 2 RS232C Serial Ports -- 1 Centronics Printer Port
  - PC-DOS 2.x and 3.x compatible ROM-BIOS
- SCSI (SASI) Multi-Master I/O Expansion BUS:
  - SASI Disk/Tape Controller Compatible
  - ANSC X3T9.2 (SCSI) Compatible
  - Multiple Little Board Networking
- Expansion options include:
  - Video RAM Emulator
  - Multi-Function Expansion Adapter (RAM, 8087, clock, RS232/422)
  - Prototype Adapter
  - SCSI/IOP (data acquisition & control interface)

1.3 SCSI/PLUS (tm) Multi-Master Bus

A 50-pin "ribbon cable bus" interface which meets the specifications for the popular Small Computer System Interface (SCSI) -- formerly called "SASI" -- provides a general purpose multi-master I/O expansion bus. All SCSI Initiator and Target functions are fully supported, including bus arbitration and disconnect/reselect.

AMPRO is a leading proponent of the use of SCSI for more than mass storage. To emphasize this wider use of SCSI, we have given it a name: "SCSI/PLUS". An SCSI/PLUS Architecture Overview appears in Appendix A.

Applications for the board's SCSI/PLUS bus include both direct and shared use of a wide variety of controllers and devices, as well as tightly coupled Little Board networks. Devices soon to be available for connection via the SCSI/PLUS Bus include:

- Disk, tape, and optical mass storage subsystems
- Special function mass storage (RAM disk, bubble memory, etc.)
- Printer subsystems (serial, parallel, laser, etc.)
- Communication and LAN interfaces (Ethernet, X.25, etc.)
- Real time clock
- Co-processors
- Data acquisition and control subsystems (Analog, Digital, etc.)
- Graphics controllers and subsystems
- Non-volatile RAM
- etc...

Alternatively, the 17 bidirectional I/O signals of the board's SCSI/PLUS interface may also be used as general purpose software-controlled digital I/O lines, without SCSI compatibility. In this case, the board's 8-bit SCSI bus ID input register can serve as an additional 8-bit input port.

1.4 OTHER EXPANSION OPTIONS

Several expansion options are available for use with the Little Board/186, including:

1.4.1 Video RAM Emulator

The Video RAM Emulator allows the use of software which writes directly to "video RAM" instead of making PC-DOS or ROM-BIOS function calls. Such software would otherwise require modification for use with the Little Board/186.

The Video RAM Emulator is a 5.75 X 7.75 inch daughter board which plugs into the CPU header on the Little Board/186. It can not be used in conjunction with any of the other daughter board options.

1.4.2 Multi-Function Expansion Adapter

A Multi-Function Expansion Adapter (EXPANSION/186) is available for the Little Board/186 which provides five key system options:

- 512K additional zero-wait-state DRAM.
- 8087 math coprocessor (400% the speed of an 80287)
- Two additional sync/async serial ports, with choice of RS232C, RS422, or external drivers and receivers.
- Battery-Backed Real Time Clock
- Buffered I/O bus (128 I/O locations)

The Multi-Function Expansion Adapter is a daughter board, identical in dimensions with the Little Board/186 (5.75 X 7.75), and plugs into the CPU header on the main board. It cannot be used in conjunction with any other daughter board options.

1.4.3 Prototype Adapter

Provides a user-determined general purpose I/O and memory expansion interface for the Little Board/186. Approximately 18 square inches of wire wrap space, along with pre-decoding and conditioning of 80186 signals, facilitate custom projects and product prototypes based on the Little Board/186. Especially useful in such applications as data acquisition, process control, test instrumentation, product demonstration, SCSI special function devices, etc.

The Prototype Adapter is a daughter board, identical in dimensions with the Little Board/186 (5.75 X 7.75), and plugs into the CPU header on the main board. It cannot be used in conjunction with any other daughter board options.

1.4.4 SCSI/IOP

The AMPRO SCSI/IOP is an intelligent I/O processor (IOP) which allows the addition of a wide variety of off-the-shelf STD Bus data acquisition and control interfaces to a Little Board/186. It connects via a 50-conductor flat ribbon cable plugged into the SCSI/PLUS bus connector on the Little Board/186, and does not preclude the use of any of the optional daughter boards.

The SCSI/IOP can also be used as a non-volatile RAM storage device (up to 48K bytes), as a system boot device (EPROM-Based), and as an SCSI Real Time Clock.

1.5 SPECIFICATIONS

CPU/DMA/CTC

- 8 MHz 80186

MEMORY

- 512K bytes zero wait state DRAM
- Two 28-pin sockets for 2764-27512 (16K to 128K)

SERIAL I/O

- Two RS232C compatible ports
- Based on Signetics 2681 dual UART
- Software controlled baud rates, 50-38.4K baud
- Four signals per port: data in/out, status in/out

PARALLEL I/O

- Centronics compatible printer port
- 8-bit ID input port

FLOPPY DISK INTERFACE

- Mini and micro floppy compatible
- 1 to 4 drives, single/double density, 40/80 tracks, 1-2 sided
- 125 or 250 Kbits/second data rate
- Highly reliable digital phase-locked loop
- Software enabled write precompensation

SCSI/PLUS INTERFACE

- SASI compatible
- Full ANSC X3T9.2 (SCSI) compatible (all modes)
- Based on NCR 5380 SCSI bus controller
- Usable as 17 bidirectional I/O lines (48 mA sink)
- Max. SCSI throughput: 1 Mbyte/sec (read)
  .75 Mbyte/sec (write)

POWER

- Same power connector and voltages as mini and micro floppy drives
- +5V at 1.25 A.(typ.), +9V to +15V at .05 A.(typ)
- On-board -12 VDC converter

ENVIRONMENT

- Operating temperature: 0-55 degrees Centigrade
- Relative Humidity: 5-95% (non-condensing)
- Altitude: 0 to 10,000 feet

PHYSICAL

- 7.75 x 5.75 x 0.75 inches (5-1/4 disk drive form factor)

SOFTWARE

- AMPRO ROM-BIOS (boots PC-DOS 2.x or 3.x) in two 2764s
- PC-DOS Support Software
- Optional debugger/monitor program in two 2764s

DOCUMENTATION

- Little Board/186 Technical Manual
- PC-DOS Support Software User's Manual
- Optional: 5380 Technical Manual
  SCSI/PLUS Technical Specification

CHAPTER 2

INTEGRATING A SYSTEM

2.1 INTRODUCTION

This chapter describes what is required to build a floppy- or hard disk-based computer system using the Little Board/186. Details are provided concerning external device requirements, the board's connector pinouts, and how to prepare the board for use with periperals such as terminals, printers, and modems.

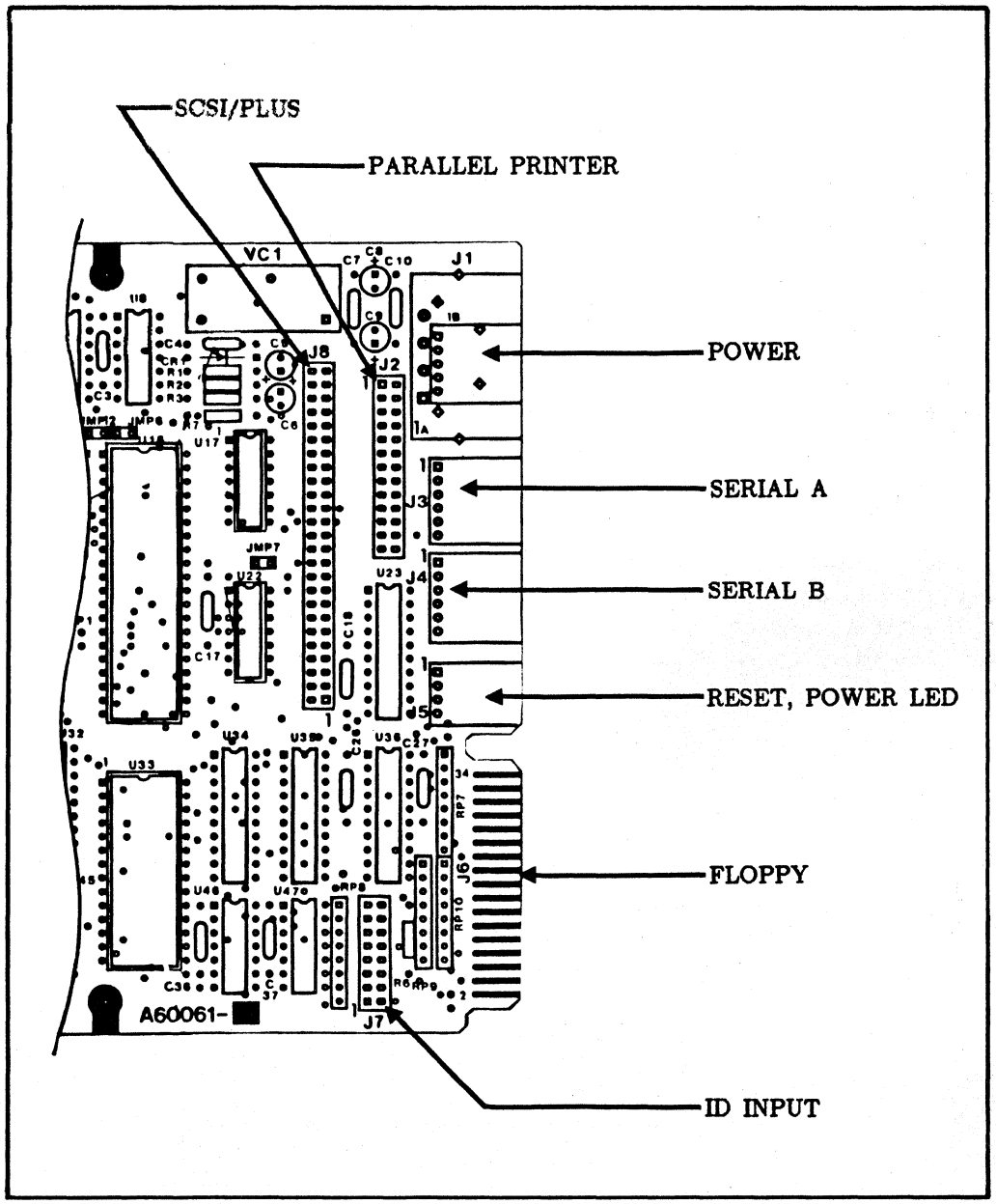


Figure 2-1. Little Board/186 Connector Locations.

2.2 BASIC REQUIREMENTS

The Little Board/186 is designed to use a standard RS232C ASCII terminal as a console device, providing both keyboard and display. You may use nearly any such terminal, providing its control codes are supported by your applications software.

The ROM-BIOS included on Little Board/186 allows you to use standard IBM PC-DOS versions 2.x or 3.x as the board's operating system. The utilities and drivers supplied on the AMPRO Little Board/186 PC-DOS Support Software diskette allow operation with 40-track, 80-track, single- or double-sided mini and micro floppy disk drives -- in various combinations -- and support a wide variety of SCSI hard disk controllers and drives. PC-DOS version 3 is required for hard disk systems; version 3.2 (or later) is required if full 80-track support is desired.

Centronics type parallel printers, and most RS232-compatible serial printers and modems may be directly connected to Little Board/186's I/O interface connectors.

2.3 SYSTEM CONNECTIONS

Figure 2-1 shows the board's external I/O connectors. The following paragraphs describe each connector interface and indicate special requirements for external devices. Mating connectors for the seven I/O interface connectors on the Little Board/186 are in Table A-4 of Appendix A.

2.3.1 DC Power Input

The power connector (J1) pinout is identical with that of power connectors on nearly all 5-1/4 inch floppy disk drives. Note that pin 1 on J1 is reversed from the other connectors on the Little Board/186. Refer to Table 2-1 for power connections, and Figure 2-2 for typical connector wiring.

CAUTION

Be sure the power plug is correctly wired before applying power to the board.

Table 2-1. Power Connections (J1)

Pin	Signal Name	Function
1	+12VDC	+6 to + 15VDC
2	Ground	Ground return
3	Ground	Ground return
4	+5VDC	+5VDC +/- 5%

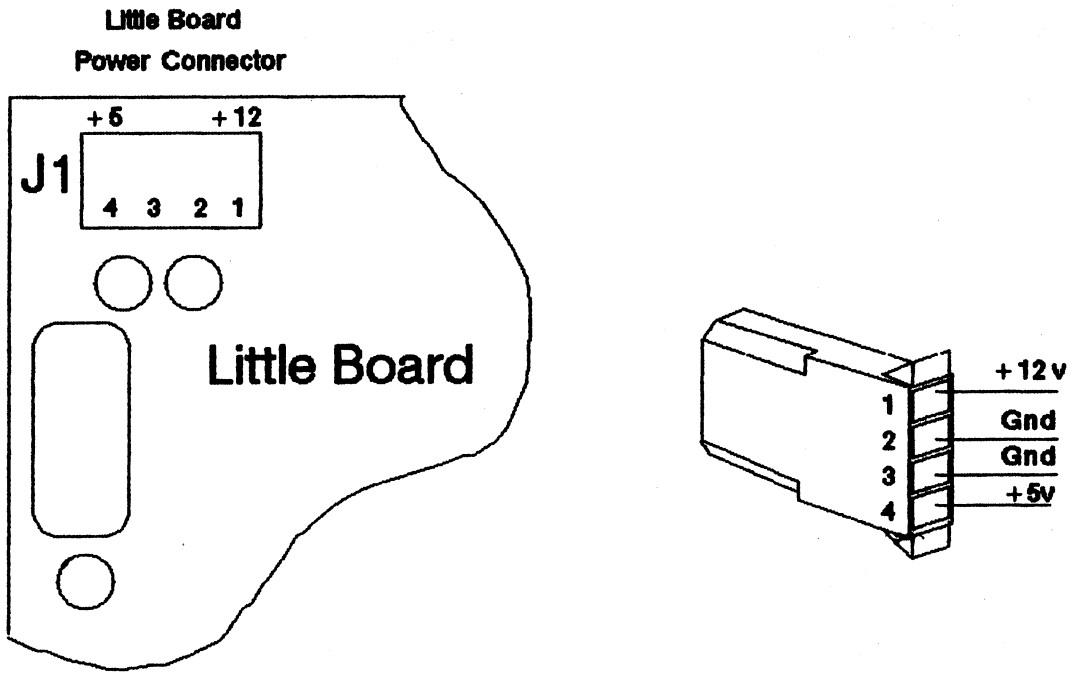


Figure 2-2. Power Connector Wiring

2.3.2 RESET, Power LED

J5 provides connection for an external s.p.s.t normally open switch to provide a manual RESET signal. In addition, a 15 mA current source provides power for an LED power-on indicator. Refer to Table 2-2 for the pinout of connector J5.

Table 2-2. RESET, Power LED Connector (J5)

Pin	Signal Name	Function
1	Ground	To LED Cathode
2	LED	To LED Anode
3	Ground	To one side of RESET switch
4	RESET	To other side of RESET switch

2.3.3 Serial Ports

Table 2-3 gives the connector pinout and signal definitions for each of the two RS232C serial ports. Serial Port A is board connector J3, and Serial Port B is board connector J4. Appendix B gives typical cable wiring for connection to terminals, serial printers, and modems.

Table 2-3. Serial Connectors (J3/J4)

Pin	Signal Name	Function	in/out	DB-25 Pin (DCE)
1	Ground	Protective Ground	--	1
2	Ground	Signal Ground	--	7
3	TxD	Data Output	out	3
4	HSO	Hand Shake Out (RTS)	out	5
5	RxD	Data Input	in	2
6	HSI	Hand Shake In (CTS)	in	20

The default ROM-BIOS "console" interface is Serial Port A. To this port you may connect nearly any type of RS232C ASCII terminal to provide monitor and keyboard functions. Asynchronous baud rates up to 38.4 Kbits per second may be utilized.

Some terminals require hardware handshaking when used at baud rates in excess of 9600. If hardware handshaking is needed, connect the terminal's CLEAR TO SEND output to the board's HANDSHAKE IN signal. Each port's HANDSHAKE OUT signal can optionally be used to stop external devices from transmitting data to the board. (The AMPRO SETCON.SYS software driver is used to enable this feature.)

Serial Port B can be used for connection of a printer, modem, or other serial device. As with port A, the main interface consideration is that the device be RS232C ASCII compatible. Printers and modems generally require handshaking in one or both directions.

2.3.4 Parallel Printer Port

The parallel printer connector, J2, has a pinout that allows the use of flat ribbon cable between the J2 header and the first 26 lines of a 36 pin male Centronics type connector at the printer end. Note that the pin numbering for the printer's interface connector differs from that of the header connector on the board. J2 is numbered as shown in Figure 2-1. Although some printers may include unique signals not shown, the signals provided by J2 are adequate for normal operation of most printers.

Refer to Table 2-4 for printer connector signal pinouts and definitions. Note that the pin numbering convention for the board's header connector differs from that of Centronics connectors. To clarify this, each signal's corresponding Centronics connector pin number has been included in Table 2-4.

Table 2-4. Parallel Printer Connector (J2)

J2 Pin	Signal Name	Function	in/out	Centronics Conn. pin
1	-DS	Sample input data	out	1
3	Data 1	LSB of printer data	out	2
5	Data 2	:	out	3
7	Data 3	:	out	4
9	Data 4	:	out	5
11	Data 5	:	out	6
13	Data 6	:	out	7
15	Data 7	:	out	8
17	Data 8	MSB of printer data	out	9
19	-ACK	Character accepted	in	10
21	BUSY	Cannot receive data	in	11
23	PAPER END	Out of paper	in	12
25	SELECT	Ready to receive data	in	13
2-22	GROUND	Signal grounds	--	19-29
24	---	(Not used)	--	30
26	-INIT	Initialize controller	out	31

The cable required is identical to that used with the Tandy (Radio Shack) TRS-80 Model 100 portable computer. The Tandy cable part number is 26-1409.

2.3.5 Floppy Disk Interface

Table 2-5 lists the floppy disk drive interface connector (J6) pinout and signals. A single 34-conductor PC edgecard connector is used at the Little Board end, while there can be from 1 to 4 connectors for connecting the disk drives.

Table 2-5. Floppy Disk Interface Connector (J6)

Pin	Signal Name	Function	in/out
2	-LOW SPEED	Speed select (option)	out
4	---	(Not used)	--
6	-DRIVE DEL 4	Drive Select 4	out
8	-INDEX	Index pulse	in
10	-DRIVE SEL	Drive Select 1	out
12	-DRIVE SEL 2	Drive Select 2	out
14	-DRIVE SEL 3	Drive Select 3	out
16	-MOTOR ON	Motor on control	out
18	-DIR SEL	Direction select	out
20	-STEP	Step	out
22	-WRITE DATA	Write data	out
24	-WRITE GATE	Write gate control	out
26	-TRACK 00	Track 00	in
28	-WRITE PRT	Write protect	in
30	-READ DATA	Read data	in
32	-SIDE ONE	Side select	out
34	-READY	Drive ready	in
1-33	(all odd pins)	Signal grounds	--

Nearly any type of soft-sectored, single or double-sided, 40 or 80 track, mini or micro floppy disk drive is usable with the Little Board/186. Naturally, the higher the quality of the drives you use, the better your system's reliability. Here are some considerations:

- The drives used must be compatible with the AMPRO floppy disk interface (see Table 2-5), and must provide the Drive Ready signal.

- High quality, DC servo direct drive motor floppy disk drives are recommended.

- More than one type of floppy disk drive, up to four, can be present in the system, and in any mix.

- If you plan to "boot" a standard copy of PC-DOS, drive A must be a 48 tpi (40 track) 5-1/4 inch double-sided mini floppy drive. Drives B, C and D can be any other system-compatible drive. PC-DOS version 3.2 (or later) allows the creation of a 96-tpi (80 track) boot diskette, but a 48 tpi drive is required to initially load the system diskette as it is supplied by IBM.

- Each disk drive must be jumpered for a specific Drive Select value, 1 through 4. Consult your drive documentation.

- Resistive terminations should be installed only on the drive connected to the last interface cable connector (farthest from the computer).

- When using drives with a Head Load option, jumper the drive for "head load with motor on" rather than "head load with drive select."

2.3.6 SCSI/Plus Interface

The SCSI/Plus interface (J8) uses a 50-pin male header connector to interface with SCSI-compatible peripherals. Table 2-6 shows the signal names and pin numbers. Refer to your disk controller documentation, or the ANSI SCSI specification for information on the signal functions.

Table 2-6. SCSI Interface Connector (J8)

Pin	Signal	Function
1 - 49	(All odd pins)	Signal grounds
2	-DB0	Data Bit 0 (LSB)
4	-DB1	" " 1
6	-DB2	" " 2
8	-DB3	" " 3
10	-DB4	" " 4
12	-DB5	" " 5
14	-DB6	" " 6
16	-DB7	" " 7 (MSB)
18	-DBP	Data Parity
20,22,24	Ground	Signal Ground
25	---	(Not used)
26	TERMPWR	Termination +5VDC
28,30	Ground	Signal Ground
32	-ATN	Attention
34	Ground	Signal Ground
36	-BSY	Busy
38	-ACK	Transfer Acknowledge
40	-RST	Reset
42	-MSG	Message
44	-SEL	Select
46	-C/D	Control/Data
48	REQ	Transfer Request
50	-I/O	Data direction

This interface can serve a variety of purposes, including connection of hard disk controllers, tape controllers, printer and communications servers, etc. In addition, the interface signals may be used as direct input/output lines, allowing the connection of TTL-level controlled devices and sensors, etc. (The output signals are open collector drivers capable of sinking 48 mA, and may be enabled and disabled under software control.) On-board removable resistor networks provide bus termination.

NOTE

The on-board resistive termination networks (U17, U22) should be present on two, and only two, SCSI bus devices. Be sure that the board's SCSI bus is terminated in at least one place (generally on the board); a non-terminated SCSI bus may "hang" the system up due to indeterminate signal levels.

2.3.7 ID Input Port / Drive Quantity Jumpers

Eight pairs of jumper pins (J7) provide an ID Input Port, which are generally used by SCSI-related software to determine the board's SCSI bus ID for bus arbitration, and by the board's ROM-BIOS to determine the number of floppy drives connected to the system (see next section). If not required for these purposes, the ID Input Port can be used as a general purpose 8-bit input port connector, with a flat ribbon cable plugged onto J7.

2.4 BOARD JUMPER CONFIGURATIONS

The board contains nine sets of jumpers, which may be used to customize the board's operation. The jumper sets consist of either two or three pins, with pin 1 a square pad, and are outlined with white boxes on the component side of the board. The options available through these jumpers are described in the following paragraphs.

NOTE

For normal operation, no jumper setup is required, other than the possible use of J7 for SCSI address jumpering and floppy drive quantity setup.

Most factory jumper settings are made by means of traces on the bottom side of the board. One exception is JMP5 which is generally shorted by means of a shorting plug on the component side of the board.

J7 - ID Input Port / Drive Quantity Jumpers

This 16-pin header is normally used to set the board's SCSI bus address, and also to indicate the number of floppy disk drives to be used. The connector pins are numbered from 1 to 16, with odd pins opposite even. The pairs are designated: J7-1/2, J7-3/4, J7-5/6, J7-7/8, J7-9/10, J7-11/12, J7-13/14, J7-15/16. Two pairs are used to indicate to the ROM-BIOS how many floppy drives are connected to the system; three pairs designate the board's SCSI bus ID; the remaining three pairs are currently unused. Refer to Table 2-7 for jumper settings.

Table 2-7. ID Input Port (J7) Jumpering

Pairs 1/2 and 3/4: Used to initialize the AMPRO ROM-BIOS floppy drive quantity value. Programmed as follows:

No. of Drives	1/2	3/4
1	in	in
2	in	out
3	out	in
4	out	out

Pairs 5/6, 7/8, 9/10: Reserved additional ID bits for SCSI/PLUS 6-bit ID, future use.

Pairs 11/12, 13/14, 15/16: Sets the board's SCSI ID, as follows:

SCSI Bus ID	11/12	13/14	15/16
0	in	in	in
1	in	in	out
2	in	out	in
3	in	out	out
4	out	in	in
5	out	in	out
6	out	out	in
7	out	out	out

JMP1,2 - EPROM Type

These jumpers are used to program the board for various types of EPROM devices. Both EPROM sockets must contain the same type of EPROM. These two jumpers are set as shown in Table 2-8.

Table 2-8. EPROM Jumper Configurations

EPROM Type	JMP1	JMP2: 1-2	JMP2: 2-3
2764	open	short	open
27128	open	short	open
27256	short	short	open
27512	short	open	short
Factory Setting	open	short	open

JMP3 - DRQ1

This jumper, when shorted, holds the 80186 DRQ1 input inactive.

FACTORY SETTING: shorted

JMP4 - 64K/256K RAM Select

This jumper, when shorted, selects 64K DRAM timing and control.

FACTORY SETTING: open

JMP5 - Clock

This jumper, when shorted, connects the system 16 mHz master clock bus to the 16 mHz hybrid oscillator (U6). It is intended for test purposes only.

FACTORY SETTING: shorted

JMP6 - SCSI-RDY

This jumper, when shorted, connects the READY output of the 5380 SCSI device to the ARDY input of the 80186. This signal is not used by the AMPRO ROM-BIOS.

FACTORY SETTING: shorted

JMP7 - SCSI Termination Power

This jumper, when shorted, connects the board's +5VDC to pin 26 of the SCSI bus connector. This is intended to provide a current source for external termination, and is only required if a cable-mounted terminator is to be used.

FACTORY SETTING: open

NOTE

No on-board protection diode is provided. Consequently, this jumper must NOT be installed on more than one bus device.

JMP8 - U45 Option

This jumper is shorted for 8 mHz-only operation of the 1772 FDC device.

NOTE

If this jumper is shorted, the clock multiplexor IC, U45 must NOT be present; if U45 is present, this jumper must be open.

FACTORY SETTING: open (U45 present)

JMP9 - NMI

This jumper, when shorted, holds the 80186 NMI input inactive.

FACTORY SETTING: shorted

JMP10 - TEST

This jumper, when shorted, holds the 80186 -TEST input low.

FACTORY SETTING: shorted

JMP11 - HOLD

This jumper, when shorted, holds the 80186 HOLD input inactive.

FACTORY SETTING: shorted

JMP12 - Diagnostic Jumper

Shorting this jumper grounds the 2681 serial controller's input port bit 6 (IP6). In addition, loopback from the 2681's output bit 6 (OP6) to IP6 is not functional (OP6 is open collector).

FACTORY SETTING: open

2.5 BOOTING THE SYSTEM

Assuming that you intend to boot the system from a standard IBM PC-DOS version 2.x or 3.x system diskette, you will only need to connect the board to one or more double-sided 48 tpi mini floppy drives, a terminal, and a source of power. A PC or other computer can be used as the terminal, using a suitable terminal emulation program (i.e., the AMPRO SuperDuo program).

The cable connections between the board's Serial Port A and an RS232C ASCII terminal are given in Appendix B. For first time startup, set the terminal as follows:

Baud Rate:	9600
Data Bits:	8
Parity:	off
Stop Bits:	1
Handshaking:	none

Set your terminal so that the Most Significant Bit (data bit #8) is transmitted as a 0 ("low" or "space"). Some terminals do not have a switch to do this, automatically sending a zero for data bit #8 when parity is off. The AMPRO ROM-BIOS does not mask the MSB when 8 bit transmission is selected.

With a terminal connected and turned on, the system is ready to boot. When power is applied, the ROM-BIOS will attempt to read the operating system from disk. If no disk is in the drive, the system will wait until a disk is in place, and the drive latch is closed. The system will then read the operating system from the disk in drive A.

If the drive's LED lights, but nothing else happens, try inserting the flip-side of the disk and pressing RESET. If this doesn't help, refer to the next section for troubleshooting information.

2.6 TROUBLESHOOTING

If the system did not work the first time, or fails sometime, you may have to troubleshoot it. The following are some suggestions:

- Recheck all wiring, soldered connections.
- Check that power is available from the power supply.
- Be certain that the drives are working, and are jumpered correctly.

> **NOTE**
> IBM PC drives are not jumpered in a "standard" manner; for use with Little Board/186, be sure drive A is jumpered as Drive Select 1, B as 2, etc. Also, the PC's drive cables have swapped drive select wires, rather than straight through connections as required by the Little Board/186.

- Verify that the drive you are using provides the required Drive Ready signal.
- If more than one drive select LED indicator lights during power-up, with drive handles closed (across slot), the board may be incorrectly connected to the drive cables. Switch the computer OFF and reverse the drive cable connector at the Little Board/186.
- Check the drive termination resistor pack(s) for proper location. Normally, this will be located at the drive connected at the end of the drive cable, and on only one drive.
- If you have the debugging Monitor EPROM option, you can verify some of the system functions using the debugger and other tools in the Monitor. Refer to the EPROM Monitor User's Manual.

If your system still does not boot after following these instructions, contact AMPRO customer service for assistance.

CHAPTER 3

OPERATION WITH PC-DOS

3.1 INTRODUCTION

Assuming you have successfully booted PC-DOS as described in Chapter 2, you will probably want to take advantage of the flexibility designed into the Little Board/186 ROM-BIOS and support software to create a customized PC-DOS based system.

This chapter provides an overview of the system configuration options that are available under the PC-DOS operating system, as well as a discussion of the degree of compatibility that the Little Board/186 offers relative to software written for operation on the IBM PC and compatibles.

A combination of standard IBM PC-DOS utilities, along with AMPRO-specific drivers and utilities, allows you to create a highly customized system. The required AMPRO-specific drivers and utilities are supplied on the Little Board/186 PC-DOS Support Software diskette. Please refer to the user's manual (P/N A74012) provided with that software, for full descriptions, operating instructions, and installation infomation.

3.2 PC SOFTWARE COMPATIBILITY

This section is intended to help you understand the extent of compatibility provided by the Little Board/186 with software written for the IBM PC and compatibles ("standard PC").

3.2.1 Hardware Considerations

The Little Board/186 is based on a 80186 integrated, high-performance 16-bit microprocessor, which provides a functional superset of the 8-bit 8088 microprocessor used in the "standard PC." Programs written for an 8088 microprocessor can run on an 80186 without modification, but with a performance improvement of up to 300 percent.

Many of the hardware devices present on the Little Board/186, and their I/O port addresses, differ from those of the "standard PC." This includes: the 2681 serial communications controller; the 1772 floppy disk controller; the hard disk interface (SCSI); and the DMA, interrupt, and timing controllers contained within the 80186 microprocessor. Furthermore, the Little Board/186 utilizes an RS232C ASCII terminal as a user console (keyboard and monitor) rather than the keyboard and display controller used in a "standard PC."

As a result, programs which make direct access to board hardware, including both I/O ports and display controller "video RAM," rather than using the operating system or ROM-BIOS functions provided for the same purpose, can not be used without I/O driver modifications.

3.2.2 Operating System

System Boot

The AMPRO ROM-BIOS normally supplied on the Little Board/186 allows the use of IBM PC-DOS versions 2.x or 3.x as the board's operating system. PC-DOS version 3.x is required for hard disk usage, while version 3.2 is required for systems with an 80-track (720K) drive A.

Files and Data

When operated under PC-DOS, the Little Board/186 provides full PC file and data compatibility. Diskettes may be copied, formatted, verified, etc. on either for the other. All of the PC-DOS 40-track mini floppy formats, and the PC-DOS Version 3.2 80-track micro floppy formats, are supported, including single- and double-sided, and eight and nine sectors per track.

Commands, Drivers, and Utilities

Most of the standard PC-DOS operating system internal commands, and many of the disk-based utilities, function normally on the Little Board/186. Because the Little Board/186 ROM-BIOS does not contain Basic, however, none of the Basic programs included on the PC-DOS diskettes are usable. In addition, the graphics related utilities can not be used. Several PC-DOS functions require the use of the installable AMPRO drivers contained on the Little Board/186 PC-DOS Support Software diskette, as discussed in the software user's manual.

3.2.3 ROM-BIOS Functions

In addition to supporting the standard PC-DOS functions, the Little Board/186 ROM-BIOS provides a software interrupt interface which is a highly compatible subset of the "standard PC" ROM-BIOS software interrupt structure.

The console and keyboard ROM-BIOS interrupts are mapped to Serial Port A, so that an RS232 ASCII terminal can be used for the required keyboard/monitor functions. The board's Serial Port B is supported as the PC-DOS COM1 port; the Centronics printer port is LPT1. The Time of Day clock and Disk I/O functions are also supported in the standard manner.

The video display interrupt (INT 10H) of the ROM-BIOS supports the Write TTY function only. This results in full compatibility with "MS-DOS generic" programs, provided the program includes a terminal installation utility. Compatibility with the "standard PC" ROM-BIOS functions for cursor addressing, clear screen, etc., requires use of the terminal driver included on the Little Board/186 PC-DOS Support Software diskette.

Hard disk support (INT 13H) is also provided, using the board's SCSI port. This includes the ability to configure a system to boot from a SCSI-based hard disk drive.

A detailed discussion of the Little Board/186 ROM-BIOS software interrupts is given in Chapter 5. Information on the installation and use of the AMPRO-specific drivers and utilities are provided in the Little Board/186 PC-DOS Support Software User's Manual.

3.2.4 Applications Software

MS-DOS Generic Programs

Programs and utilities written for operation on any MS-DOS system are called "MS-DOS Generic." These programs restrict their system access exclusively to functions provided by the operating system. MS-DOS Generic programs are "hardware independent"; they can be used on a variety of hardware implementations, including systems such as the Little Board/186 that use RS232C ASCII terminals as the system console. Such "well behaved programs" nearly always run without modification on the Little Board/186. MS-DOS Generic programs generally have a terminal installation utility which allows you to specify the terminal to be used as the system console device, usually from a menu.

Here are a few popular application programs known to be available in MS-DOS Generic versions: Multiplan (Version 1), Wordstar (Version 3), DBase II, SuperCalc2, the T/Maker Integrated Software package. In addition, most languages and software development tools have MS-DOS Generic versions, including: Microsoft C (Version 3), Basic, and MASM; Turbo Pascal, Palasm, Abel; the Intel 86-family development tools.

Programs Which Make ROM-BIOS Calls

Many popular programs written for use on the "standard PC" violate the rules of operating system usage, making direct ROM-BIOS calls, accessing system I/O ports and video RAM directly, etc. Of these, programs which only make direct ROM-BIOS calls can often be used on the Little Board/186, providing that the optional video and keyboard driver software from the Little Board/186 PC-DOS Support Software are in use.

Programs Which Make Direct Hardware Access

Programs that talk directly to hardware (floppy controller, serial ports, keyboard port, video RAM, etc.) generally do not run on the Little Board/186 without modification. Some examples include:

- Copy protected programs: often use floppy controller, serial port, or printer port hardware.
- Communications programs: generally access serial port hardware
- Programs using graphics: generally access display controller hardware
- Programs that write to Video RAM

Included in this group are many "consumer programs," including: Lotus 1-2-3, Symphony, DBase III, Flight Simulator, etc.

A "Video RAM Emulator" daughter board is available for the Little Board/186, which simulates a PC display controller. The Video RAM Emulator detects writes to "video RAM"; its associated software drivers forward the data to the RS232C console device, using appropriate terminal control sequences. Using

the Video RAM Emulator, software which is intended for operation on a standard monochrome video display controller will usually run properly, with the exception of the use of bit-mapped graphics.

Designing Software for Compatibility

To provide full compatibility between the Little Board/186 and the standard PC, the application software must simply confine itself to standard PC-DOS and the supported ROM-BIOS functions, rather than performing direct hardware accesses.

3.3 CONFIGURATION OPTIONS

Chapter 2 discussed the basic requirements for booting from a standard "out of the box" IBM PC-DOS (versions 2 or 3) system diskette. Once your system has booted successfully, you can tailor your software configuration to a custom hardware configuration. Your options include:

- Terminal baud rates other than 9600 baud (up to 38.4K)
- Parallel or Serial Printers
- Modems
- Additional -- or different types of -- floppy drives
- One or more hard disk drives
- RAM disk

A brief discussion of each of these configuration options follows. Actual configuration and installation details are provided in the Little Board/186 PC-DOS Support Software User's Manual (P/N A74012).

3.3.1 Terminal

Nearly any RS232C ASCII terminal can be used with your system. The initial terminal characteristics for first time booting must be set as described in Chapter 2 (9600 baud, 8 data bits, etc.). Once your system is initially booted, you can configure a custom system boot diskette for alternate console parameters. Baud rates up to 38.4K baud, as well as alternative data word formats, are available.

Display and keyboard drivers, available on the Little Board/186 PC-DOS Support Software diskette, provide mapping of the functions used to control a standard PC video display (cursor positioning, clear screen, etc.) and keyboard into the control codes required by your specific terminal.

3.3.2 Printers

The system can be used with both Centronics type parallel printers, and with RS232C ASCII serial printers. The Centronics port is supported as the DOS "LPT1" device. Most application software uses LPT1 as the default printer port, so printing to the parallel printer interface is automatic.

The board's Serial Bort B, supported as the DOS "COM1" device in the ROM-BIOS, can be used as a serial printer port, and can be configured for a wide variety baud rates and data word formats. Hardware handshaking (e.g. RTS/CTS) can also be configured for use with printers that require it.

3.3.3 Modems

Serial Port B can also be used as a modem interface, accessed as the DOS "COM1" device. Serial Port B data characteristics are initialized in the same manner as when the port is used as a serial printer interface, using the AMPRO utilities from the PC-DOS Support Software.

Most communications programs perform direct serial port I/O, rather than using the DOS or ROM-BIOS functions. Such programs must be customized before use. (See Chapter 5.)

The AMPRO LBCOMM.EXE program, included on the AMPRO PC-DOS Support Software diskette, is a full-featured communications program for the Little Board/186. LBCOMM offers remote system access, terminal emulation, and both XMODEM and ASCII file transfer functions.

Several other high quality communications programs are available specifically configured for the Little Board/186. These include:

- MEX-PC (NightOwl Software, Inc.)
- MICROLINK II (Wordcraft).

3.3.4 Unique Floppy Configurations

When your system initially boots from a standard PC-DOS system diskette, the floppy configuration is set for one to four 40-track (48 tpi) drives, depending on the setting of jumpers at position J7 (see Chapter 2). You can configure a system for use with 80-track mini or micro floppy drives, including the ability to boot from an 80-track device (requires PC-DOS 3.2).

One handy feature of PC-DOS is its built-in support for single-drive systems. If you jumper the Little Board/186 drive quantity jumpers (J7) to "one," the operating system will automatically assign drive letters "A" and "B" to the single floppy drive. With this configuration, you can copy files between two diskettes as though your system has two drives; PC-DOS will prompt you to change diskettes when needed.

3.3.5 Hard Disk Drives

Hard disk drives and controllers with SCSI (SASI) interfaces may be easily added to your system. Support for the first hard disk device, from which your system can be configured to boot, is contained within the Little Board/186 ROM-BIOS. Additional drives can be added as well, using the AMPRO HARD.SYS device driver. Consult the AMPRO PC-DOS Support Software user's manual for further details.

3.3.6 RAM Disk

PC-DOS Version 3 provides a useful virtual disk device driver, called VDISK.SYS, which allows you to configure one or more RAM disk drives based on memory on the Little Board/186.

CHAPTER 4

THEORY OF OPERATION

4.1 INTRODUCTION

This chapter is intended to provide a basic understanding of the functional operation of the Little Board/186 for programmers, hardware engineers, system integrators, and other technically oriented users.

The functional behavior of many of the board's devices is highly dependent on programming options. Therefore, to assist your understanding of the normal functions performed by these devices as used on the Little Board/186, some reference is made to the AMPRO ROM-BIOS default device configurations.

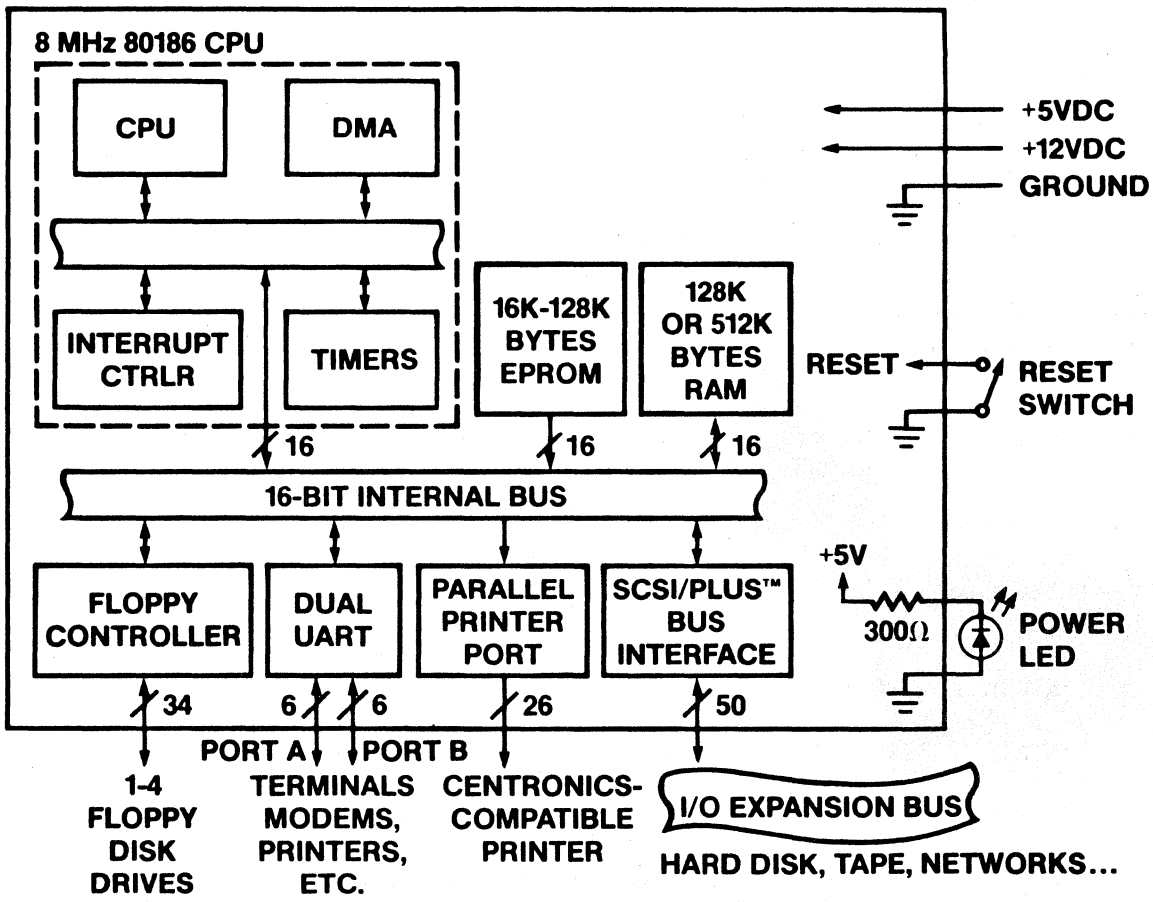


Figure 4-1. Little Board/186 Block Diagram

Data sheets are included in Appendix D for the 80186 (CPU), 2681 (UART), 1772 (FDC), and 5380 (SCSI) devices. You may also wish to obtain a copy of the following especially valuable publication, available from Intel Corporation:

AP-186: Introduction to the 80186 Microprocessor

A detailed technical manual on the 5380 is also available AMPRO for a nominal charge.

4.2 THE 80186 INTEGRATED CPU

The architecture of Little Board/186 (see Figure 4-1) is based on the 80186 integrated microprocessor unit. The 80186 device includes a 16-bit CPU, a 2-channel DMA controller, three 16-bit timers, a programmable interrupt controller, and programmable memory and I/O chip-select logic. The 80186 is more than a highly integrated version of the 8086 microprocessor; instruction execution and pipeline efficiency of the 80186 represent substantial performance improvements relative to the older 8086 device.

Since many of the 80186 characteristics are programmable -- including memory chip select addresses and timing -- the software has the responsibility to initialize these immediately on powerup. The discussion in this chapter assumes the default initialization performed by the AMPRO ROM-BIOS. Refer to Chapter 5 (Programmer's Reference) for additional programming details.

4.2.1 Chip Selects

The chip select outputs provided on the 80186 are used to control the I/O peripheral chips and EPROMs. Both the access addresses and wait state timing of these chip select outputs are programmable. Five of the six Peripheral Chip Selects (PCS0-5), in combination with a custom I/O Controller IC are used to provide device selects for all of the board's on-board I/O functions. No wait states are required.

4.2.2 Interrupts

The 80186 contains an internal four channel interrupt controller, utilized as follows:

INTR0: 2681 serial controller device interrupt.

INTR1: 5380 SCSI interface controller device interrupt.

INTR2: 1772 floppy disk controller device interrupt.

INTR3: Centronics printer port interrupt.

All are edge triggered interrupts and are sensed by the 80186 as active high. These interrupts are controlled using the 80186 internal interrupt controller.

4.2.3 Counter/Timers

The 80186 provides three 16-bit internal timers. Two of these have off-chip input and output terms, allowing their use as counters as well. The 2681 serial controller provides an additional general purpose timer channel, which is discussed in the section on that device. The three channels of the 80186 are used as follows:

Channel 0 — The AMPRO ROM-BIOS uses this counter/timer channel in conjunction with Channel 2 to provide an accurate real time clock function (18 ticks/second). Channel 2 acts as a prescaler, providing a 16 microsecond input rate to this channel.

Channel 1 — This counter/timer channel has its external counter input driven by the timer output of the 2681 serial controller. This allows the timing of long intervals, etc.

Channel 2 — This timer-only channel must be programmed to provide the required 16 microsecond DRAM refresh rate.

4.2.4 DMA

The 80186 internal DMA controller provides two powerful DMA channels. These are used as follows:

Channel 1 — This DMA channel forms an important part of the DRAM refresh logic. (See next section.) It is programmed to generate a linear 20-bit address, timed by timer channel 2, with no terminal count.

Channel 2 — This DMA channel is shared between the floppy and SCSI interfaces, as programmed in the FDC Control Register.

4.3 MEMORY

Little Board/186 supports EPROM address space of 16k to 128K bytes, for use with 2764 to 27512 standard JEDEC parts. The 80186 Upper Memory Chip Select (UMCS) is used for selection of the on-board EPROM memory. This chip select is programmed for 0 wait states and ignoring of external ready, for use with 250 nS access time EPROM's. The use of 250 nS EPROM's is required due to the 80186 data bus maximum output disable time.

A custom RAM Controller IC decodes DRAM timing directly from the 80186 control and status signals, so that 0 wait state operation with 150 nS DRAM is possible. RAM refresh addresses are provided by DMA channel 1 with control timing provided by one of the 80186 internal timer channels (ch2). The RAM Controller IC has an input pin which is used to select operation with either 64K or 256K bit DRAM devices. A low level on this pin programs the board for 64K bit DRAM operation.

The Lower Memory Chip Select (LMCS) and Middle Memory Chip Select (MMCS) lines are unused. Instead, the RAM Controller decodes its address range directly

from the address lines. Direct address decoding will also be used to decode the upper 512K of the 80186 megabyte address space on a plug-in memory expansion module. When this is done, UMCS will be used to disable access to the added memory in the high area for which the UMCS is programmed, depending on the EPROM size.

4.4 2681 SERIAL CONTROLLER

A 2681 dual Universal Asynchronous Receiver/Transmitter (UART) provides two channels of serial I/O. The 2681 also provides a number of signals for the parallel printer and floppy disk interfaces, and contains a programmable timer whose output can drive the 80186 counter/timer channel 1 input.

4.4.1 Serial I/O Ports

The 2861 provides two asynchronous serial I/O channels, featuring fully programmable serial data characteristics, including word length, parity, start/stop bits, and baud rate.

The device contains an internal baud rate generator, which operates from a 3.6864 mHz crystal source. Baud rates from 50 through 115.2K may be selected for each channel's receive and transmit data (independently). Both channels of the 2681 function in an identical manner.

RS232C signal levels are converted to and from TTL levels by a 75188/1488 line driver, and a 75189/1489 line receiver. An on board -12 volt DC-to-DC converter provides the -12VDC power for the line driver.

The 2681's OP0 (RTSA) and OP1 (RTSB) outputs generate each channel's Hand Shake Out signal, while the IP0 (CTSA) and IP1 (CTSB) inputs are used to provide each channel's Hand Shake In signal.

4.4.2 Signals Used by Other Interfaces

The 2681 Serial Controller contains an eight bit output port and a seven bit input port. Only four of these signals (OP0, OP1, IP0, and IP1) are used in conjunction with the board's serial ports. The other eleven signals are assigned as follows:

OP2 Drives the printer interface -DATA STROBE signal.
OP3 Drives the 80186 Counter/Timer channel 1 input signal.
OP4 Unused.
OP5 Drives the 1772 FDC controller -RESET input.
OP6 Looped to input pin IP6.
OP7 Drives the printer interface PRINTER INIT signal

IP2 Senses the printer interface BUSY signal.
IP3 Senses the floppy interface DRIVE READY signal.
IP4 Senses the printer interface -PRINTER SELECT signal.
IP5 Senses the printer interface -PAPER END signal.
IP6 Senses the state of output pin OP6 or JMP12.

The specific use of each of the above signals is covered in the floppy and printer interface sections, below. Note that output OP6 is looped to input

IP6, and also connected to JMP12 pin 1. Since OP6 is open drain, the state of jumper JMP12 can be sensed if OP6 is high.

NOTE

The 2681's output pins (OP0-7) are the compliment of the contents of the output port register. That is, setting a bit in the output port register results in a low (0) logic level at the corresponding output pin, and resetting a bit results in a high (1) level. The input pins (IP0-6), however, are not complimented. Therefore, the bits of the input register directly reflect the state of the corresponding input pins.

4.5 PARALLEL PRINTER PORT

An octal D-latch with a 24mA current sinking capacity is used as a Printer Data Register, driving the eight parallel printer port data lines. The eight data signals are written directly by the low order byte of the 80186 data bus by means of an output instruction. Six other input/output signals associated with this interface are also supported, as follows:

Signal Name	Connects To...	In/out	Polarity
DATA 1-8	Printer Data Register	Output	Normal
-DATA STB	2681 output OP2	Output	Normal
-PRINTER INIT	2681 output OP7	Output	Inverted
BUSY	2681 input IP2	Input	Normal
-ACK	80186 interrupt INT3	Input	Normal
PAPER END	2681 input IP5	Input	Inverted
SELECT	2681 input IP4	Input	Inverted

Each signal is shown as either inverted or non-inverted, which is relative to the actual external peripheral interface signal. When a signal is listed as "normal" polarity, this means that the interface connector pin has the **same** value as the programmed IC pin. For example, when 2681 input pin IP2 is high, the printer BUSY signal (J2, pin 21) is in a high state. (Refer to the NOTE in the section pertaining to the 2681 device, concerning output and input pin polarities.)

4.6 FLOPPY DISK INTERFACE

Most of the logic required for the floppy disk interface is provided by the WD1772 Floppy Disk Controller (FDC) device. The 1772 is a highly integrated device, and contains internal digital phased locked loop, digital write precompensation, CRC generation and checking, and motor control timing.

The 1772's interrupt request output connects to the INT2 interrupt input of the 80186. The 1772's DMA request output is one of two such signals selectively routed to the 80186 DR10 DMA request input, under control of the FDC Control Register.

The 1772's master reset pin is driven by the 2681 serial controller's OP5 output signal, such that the 1772 is held in a reset state when OP5 is low. This allows the 1772 to be reset under software control.

4.6.1 FDC Control Register

Additional control output signals, such as the four drive selects, are generated by the FDC control register. The FDC Control Register is written to directly by the low order byte of the 80186 data bus, using an output instruction. The FDC control register's output signals are utilized as follows:

The floppy disk interface READY input is inverted, and then connects to the 2681 serial controller's IP3 input signal. Not all floppy disk drives provide this signal.

The output signals of this register are all forced to 0's on powerup or reset. Output bits 0-4 and 7 drive signals on the floppy disk drive interface connector.

The 8-INCH SELECT signal allows the 1772 FDC device to send and receive data at 8-inch drive data rates, by doubling the device's master clock input frequency. However, the 1772 is currently not guaranteed to fuction properly at this doubled clock rate, and should therefore only be used in this manner with caution. A software-controlled RESET signal has been provided to assure that the 1772 does not become hung up when its master input clock is changed during operation. It is recommended that a FORCED INTERRUPT command be issued to the 1772 immediately following a change in its clock rate, and that the software-controlled RESET only be used when an error is detected, and as part of the 1772 initialization sequence following system powerup or reset.

4.7 SCSI/PLUS INTERFACE

The SCSI interface is completely controlled by the NCR 5380 protocol controller device. The 5380 provides an interface which meets the ANSI specification for SCSI, including Initiator role, Target role, Arbitration, and the Disconnect-Reselect function. It also supports the Initiator role of the AMPRO-proposed SCSI/PLUS enhancement to SCSI. The 5380 SCSI controller allows full programmable control of 17 bi-directional bus signals, and provides both buffered (low leakage) bus inputs and high current (48 mA) bus output drive capacity. The 5380 is described thoroughly in its data sheet, which appears in Appendix D.

The interrupt output of the 5380 connects to the 80186's INT1 interrupt input. The 5380's DMA request output signal is selectively connected, under control of the FDC Control Register, to the 80186 DMA request 0 input.

4.7.1 ID Input Register

An ID Input Register is used to read the state of eight sets of board jumpers. The eight bits are read in the lower half of the 80186 data bus, and represent the jumper settings as follows: when a jumper is installed, the bit will be read as a 0 (low), and when a jumper is not installed, the bit will be read as a 1 (high).

The AMPRO ROM-BIOS and AMPRO PC-DOS support drivers use these eight jumpers to determine various powerup defaults associated with the floppy and SCSI interfaces.

This port can also be used as a general purpose 8-bit input port in applications not using the AMPRO ROM-BIOS. In this case, the physical board layout of the eight pairs of jumper pins permits a 16-pin flat ribbon cable connector to be plugged directly onto them.

CHAPTER 5

PROGRAMMER'S REFERENCE

5.1 INTRODUCTION

This chapter contains information useful to programmers who intend to program Little Board/186 hardware directly. Normally PC-DOS handles the hardware for you, so that you need not be concerned with direct programming of the board's hardware -- that is the advantage of using a standard operating system. Since the AMPRO ROM-BIOS provides a high degree of compatibility with the IBM PC ROM-BIOS, there should be little need for custom programming.

The hardware theory of operation and interconnection is covered extensively in Chapter 4, whereas this chapter concentrates on details which directly relate to programming of the board's devices and I/O ports. Please read Chapter 4 before attempting to utilize the information provided in this chapter. For additional information about the board's LSI devices (80186, 2681, 1772, and 5380), refer to the data sheets in Appendix D, and to additional documentation available from the device manufacturers.

5.2 80186 UTILIZATION AND INITIALIZATION

Since the 80186 microprocessor has many programmable hardware functions which assume default values under control of the Little Board/186 ROM-BIOS, much of the information in this chapter is based on these defaults. Should you choose not to use the ROM-BIOS, these defaults (I/O port addresses, etc.) become your responsibility.

The 80186 contains a large number of internal read/write registers, called the "Peripheral Control Block," which is used for programming of the 80186's many internal functions, including the interrupt controller, DMA controller, and Counter/Timing Controller (CTC). The ROM-BIOS leaves the Peripheral Control Block at its power-up default I/O base address, FF00H, so that the registers within the Peripheral Control Block occupy I/O addresses FF20H through FFFEH. You can determine each individual register's address by adding the specific register's address offset, shown in the 80186 data sheet, to the Peripheral Control Block's base address, FF00H. For example: the UMCS Register has offset A0H, so its I/O address is FFA0H; the DMA Channel 0 Transfer Count Register has offset C8H, so its I/O address is FFC8H; etc.

5.2.1 Memory Map

The two byte-wide EPROM sockets are selected by the Upper Memory Chip Select (UMCS) output from the 80186. The 80186 UMCS is programmed on cold reset before jumping to the main task. There is enough code space at FFFF0H (the reset jump address) to program the EPROM size before jumping. The jump must be a long intersegment jump to initialize the code segment register. If a larger EPROM is to be used, the code at FFFF0H must be modified appropriately, to initialize UMCS for the required address space.

The ROM-BIOS programs UMCS for zero wait states, and ignoring of external ready. The base address of the EPROM is FC000H, to accommodate a pair of 2764

EPROM's, providing 16K of EPROM space.

The 80186 Lower Memory Chip Select (LMCS) and Middle Memory Chip Select (MMCS) lines are unused, and unprogrammed by the ROM-BIOS. This allows the lower and middle memory chip selects to be used for RAM wait state generation -- if needed -- though wait states should never be required with 150 nS, or faster, RAM. Due to the block size that these chip selects can span, they must be programmed wisely, if used.

On-board 512K or 128K RAM memory chip selection is decoded directly from the 80186 address lines, and is independent of the programming of the MMCS and LMCS lines. The 512K RAM expansion unit available from AMPRO also decodes its address space directly from the 80186 rather than using MMCS or LMCS. The 512K RAM expansion unit automatically deselects when UMCS is active, so that EPROM access overrides that of RAM.

5.2.2 I/O Map

The board's I/O device addresses are programmed by the ROM-BIOS to occupy I/O addresses from 1000H through 137FH, as shown in Table 5-1. Chip selects for the various I/O devices on the board are derived from the 80186 Peripheral Chip Selects (PCS), which are utilized as shown in Table 5-2. PCS6 is reserved for future use, such as selection of I/O devices on a plug-in daughter board. Wait state values shown in Table 5-2 are the defaults programmed by the ROM-BIOS.

Table 5-1. Summary of I/O Ports

Address	Input/Output	Function
1000H -101EH	I/O	2681 Internal Registers (See Table 5-4)
1080H -108FH	I/O	5380 Internal Registers (See Table 5-8)
1100H	I/O	1772 Command/Status Register
1102H	I/O	1772 Track Register
1104H	I/O	1772 Sector Register
1106H	I/O	1772 Data Register
1180H	I/O	5380 DMA Acknowledge
1200H	O	FDC Control Register
1280H	O	Printer Data Register
1300H	I/O	80186 PCS6 (currently unused)
FF20H -FFDAH	I/O	80186 Peripheral Control Block

Table 5-2. 80186 Peripheral Chip Select Usage.

PCSX	Read/Write	Wait States	Usage
PCS0	R/W	1	2681 chip select
PCS1	R/W	1	5380 chip select
PCS2	R/W	1	1772 chip select
PCS3	R/W	1	5380 DACK pin
PCS4	W	0	FDC Control Reg. data strobe
PCS5	W	0	Printer port data strobe
PCS6	R/W	0	Reserved for future use

5.2.3 Interrupts

The 80186 internal and external hardware interrupts are used as shown in Table 5-3. The table shows both internal interrupts from the timers and DMA channels and external interrupts from the four interrupt pins (INTRX). All of the external interrupts are edge sensitive, and are triggered on a low-to-high transition.

Table 5-3. 80186 Interrupt Usage

Priority (if used)	80186 Function	Vector "Type"	Usage
		Internal Interrupts	
0	Timer	08H	Real time clock (18.2/sec)
n/a	Timer 1	12H	Not used
n/a	Timer 2	13H	Refresh timer (16 usec/tick)
n/a	DMA 0	0AH	SCSI EOP
n/a	DMA 1	0BH	Refresh DMA EOP
		External Interrupts	
1	INTR 0	0CH	2681 interrupt
2	INTR 1	0DH	5380 interrupt
3	INTR 2	0EH	1772 interrupt
7	INTR 3	0FH	Printer interrupt

The ROM-BIOS uses each external hardware interrupt channel as follows:

Interrupt Channel 0 - 2681 Serial Controller Interrupt

Vector: 0CH
Priority: 1

This vector corresponds to that of the COM1 IBM PC device. The 2681 is used to implement the console port, a spare counter-timer channel, and miscellaneous I/O. Receive and transmit interrupts, break detect, counter timeout, and port bit input change interrupts are all possible. An interrupt vector table is provided by the ROM-BIOS to sort out the specific interrupt condition.

Interrupt Channel 1 - 5380 SCSI Controller Interrupt

Vector: 0DH
Priority: 2

This vector indicates an interrupt condition from the 5380 SCSI controller device. The 5380 interrupt occurs from a variety of bus events.

Interrupt Channel 2 - 1772 Floppy Disk Interface Controller

Vector: 0EH
Priority: 3

This vector is normally used to monitor the completion of a disk command when using DMA for floppy data transfers. It corresponds to that of the IBM PC.

Interrupt Channel 3 - Printer Interface Interrupt

Vector: 0FH
Priority: 7

This interrupt is triggered when the Centronics interface PRINTER ACKNOWLEDGE signal goes from inactive to active. It corresponds to the IBM PC printer interrupt vector.

5.2.4 DMA

The 80186 internal 2-channel DMA controller is normally used as follows:

DMA Channel 0 - SCSI/FDC DMA Data Read/Write

This channel is shared by the 1772 floppy disk controller and the 5380 SCSI interface controller. The contents of the FDC Control Register bit 6 determines which interface drives the 80186 DMA channel 0 transfer request input. Sharing of this DMA function between the floppy and SCSI interfaces has minimal impact since most software only performs one mass storage transfer operation at a time. Since SCSI data transfers are asynchronous, regulated by REQ/ACK handshaking, they could be done under programmed I/O if an application calls for interleaved transfer of SCSI

and FDC data.

The 1772 DMA acknowledge is a read or write of the 1772 data register, while the 5380 has a dedicated DMA acknowledge pin which is controlled by a dedicated 80186 Peripheral Chip Select. Maximum transfer speed is 2 megabytes per second.

DMA Channel 1 - Dynamic RAM Refresh Control

This channel controls dynamic RAM refresh. It is programmed for a memory-to-I/O transfer, with writes to nonexistant I/O space. A DMA cycle is requested by Timer Channel 2 every 16 microseconds. The DMA channel is programmed to continuously run through the entire megabyte of memory address space, with automatic rollover and no terminal count. It is set for word transfers (16-bit), so that the address increments by two each cycle (ADR1 is the LSB address input to the DRAM'S). This method of RAM refresh uses approximately 6 percent of the 80186 processing bandwidth. Since there is no terminal count interrupt service requirement, DRAM refresh continues uninterrupted even during 80186 halt states. This DMA channel should always be given the highest priority.

5.2.5 Counter/Timers

The 80186 internal 3-channel counter/timer controller (CTC) provides three interrupt sources. CTC Channel 0's interrupt is normally used to provide a real time clock tick, and interrupts from CTC Channels 1 and 2 are normally disabled. Channel 2 is used for regulating the time period of the DMA-controlled RAM refresh, and no software maintenance is required once that function is initialized. Channels 1 and 2 should not allowed to generate interupts, unless they can distinguish between a software call and hardware interrupt service request. (The 80186 interrupt request pending register can be read, to determine this.)

The ROM-BIOS uses the three CTC channels as follows:

CTC Channel 0 - Real Time Clock Tick

This channel is used for the real time clock tick, and may be programmed to simulate the IBM PC interrupt service routine. The required real time clock interrupt rate for compatibility with the IBM PC is 18.2 interrupts per second. In order to achieve this rate, CTC Channel 2 must be used as a prescaler.

This channel's interrupt vector, as used in the AMPRO ROM-BIOS, overlaps the IBM PC timer interrupt. (The ROM-BIOS calls INT 1CH.) The interrupt control register is programmed for non-nested mode and priority of 1; this timer's interrupt mask bit is cleared.

CTC Channel 1 - Not used

This channel is not used by the ROM-BIOS, and can be used to generate timing required by software functions. This channel is also cascaded with the 2681 counter/timer output, and can be used in conjunction with the 2681 for generating very accurate time intervals.

CTC Channel 2 - DRAM Refresh Rate

This channel is used to request DMA cycles for DRAM refresh. It is programmed for a 16 microsecond rate. The output of this channel is internally routed to Channel 0, which further divides the rate to obtain the real time clock tick.

5.3 SERIAL CONTROLLER

A 2681 Dual Asynchrounous Receiver/Transmitter (DUART) device provides two serial ports, a counter/timer, 7 signal input pins, and 8 signal output pins. There is one common interrupt to the 80186 for all functions. Multiple interrupts require a dispatch routine. This section defines the usage of the various portions of the 2681 device.

The 2681 data I/O lines (D0-D7) connect to the lower byte (D0-D7) of the system internal data bus. The 2681's internal registers occupy a block of 32 I/O addresses, beginning at the default base address of the 2681 programmed by the ROM-BIOS, 1000H. The A0-A3 address inputs to the 2681 connect to system address lines A1-A4. The 2681's internal register addresses are shown in Table 5-4.

5.3.1 Serial Port Interface

The board's two serial ports are implemented directly by the DUART's two serial channels. As indicated in the 2681 data sheet, the device's two channels can be programmed for data characteristics, baud rate, and optional RTS/CTS handshaking.

Transmit and receive baud rates for each channel are independently programmable, from 50 to 38.4K baud. Each channel's receiver has a four byte FIFO, with status flags for FIFO full and ready. Each channel can be programmed to wait until the FIFO is full before interrupting. This reduces CPU overhead when handling high baud rates. Each channel's transmitter has a holding register and a shift register, and provides both empty and ready status flags.

Table 5-4. 2681 Internal Registers

Address	Input/Output	Function
1000H	Input	Mode Register A (MR1A, MR2A)
1002H	Input	Status Register A (SRA)
1004H	Input	(Reserved)
1006H	Input	RX Holding Register A (RHRA)
1008H	Input	Input Port Change Reg. (IPCR)
100AH	Input	Interrupt Status Register (ISR)
100CH	Input	Counter/Timer Upper (CTU)
100EH	Input	Counter/Timer Lower (CTL)
1010H	Input	Mode Register B (MR1B, MR2B)
1012H	Input	Status Register B (SRB)
1014H	Input	(Reserved)
1016H	Input	RX Holding Register B (RHRA)
1018H	Input	(Reserved)
101AH	Input	Input Port
101CH	Input	Start Counter Command
101EH	Input	Stop Counter Command
1000H	Output	Mode Register A (MR1A, MR2A)
1002H	Output	Clock Select Register A (CSRA)
1004H	Output	Command Register A (MRA)
1006H	Output	TX Holding Register A (THRA)
1008H	Output	Aux. Control Register (ACR)
100AH	Output	Interrupt Mask Register (IMR)
100CH	Output	C/T Upper Register (CTUR)
100EH	Output	C/T Lower Register (CTLR)
1010H	Output	Mode Register B (MR1B, MR2B)
1012H	Output	Clock Select Register B (CSRB)
1014H	Output	Command Register B (CRB)
1016H	Output	TX Holding Register B (THRA)
1018H	Output	(Reserved)
101AH	Output	Output Port Configuration Reg (OPCR)
101CH	Output	Set Output Port Bits Command
101EH	Output	Reset Output Port Bits Command

5.3.2 Auxiliary Signals

The 7 input and 8 output general purpose I/O bits provided by the 2681 are used to provide control and status signals for a variety of Little Board/186 hardware functions. The 2681's Output Control Register is used to program the mode of these 15 I/O pins; the Output Port Register is used to the set the state of the output pins; the Input Port is used to read the state of the input pins.

The programmable output pins of the 2681 are set or reset according to values programmed in the 2681 Output Port Register. Bits of the Output Port Register can be individually set and reset. A bit is set by performing a write operation to one I/O port address, and reset by performing a write operation to a second I/O address. NOTE: The actual OP0-OP6 output pin values are the complements of the Output Port Register contents. Refer to the 2681 component

data sheet for further details.

The seven 2681 general purpose inputs are utilized as shown in Table 5-5; signal outputs provided by the 2681's general purpose outputs are shown in Table 5-6.

Table 5-5. 2681 General Purpose Input Signals

Bit 6	5	4	3	2	1	0
LOOP	PAPER	SEL	-RDY	BSY	HSIB	HSIA

Bit	Signal/Function
IP6	**LOOP** - Looped from output signal OP6; also reads the status of the diagnostic jumper, JMP12. If JMP12 is shorted, IP6 will be a 0 regardless of state of OP6.
IP5	**-PAPER** - Indicates the state of the OUT OF PAPER signal input from the printer interface (J2 pin 23). Inverted relative to the level on the connector pin. When the printer is out of paper this bit is a 0.
IP4	**-SELECTED** - Indicates the state of the SELECTED signal input from the printer interface (J2 pin 25). Inverted relative to the level on the connector pin. When the printer is on-line (SELECTED) this bit is a 0.
IP3	**DRIVE RDY** - Indicates the state of the floppy disk interface -READY signal (J6 pin 34). The signal is inverted relative to the connector pin. When this bit is a 1, the floppy media is up to speed.
IP2	**PRINTER BUSY** - Indicates the state of the BUSY signal input from the printer interface (J2 pin 21). Directly reflects the level on the connector pin. When the printer is busy, this bit is a 1.
IP1	**HSIB** - Indicates the state of Serial Port B Handshake In (HSI) signal (J4 pin 6). When Hand Shake In is active from the RS232 interface, this bit is a 1. Normally used as Channel B CTS.
IP0	**HSIA** - Indicates the state of Serial Port A Handshake In (HSI) signal (J3 pin 6). When Hand Shake In is active from the RS232 interface, this bit is a 1. Normally used as Channel A CTS.

Table 5-6. 2681 General Purpose Output Signals.

Bit 7	6	5	4	3	2	1	0
INIT	LOOP	-FDRST		TIMER	-STB	HSOB	HSOA

Bit	Signal/Function
OP7	**INITIALIZE** - Initializes the printer, when 1. The printer interface -INIT signal (J2 pin 26) is inverted relative to this pin of the 2681.
OP6	**LOOP** - Looped to input signal IP6. Has no effect when jumper JMP12 is shorted.
OP5	**-FDRST** - Hardware reset signal to the 1772 FDC. When this bit is a 0, the 1772 is held reset. (NOTE: The 1772 requires a minimum reset pulse width of 50 microseconds.)
OP4	Not used.
OP3	**TIMER** - Connected to 80186 Timer Input 1, for use as a prescaler input to 80186 CTC Channel 1.
OP2	**-STROBE** - Controls the printer interface -DATA STROBE signal (J2 pin 1). When this bit is a 0, -DATA STROBE is in its active low state.
OP1	**HSOB** - Sets the state of Serial Port B Handshake Out (HSO) output signal (J4 pin 4). When this bit is a 1, HSOB is active. Normally used as Channel B RTS.
OP0	**HSOA** - Sets the state of Serial Port A Handshake Out (HSO) output signal (J3 pin 4). When this bit is a 1, HSOA is active. Normally used as Channel A RTS.

NOTE

The 2681's output pins (OP0-7) are the complement of the contents of the Output Port Register. That is, setting a bit in the output port register results in a low (0) logic level at the corresponding output pin, and resetting a bit results in a high (1) level. The input pins (IP0-6), however, are not complemented. Therefore, the bits of the Input Port directly reflect the state of the corresponding pins.

5.3.3 Use of the Handshake In Signal

As indicated in Table 5-5, each serial channel has ony a single HANDSHAKE IN signal, connected to the respective channel's CTS pin on the 2681 serial controller. However, in most modem applications Data Carrier Detect (DCD) is the important input status signal. In this case, the HANDSHAKE IN signal can be connected to the modem's DCD signal instead of CTS; the software must

interpret the signal as DCD rather CTS.

If desired, the board's ROM-BIOS functions (INT14) can be used to transfer data and read the state of the DCD status signal. Again, "DCD" is not directly supported, but must be sensed through the "CTS" status bit supplied by the INT14 status function, since that bit reflects the state of HANDSHAKE IN status line.

5.4 Parallel Printer Interface

The Centronics printer interface consists of eight output data lines, two output control signals, and four input status signals. The data output lines are generated by the Printer Data Register. The output bits of the Printer Data Register drive the printer interface connector pins directly. The connector signals directly reflect the bit values written to the data register.

The -ACK signal from the printer interface connector (J2 pin 19) is used to provide an interrupt input to the 80186. The sense of interrupt sensing is such that an interrupt occurs when the -ACK signal makes a transition from 0 to 1, or active to inactive.

In addition, two output and three input signals associated with the printer interface are supported by the 2681 serial controller's general purpose I/O pins, as shown in Tables 5-5 and 5-6. Three of these signals are inverted relative to the printer interface connector (J2), as shown in the following summary of the five printer control and status signals:

INIT - Generated by 2681 output OP7. When this pin of the 2681 is a 1, the printer's internal logic is initialized. This output of the 2681 drives J2 pin 26 through an inverting buffer.

-DATA STROBE - Generated by 2681 output OP2. When active, signals the printer to accept (and print) data. This output of the 2681 drives J2 pin 1 through a non-inverting buffer.

-OUT OF PAPER - Sensed by 2681 input IP5. When 0, indicates that the printer is out of paper. The signal at 2681 input IP5 is inverted relative to the signal level at J2 pin 23.

-SELECTED - Sensed by 2681 input IP4. When 0, indicates that the printer is selected. The signal at 2681 input IP4 is inverted relative to the signal level at J2 pin 25.

BUSY - Sensed by 2681 input IP2. When 1, indicates that the printer is busy. The signal at 2681 input IP2 corresponds directly to the signal level at J2 pin 21.

5.5 Floppy Disk Interface

Most of the floppy disk interface support is provided by the 1772 Floppy Disk Controller (FDC) device. In addition, the FDC Control Register controls a number of signals associated with floppy disk interface functions.

Due to the complexity of programming of the floppy disk interface, we recommend that you utilize the standard functions available through PC-DOS function calls and ROM-BIOS interrupts rather than attempting to program this interface yourself.

The 1772 device is programmed as indicated in its component data sheet. For a variety of reasons beyond the scope of this document, the ROM-BIOS performs several functions in software which the 1772 data sheet claims the 1772 can do automatically. These are: motor on delay, head load delay, seek verification.

The 1772's data I/O lines (D0-D7) connect to the lower byte (D0-D7) of the system internal data bus. The 1772's internal registers occupy a block of 8 I/O addresses, beginning at the default base address of the 1772 programmed by the ROM-BIOS, 1100H. The A0 and A1 address inputs to the 1772 connect to system address lines A1 and A2, respectively. The 1772's internal register I/O port addresses are shown in Table 5-1.

The 1772 master reset pin is driven by one of the general purpose output bits of the 2681 serial controller, OP5. This allows the 1772 to be reset under software control. OP5 connects directly to the 1772's -RESET pin (pin 13), so that the 1772 is held in a reset state when 2681 output OP5 is a 0.

Pin 34 of the floppy disk drive interface (J6) connects to the 2681 serial controller's general purpose input IP3 through an inverting buffer. This signal is often used as DRIVE READY status signal, but varies among different drive manufacturers. The ROM-BIOS does not use this signal; software time delays (for head load and motor on) are used instead.

5.5.1 FDC Control Register

The FDC Control Register provides a number of important functions. The output bits of the FDC Control Register directly reflect the contents of the byte of data written to the register by the 80186 CPU. The functions of the eight output bits of this register are shown in Table 5-7.

Table 5-7. FDC Control Register Programming.

Bit 7	6	5	4	3	2	1	0
FDCLK	DMASEL	-DDEN	SIDE1	DS4	DS3	DS2	DS1

Bit	Signal/Function
Bit 7	**FDC Clock** - Selects either 8 mHz (when 0) or 16 mHz (when 1) as the clock input frequency to the 1772 FDC device. NOTE: Observe the precautions indicated in Chapter 5.
Bit 6	**DMA Select** - Switches the DMA request input to 80186 DMA channel 0 between the 1772 FDC (when 0) and the 5380 SCSI controller (when 1). Please observe the precaution indicated in Chapter 5.
Bit 5	**DDEN** - Double density enable. When 0, places the 1772 FDC in double density mode. When 1, enables single density.
Bit 4	**SIDE1** - Side one select. When 1, selects floppy disk drive side one. When 0, selects side zero.
Bits 0-3	**DS1-DS4** - Floppy disk drive selects. When the bit is 1, selects the corresponding floppy disk drive. Only one of these bits should active (1) at a time.

Here is a brief description of the FDC Control Register bit functions:

FDC Clock - Selects either 8 mHz (when low) or 16 mHz (when high) as the clock input frequency to the 1772 FDC device. Please observe the following precautions in programming this bit:

The 8-INCH SELECT signal allows the 1772 FDC device to send and receive data at 8-inch drive data rates, by doubling the device's master clock input frequency. However, the 1772 is currently not guaranteed to function properly at this doubled clock rate, and should therefore only be used in this manner with caution. A software-controlled RESET signal has been provided in case the 1772 hangs up when its master input clock is changed during operation.

It is recommended that a FORCED INTERRUPT command be issued to the 1772 immediately following a change in its clock rate, that the 1772's internal registers be re-written following clock changes, and that the software-controlled RESET only be used when an error is detected, and as part of the 1772 initialization sequence following system powerup or reset. It is further recommended that the state of this bit never be changed while any drive selects are active (i.e.: deselect all drives, change clock rate, reselect desired drive, delay appropriate head load delay).

DMA Select - Switches the DMA request input to 80186 DMA channel 0 between the 1772 FDC (when 0) and the 5380 SCSI controller (when 1). Please observe the following precaution in programming this bit:

WARNING

Never change the state of this bit while DMA channel 0 is enabled.

DDEN - Double density enable. When 0, places the 1772 FDC in double density mode. When 1, enables single density.

SIDE1 - Side one select. When 1, selects floppy disk drive side one.

DS1-DS4 - Floppy disk drive selects. When the bit is 1, selects the corresponding floppy disk drive. Only one of these bits should be active at a time.

5.6 ID INPUT PORT

This port can either be used for SCSI bus ID, for general purpose jumper settings, or as an 8-bit general purpose data input port.

The ID input port is read by an 80186 I/O input instruction. The jumpering of the eight pairs of pins at location J7 on the board determines the data byte obtained. The input buffer is non-inverting: the data read directly reflects the level on the input pin. When a jumper is inserted, the corresponding data is low (0); when out, the data bit is high (1).

Jumper assignment is as follows: J7 pins 1 and 2 corresponds to data bit 0; pins 3 and 4 are data bit 1; ...; pins 15 and 16 are data bit 7.

The ROM-BIOS default uses for these jumpers is discussed in Chapter 2, Board Jumper Configurations.

5.7 SCSI/PLUS INTERFACE

The SCSI/PLUS interface is controlled by means of an NCR 5380 SCSI Protocol Controller device. The 5380 contains 8 readable and 8 writable registers. These are accessed by 80186 I/O input and output instructions.

The 5380's data I/O lines (D0-D7) connect to the lower byte (D0-D7) of the system internal data bus. The 5380's internal registers and functions occupy a block of 16 I/O addresses, beginning at the default base address of the 5380 programmed by the ROM-BIOS, 1080H. The A0-A2 address inputs to the 5380 connect to system address lines A1-A3. The 5380's internal registers are shown in Table 5-8.

Table 5-8. 5380 Internal Registers.

Address	Input/Output	Function
1080H	Input	Current SCSI Data
1082H	Input	Initiator Command Register
1084H	Input	Mode Register
1086H	Input	Target Command Register
1088H	Input	Current SCSI Bus Status
108AH	Input	Bus & Status Register
108CH	Input	Input Data Register
108EH	Input	Reset Parity/Interrupt Command
1080H	Output	Output Data Register
1082H	Output	Initiator Command Register
1084H	Output	Mode Register
1086H	Output	Target Command Register
1088H	Output	Select Enable Register
108AH	Output	Start DMA Send Command
108CH	Output	Start DMA Target Receive Command
108EH	Output	Start DMA Initiator Receive

The SCSI/PLUS interface has a wide variety of applications, including:

- Use with SCSI (SASI) disk controllers and devices
- Use with the AMPRO SCSI/IOP for data acquisition and control
- Use as a bidirectional I/O port
- Use as a multi-master network bus

The AMPRO PC-DOS Support Software diskette provides SCSI support for "generic SCSI" hard disk controllers and drives. If you plan to use SCSI (SASI) devices not supported by the standard AMPRO drivers and utilities, you can either create a custom SCSI driver, modify the AMPRO drivers and utilities, or provide SCSI interface control within your program.

If you plan to program the 5380 yourself, you will probably require a copy of the NCR 5380 SCSI Interface Chip Design Manual, available through AMPRO for a nominal charge.

Copies of the ANSI X3T9.2 SCSI specification may be obtained by sending $20 and a self-addressed mailing label (for each copy desired) to:

The X3 Secretariat
Computer and Business Equipment Manufacturers Association
311 First Street, N.W. - Suite 500
Washington, DC 20001

The SCSI/PLUS Preliminary Technical Specification, which details AMPRO's proposed enhancement to SCSI to allow 64 (rather than 8) bus devices, is available through AMPRO for a nominal charge.

5.7.1 SCSI (SASI) Programming

When using the SCSI/PLUS interface with SCSI (SASI) disk controllers, special programming is not generally required; the AMPRO ROM-BIOS, and PC-DOS Support Software drivers and utilities accommodate many types of disk controllers and disk drives. Installation of the hard disk software is all that is generally required, providing you are using controller and drive types supported. (Refer to the AMPRO PC-DOS Support Software User's Manual.)

When using the 5380 in SCSI (SASI) applications, care must be taken to meet the specified timing constraints. For detailed timing information, consult your peripheral controller's technical manual, or the SCSI specification referenced above.

5.7.2 Simple Bidirectional I/O

If you plan to program the 5380 yourself, you will need a copy of the NCR 5380 design manual mentioned above. The 5380 has 17 bidirectional I/O lines, which may be used as inputs or outputs under software control.

The 5380 has two operating modes: Initiator and Target modes. In Initiator mode, several conditions are required before data output to the I/O bus can be active. If the device is used in the Target mode, however, these special conditions are not applicable. This results in more straight forward programming of simple I/O applications, and is recommended for simple bidirectional I/O.

The 5380 is placed in Target mode by writing 40h to the Mode Register. Once in Target mode, all 17 I/O signals except ACK and ATN may be used as both inputs and outputs. In Target mode, ACK and ATN are inputs only. The data lines (DB0-7,P) are outputs when bit 0 ("Assert Data Bus") of the Initiator Command Register is a 1, and inputs when bit 0 of that register is a 0.

Eight additional inputs are available via the ID Input Port, discussed above. Also, the parallel printer port can also provide an additional set of eight outputs and five handshake signals, if it not required as a printer interface.

5.8 ROM-BIOS INTERRUPTS

This section provides information on the software interrupts provided by the Little Board/186 ROM-BIOS. Compatibility with the IBM PC and PC-AT ROM-BIOS software interrupt calling conventions has been maintained wherever possible.

INT 10H - Video Display

The video display BIOS call is used to send characters to the system console device connected to the 2681 serial controller's channel A. The Little Board/186 supports only the "write tty" function. All other INT 10H function calls do nothing. This allows maximum RS232 ASCII terminal flexibility.

Support of the remaining functions is left to a terminal device driver, which allows various terminals to be controlled using BIOS calls. The

required device driver installs itself over the INT 10 vector and fields the BIOS calls. (See AMPRO PC-DOS Support Software User's Manual.)

If the 2681 automatic hardware handshaking has been enabled, and the 2681 cannot accept an output character, then the driver will hang up until the character has been sent.

Usage protocal:

AH = 0EH, write character in 'AL' to screen
AL = character

All registers preserved

Other INT 10H functions return to caller with no action taken.

INT 11H - Equipment Check

This BIOS call returns the value in the equipment flag stored in RAM at same location as that used by the IBM ROM-BIOS. This is a ROM constant, and always indicates: 1 RS232 device, 0 printers, 64K planar ram, bootable, 80 x 25 monochrome display, and the number of floppies determined by the jumpering of J7 pin pairs 1/2 and 3/4 (see Table 2-9).

Usage protocol:

Inputs: none

Output: AX contains the equipment flag, as described above.

INT 12H - Determine Memory Size

This BIOS call returns the total RAM size that is determined by the ROM-BIOS on powerup or reset. The result is stored in the same location used by the IBM ROM-BIOS.

Usage protocol:

Inputs: none

Output: AX = number of contiguous 1K blocks of memory

INT 13H - Hard Disk I/O

If a 5380 SCSI controller is present on the board, the hard disk support is installed at interrupt 13H location by the ROM BIOS, first relocating the floppy interrupt to INT 40H, and then replacing the address at INT 13H.

Usage protocol:

function	register
disk reset	ah, = 0
return disk status	ah, = 1
read sector(s)	ah, = 2
write sector(s)	ah, = 3
verify sector	ah, = 4
format track	ah, = 5, note 3
flag bad track	ah, = 6, note 2
format drive at track	ah, = 7, note 2
return drive parameters	ah, = 8
set drive parameters	ah, = 9, note 1
read with/ecc	ah, = A, note 1
write with/ecc	ah, = B, note 1
seek	ah, = C
reset disk controller	ah, = D, note 1
read controller buffer	ah, = E, note 2
write controller buffer	ah, = F, note 2
test drive ready	ah, = 10
recalibrate	ah, = 11
controller ram diagnostics	ah, = 12, note 2
disk drive diagnostics	ah, = 13, note 2
disk controller diagnostics	ah, = 14, note 1
read dasd	ah, = 15

Parameters passed in the following registers:

dl, = drive number, 0 to 7
dh, = head, 0 or 7
ch, = track number, 0 to max track
cl, = sector number, 0 to 17, bits 6,7 = msb of track
al, = number of sectors to be transfered
es:bx = pointer, segment in es:, offset in bx:

Returns:

good status	ah, = 0, sucessful read
	al, = number of sectors read
	cf, = 0, sucessful read
bad status	ah, = status
	cf, = 1, unsucessful read

Notes: (1) Returns good status
(2) Returns bad status
(3) Does not format track, HFORMAT is used to format.

INT 14H - RS232 I/O

This BIOS call controls serial channel B of the 2681 and expects to have a non-zero port address in the RAM data area at 40:0, just as in the IBM ROM-BIOS.

The serial port address at 40:0 is not used. The port number received in DX must be 0; if DX has a number greater than zero the call returns a time out. The initialization function sets the same baud rates, parity and number of bits as does the IBM ROM-BIOS.

The send and receive functions are identical to those of the IBM ROM-BIOS. Line status (returned in AH) is identical to that returned by the corresponding IBM ROM-BIOS functions, as the same status information is available from the 2681. The status call, however, differs in AL. Only status bits DSR and CTS are implemented, with DSR always a 1 and CTS reflecting the signal input to the 2681 from the hand shake in signal on serial port B. The ring indicator, receive line signal detect, and the Delta signals are not supported and are always returned as 0. This is because the 2681 does not offer support for these signals. The CTS signal can be used for a Data Carrier Detect (DCD) sensing by modem software, provided that the cable between the board and the modem has appropriate wiring.

Usage protocol:

```
DX = port number (must be 0)
AH = 0  initialize serial port B, return status in ah
AL = init value, as follows:
     bits 7,6,5 - baud rate:  110, 150, 300, 600, 1200, 2400,
                                4800, or 9600
     bits 4,3 - parity:       none, odd, even
     bit  2 - stop bits:      1 or 2
     bits 1,0 - bits/char:    7 or 8
AH = 1  sends the character in AL, returns status in AH
AH = 2  returns receive character in AL, returns status in AH
AH = 3  returns status in AX
```

INT 15H - Cassette I/O

This is a null function and returns a timeout.

INT 16H - Keyboard I/O

Interrupt 16 returns data input from the 2681 serial controller's channel A.

This ROM-BIOS call is identical to that of the IBM ROM-BIOS, with the following two exceptions:

(1) Characters are buffered in the same ring buffer space but no scan codes are stored. Consequently, there is twice the key buffering capacity for keystrokes.

(2) A scan code of zero is returned in AH, always. Keyboard shift status is always returned as caps active (40H), and it is stored at the same RAM location as in the IBM ROM-BIOS.

Usage protocol:

AH = 0 Returns the next character input from the console. Result in AL, scan code (always 0) in AH.

AH = 1 Sets Z flag to indicate if character available, as follows:

ZF = 1 Key not available
ZF = 0 Key "down"

When ZF = 0, the next character in the buffer to be read is in AX, and entry remains in the buffer.

AH = 2 Returns keyboard shift status in AL, as caps active (40H).

INT 17H - Printer I/O

The printer I/O BIOS vector supports the DOS LPT1 device (DX = 0) only, mapping it to the the board's Centronics printer port. The protocol is has a few minor differences from that of the IBM ROM-BIOS. The print character timeout is a fixed constant and not in RAM as the IBM XT and later use. The printer status returned supports all bits except the hardware I/O error bit, which has no cable input connection. If DX is non-zero on input, a timeout error results.

Usage protocol:

AH = 0 Sends character in AL to printer port, returns status in AH
AH = 1 Initializes printer, returns status in AH
AH = 2 Returns printer status in AH

Where: DX = port value. Must be zero or a timeout results.

Outputs: Same status returned as in IBM ROM-BIOS, except I/O error bit. All registers except AH are preserved.

INT 18H - Resident BASIC

The ROM basic interrupt points to the cold boot, as there is no ROM BASIC.

INT 19H - Boot Strap Loader

INT 19H, the boot strap loader interrupt, tries 3 times to read the boot sector from floppy drive 0 (A). After the 3 floppy tries INT 19H will attempt to read the boot sector from the hard disk system. During this sequence if any read is sucessful, the sector read will be tested for validity. If valid, a jump to the boot sector code is made.

NOTE: If no NCR 5380 is installed the hard disk interrupt is not installed during system initialization.

INT 1AH - Get/Set Time of Day

Interrupt 1AH is used to get or set the BIOS real time clock. The following parameters are passed.

function	register
get tod	ah, = 0
returns	current time of day cx, = high count dx, = low count al, = rollover count value
set tod	ah, = 01 cx, = high count dx, = low count

INT 1BH - Keyboard Break

The Keyboard break interrupt vector is neither called nor supported in the ROM-BIOS. This is left to the terminal drivers.

INT 1CH - Timer Tick Interrupt

This interrupt is not used by the ROM BIOS and points to an IRET. It is called during each tick of the timer. (Tick INT 8H, 18 per/sec)

Application programs that need servicing at regular intervals patch this interrupt for their own status entry routines.

INT 1DH - Video Initialization

Not used.

INT 1EH - Diskette Parameters

The diskette I/O does not use the disk parameter table pointer. Also, the RAM storage location for motor turnoff delay (MOTOR_WAIT) is not used, as the 1772 handles this automatically.

INT 1FH - Video Graphics Characters

Not used.

INT 40H - Floppy I/O

The floppy interrupt is installed at INT 40H if the NCR 5380 SCSI interface is installed. If the 5380 is not installed the floppy I/O is interrupt 13H.

Usage protocol:

function	register
disk reset	ah, = 0
disk status	ah, = 1, return last disk status
read	ah, = 2
write	ah, = 3
verify	ah, = 4, no data transfered
format track	ah, = 5

For functions 2 thru 4 the following registers are used:

dl, = drive number, 0 to 3
dh, = head, 0 or 1
ch, = track number, 0 to 79
cl, = sector number, 0 to 8, or 0 to 9
al, = number of sectors to be transfered
es:bx = pointer, segment in es:, offset in bx:

For function 5, format track, ES:BX points to a table of sector headers (1 header per sector) with the following format:

1. track number
2. head number (side)
3. sector number
4. bytes per sector
   00 = 128 bytes
   01 = 256 bytes
   02 = 512
   03 = 1028

Returns:

good status	al, = number of sectors read ah, = 0, sucessful read cf, = 0, sucessful read
bad status	ah, = status cf, = 1, unsucessful read

INT 41H - Hard Disk Parameters

Interrupt 41H points to the hard disk parameter table.

APPENDIX A

TYPICAL INTERFACE CABLES

This Appendix contains wiring information for connection of the two Little Board/186 serial ports to typical terminals, modems, and serial printers. In the tables, signal directions are relative to the Little Board/186.

TERMINAL CABLE

Table A-1 lists the pin connections generally used to connect to a terminal. To reduce EMI radiation, the cable should be shielded, with the shield connected to the connector shell. The terminal connector can be either male or female, depending upon the specific terminal.

Table A-1. Typical Terminal Cable Wiring

Board Connector (J3)	Signal Name	Function	Terminal Connector (DB-25)
1	Ground	Protective Ground	1
5	RxD	Data Input	2
3	TxD	Data Output	3
4	HSO	Handshake Signal Out	5
2	Ground	Signal Ground	7
6	HSI	Handshake Signal In	20

SERIAL PRINTER CABLE

Table A-2 lists the pin connections generally used to connect Serial Port B to a serial printer. To reduce EMI radiation, the cable must be shielded, and the shield must be connected to the connector shell. The printer connector can be either male or female, depending upon the specific printer.

Table A-2. Typical Serial Printer Cable Wiring

Board Connector (J4)	Signal Name	Function	Printer Connector (DB-25)
1	Ground	Protective Ground	1
5	RxD	Data Input	2
3	TxD	Data Output	3
4	HSO	Hand Shake Out	5
2	Ground	Signal Ground	7
6	HSI	Hand Shake In	(11)*

NOTE

"Handshake Signal In" must connect to the printer's "Busy" output, i.e., the signal which tells the computer to start/stop sending data to the printer. The specific printer connector pin required for "Handshake Signal In" may vary between printers, so be sure to consult your printer's instruction manual.

MODEM CABLE

Table A-3 lists the pin connections generally used to connect to a modem. To reduce EMI radiation, the cable must be shielded, and the shield connected to the connector shell. The connector for the computer end must be a male DB-25, while the modem connector can be either male or female (usually male), depending upon the specific modem.

Table A-3. Typical Modem Cable Wiring

Board Connector (J4)	Signal Name	Function	Modem Connector (DB-25)
1	Ground	Protective Ground	1
5	RxD	Data Input	3
3	TxD	Data Output	2
4	HSO	Hand Shake Out	20
2	Ground	Signal Ground	7
6	HSI	Hand Shake In	5

CONNECTORS

Table A-4 lists mating connectors used with the Little Board/186.

Table A-4. Mating Connector Part Numbers

Board Connector	Function	Part Number
J1	Power Connector	Housing: AMP 1-480424-0 Contacts: AMP 60619-1 (4 req.)
J2	Parallel Printer, Board end	3M: 3399-6000 T&B: 609-2601M Molex: 15-29-8262
	Parallel Printer, Printer end	AMP: 57F-30360 3M: 3366-1001 T&B: 609-36M
J3,4	Serial Ports A,B	Housing: Molex 22-01-2067 Contacts: Molex 08-50-0114 (6 req.)
J5	RESET, Power LED	Housing: Molex 22-01-2047 Contacts: Molex 08-50-0114 (4 req.)
J6	Floppy Disk Interface (Card edge connectors)	3M: 3463-0001 T&B: 609-3415M Molex: 15-29-0341
J8	SCSI/PLUS Interface	T&B: 609-5000M Molex: 15-29-8502 Berg: 66902-150

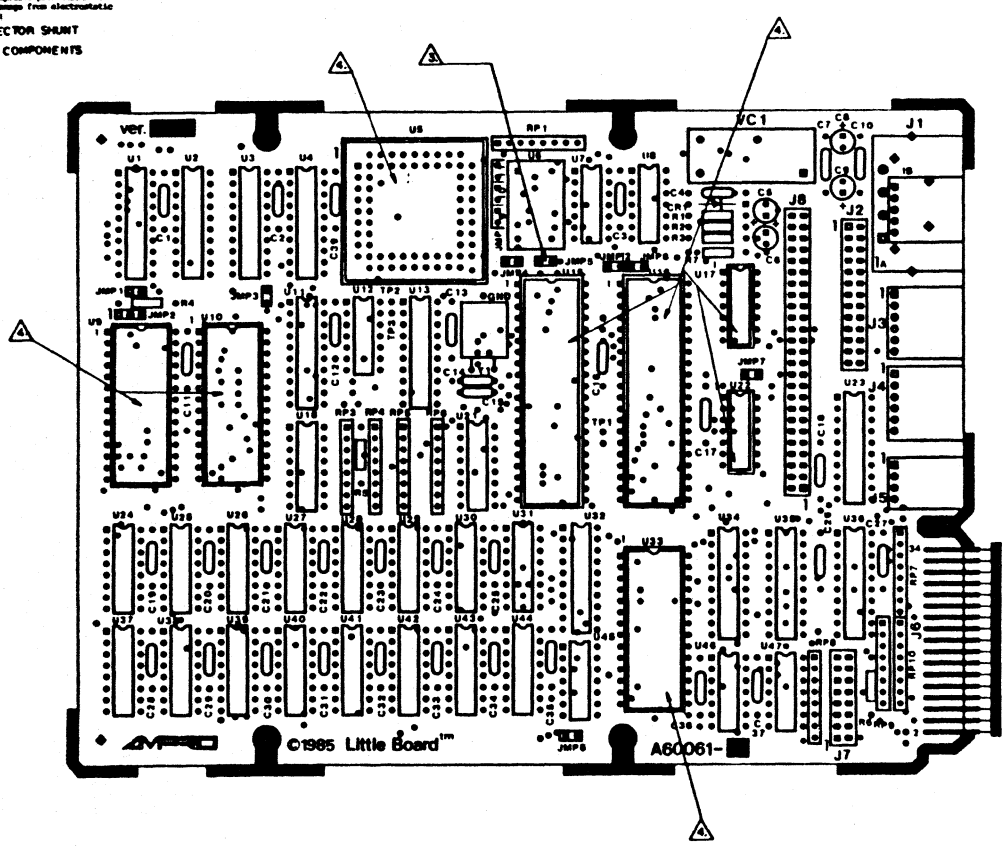
APPENDIX B

BOARD DIAGRAM, PARTS LIST, AND SCHEMATIC

NOTES: UNLESS OTHERWISE SPECIFIED

1. Square pads indicate pin one (1).
2. CAUTION: Static sensitive devices. Handle with appropriate precautions to prevent damage from electrostatic discharge!

3 INSTALL CONNECTOR SHUNT
4 SOCKETED COMPONENTS



REVISIONS					
ITEM	ZONE	REVISION		APPD	DATE
A		INITIAL RELEASE			

SIZE D — DRAWING NUMBER A60061-A

ITEM	QTY	PART NUMBER	DESCRIPTION	SPECIFICATION

	BY	DATE
DWN		
CHK		
APPV		
ENG		

APPLICATION — DIMENSIONS IN PARENTHESES ARE IN MILLIMETERS
MATERIAL
FINISH
SCALE
DO NOT SCALE DRAWING
SHEET 1 OF 3

TITLE: CPU 2A ASSY

SIZE D — DRAWING NUMBER A60061-A

AMPRO COMPUTERS INCORPORATED - PARTS LIST

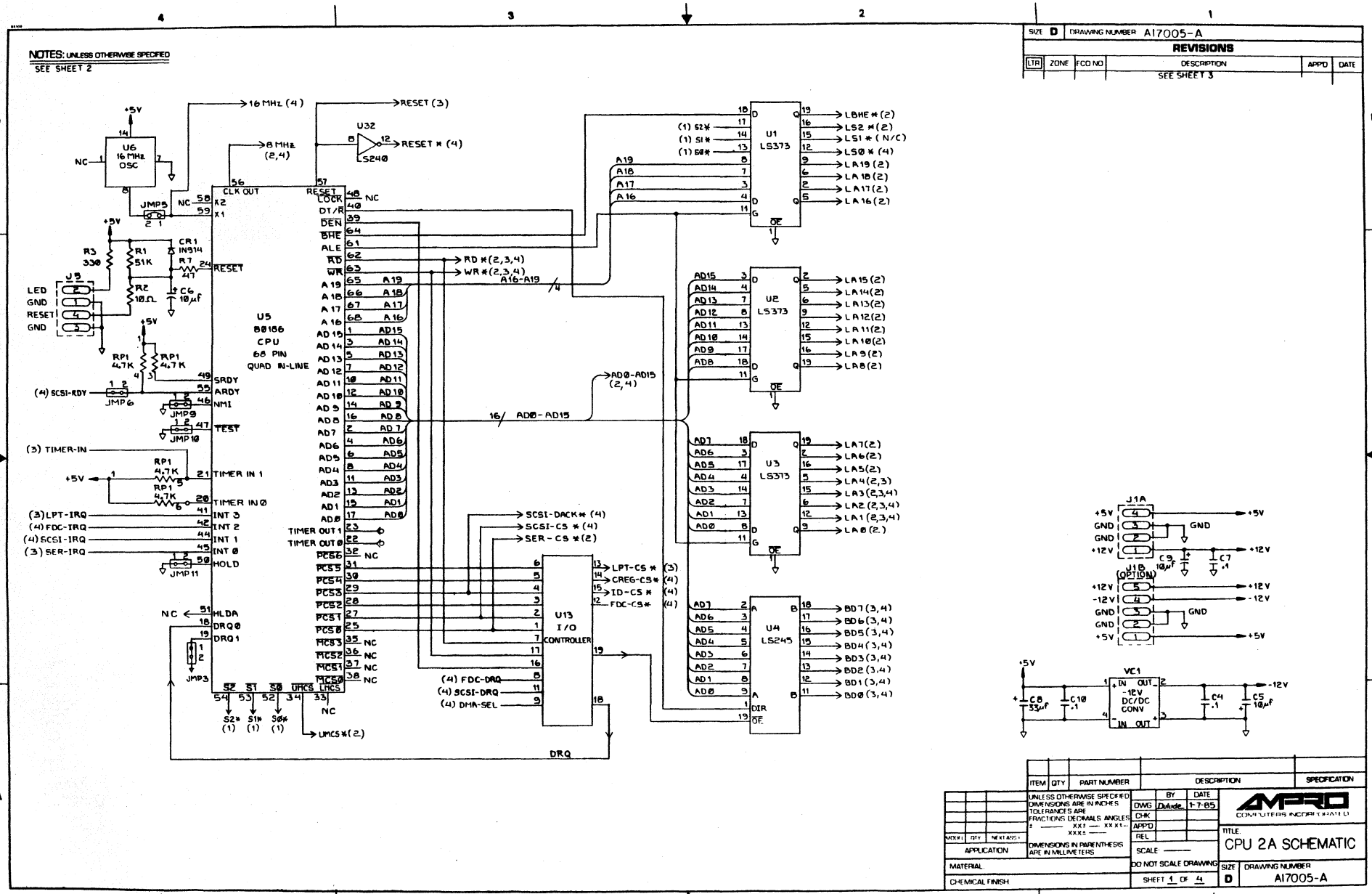
ASSEMBLY: A60061.001 - Little Board 186 REV: A DATE: 06/14/85

REF	QTY	DESCRIPTION	VENDOR P/N	AMPRO P/N
C1-4,7, 10-13, 16-38	32	CAP CER AXIAL .1UF +80% -20% 50V	MRA RPA20-Z5U-104Z50	90514-001
			CRN CAC02Z5U104Z050A	90514-001
			UNI CGC104ZDZ	90514-001
C5,6,9	3	CAP ELC RADIAL 10UF .100-OC 20% 25V	UCK LL25VB10-M	90522-001
			ALB LBRIE100S-M	90522-001
C8	1	CAP TANT 33UF 10V	SPR 199D336X0010DB1	90522-003
C14	1	CAP CER 10PF AXIAL	CORNING CAC02X74100K100A	90514-002
C15	(1)	CAP CER 5PF AXIAL (OPTION)		00000-000
J7	1	CONN HDR 16POS .100-OC STR	SAE THD6916WIS	90907-007
J2	1	CONN HDR 26POS .100-OC STR	SAE THD6926WIS	90907-002
J1	1	CONN HDR 4POS SIL RT/AG	MLX 8981-4R-1	90907-001
J5	1	CONN HDR 4POS SIL RT/AG	MLX 22-05-3041	90907-004
J8	1	CONN HDR 50POS .100-OC STR	SAE THD6950WIS	90907-009
J3,4	2	CONN HDR 6POS SIL RT/AG	MLX 22-05-3061	90907-003
JMP1,5,7	3	CONN HDR 2POS .100-OC	MLX 22-10-2021	90905-001
			AMP 64112-2	90905-001
JMP2	1	CONN HDR 3POS .100-OC		90905-003
JMP5	1	CONN SHUNT 2POS .100-OC (.40" MAX HEIGHT REQUIRED)	MLX 15-38-1024	90905-002
				90905-002
Y1	1	CRYSTAL 3.6864 MHZ	US CRYSTALS #SIG36864	90824-002
CR1	1	DIODE 1N4148		90300-001
VC1	1	HYBRID DC/DC CONV -12V	ELPAC/TDK CB3811	90702-001
U12	1	HYBRID DELAY 100NS	DATRONIC LTD #DL6307	90690-001
			BELL FUSE #0447-0100-02	90690-001
U6	1	HYBRID OSC 16MHZ	SRX NCT070C16MHZ	90824-001
			NDK TD114A-16.000MHZ	90824-001
U11	1	IC RAM CONTROLLER	AMPRO	A75509
U13	1	IC I/O CONTROLLER	AMPRO	A75506
U33	1	IC 1770 FLOPPY CTRLR		90670-001
U15	1	IC 2681 SERIAL CTRLR	SIGNETICS/MOTOROLA	90670-007
U9	1	IC 2764 250NS PROGRAMMED	INTEL/AMD/MITS/FUJ/TI/SEEQ	A75503
U10	1	IC 2764 250NS PROGRAMMED	INTEL/AMD/MITS/FUJ/TI/SEEQSU	A75504
U24-31, 37-44	16	IC 41256 RAM 150NS	*** FUJITSU ONLY ***	90680-003
				90680-003
U16	1	IC 5380 SCSI CTRLR		90670-005
U47	1	IC 7406		90620-003
U46	1	IC 7438		90620-018
U45	1	IC 74F00		90620-019
U18,21	2	IC 74LS157		90620-010
U32	1	IC 74LS240		90620-007
U34,36	3	IC 74LS244		90620-013
U4	1	IC 74LS245		90620-023
U35	1	IC 74HCT273		90650-004
U1,2,3	3	IC 74HCT373		90650-005

AMPRO COMPUTERS INCORPORATED - PARTS LIST

ASSEMBLY: A60061.001 - Little Board 186 REV: A DATE: 06/14/85

REF	QTY	DESCRIPTION	VENDOR P/N	AMPRO P/N
U23	1	IC 74LS374		90620-015
U8	1	IC 75188/1488		90660-001
U7	1	IC 75189/1489A		90660-002
U5	1	IC 80186 8MHZ *C-STEP ONLY*	INTEL/AMD	90670-006
U17,22	2	RES PK 14 DIP 12-220/330		90014-004
--	1	PCB FAB BRD MODEL 2A		A13005
--	1	SERIAL LABEL, MODEL 2A		A60073
R2	1	RES CF 10 5% 1/4W		90015-005
R4	1	RES CF 10K 5% 1/4W		90015-007
R5	1	RES CF 33 5% 1/4W		90015-004
R3	1	RES CF 330 5% 1/4W		90015-006
R6	1	RES CF 4.7K 5% 1/4W		90015-003
R7	1	RES CF 47 5% 1/4W		90015-009
R1	1	RES CF 51K 5% 1/4W		90015-008
RP7	1	RES PK 8SIP 7-330		90014-001
RP3,4,6	3	RES PK 8SIP 4-33 (INDIV)	A/B #108B330	90014-005
RP1,5,8, 9,10	5	RES PK 8SIP 7-4700		90014-003 90014-003
U17,22	2	SOCKET IC 14POS D/W	JNE J23-5014	90800-003
U7,8,32, 45,46,47	(6)	SOCKET IC 14POS D/W (OPT)	JNE J23-5014	90800-003 90800-003
U24-31, 37-44	16	SOCKET IC 16POS D/W	JNE J23-5016	90800-006 90800-006
U18,21	(2)	SOCKET IC 16POS D/W (OPT)	JNE J23-5016	90800-006
U1,2,3, 4,11,13, 23,34,35,36	(10)	SOCKET IC 20POS D/W (OPT)	JNE J23-5020	90800-007 90800-007 90800-007
U9,10,33	3	SOCKET IC 28POS D/W	JNE J23-5028	90800-004
U15,16	2	SOCKET IC 40POS D/W	JNE J23-5040	90800-002
			AMP 2-640379-3	90800-002
U5	1	SOCKET IC 68POS	TEXTOOL/3M # 268-5400-00-1102	90800-005
			AMP # ???????	90800-005
U5	1	SOCKET HDR 68POS	SAMTEC # MPAS-68-SGT-11	90800-008
			MUPAC # 3894068-02	90800-008



ITEM	QTY	PART NUMBER	DESCRIPTION	SPECIFICATION

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES
TOLERANCES ARE
FRACTIONS DECIMALS ANGLES
± —— XX± —— XX X1—
XXX± ——
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS

MODEL | QTY | NEXT ASS'Y
APPLICATION
MATERIAL
CHEMICAL FINISH
DO NOT SCALE DRAWING

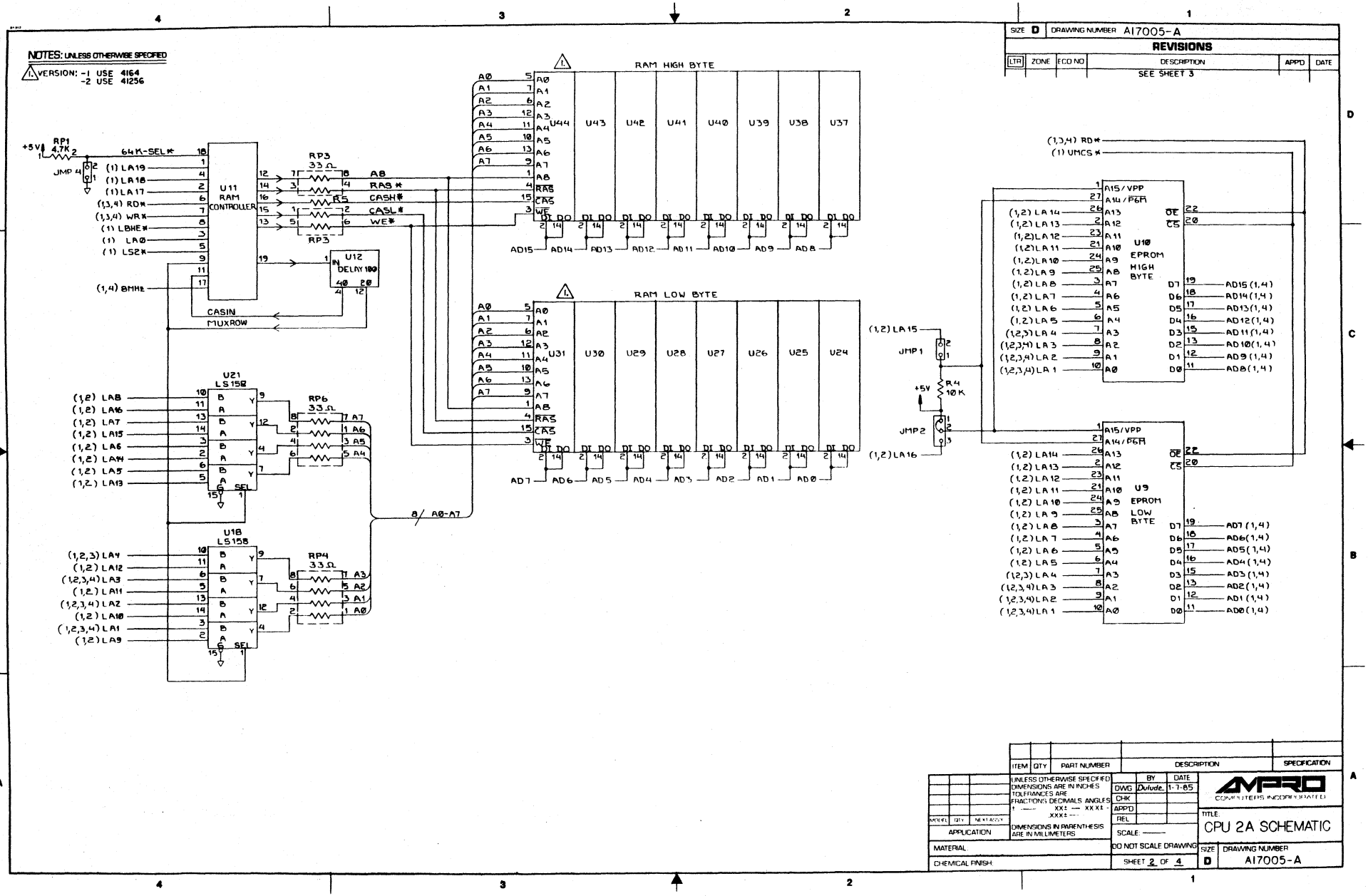
	BY	DATE
DWG	Dulude	1-7-85
CHK		
APP'D		
REL		

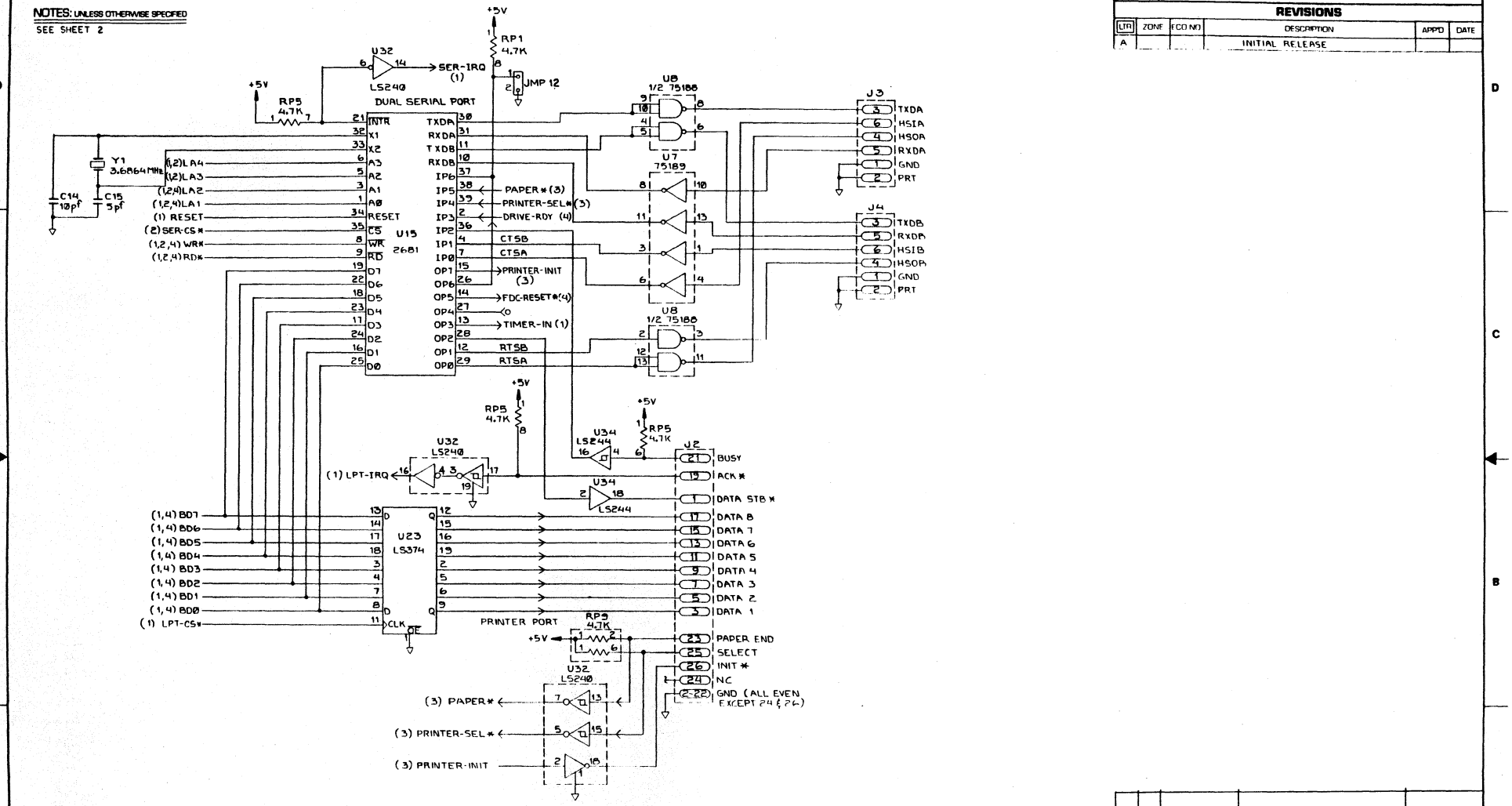
SCALE ——
SHEET 1 OF 4

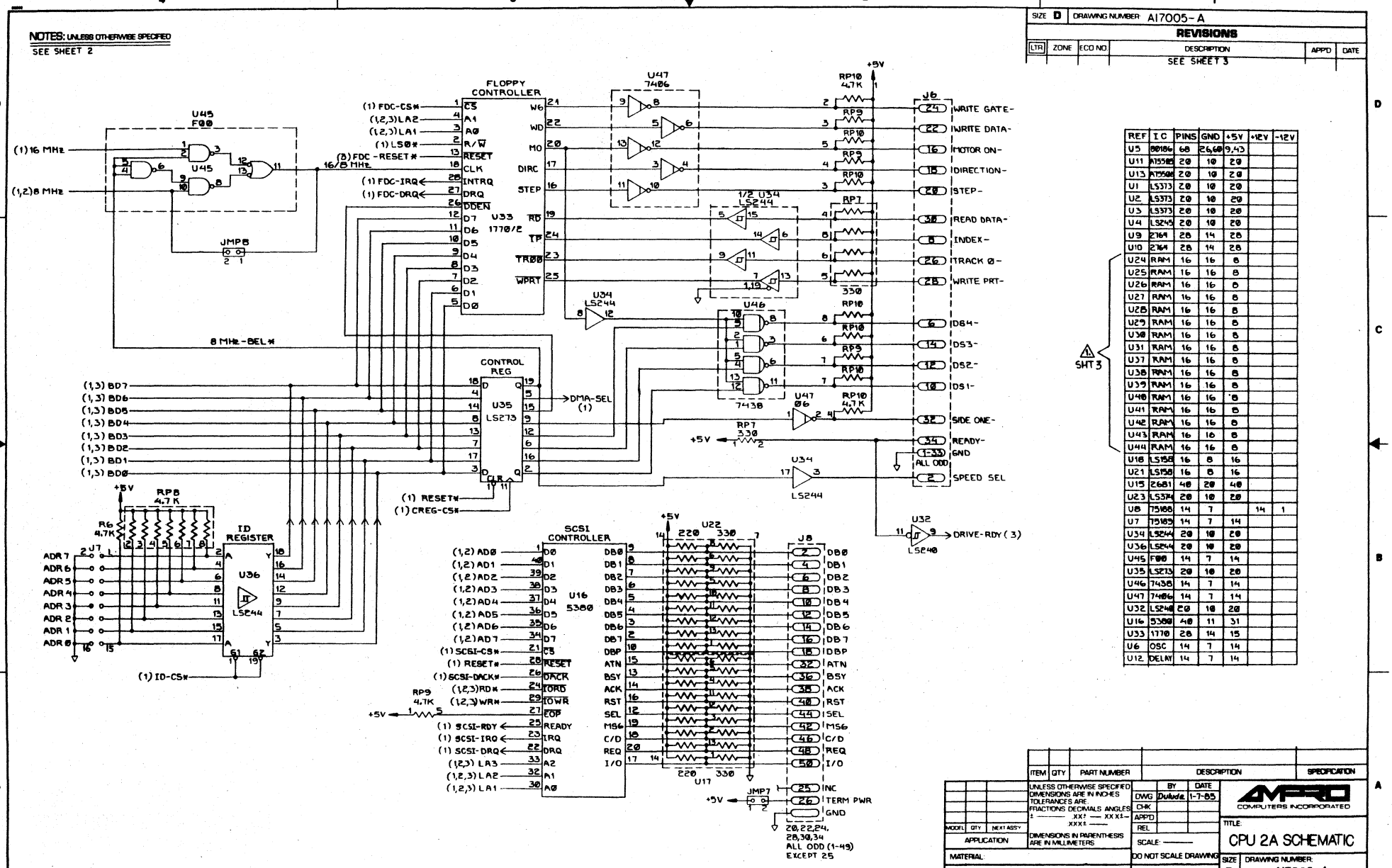
AMPRO COMPUTERS INCORPORATED

TITLE: CPU 2A SCHEMATIC

SIZE D DRAWING NUMBER A17005-A


NOTES: UNLESS OTHERWISE SPECIFIED
VERSION: -1 USE 4164
-2 USE 41256
SIZE D DRAWING NUMBER A17005-A
REVISIONS
LTR ZONE ECO NO DESCRIPTION APP'D DATE
SEE SHEET 3
RAM HIGH BYTE
RAM LOW BYTE
U11 RAM CONTROLLER
U12 DELAY 100
U21 LS158
U18 LS158
RP1 4.7K
RP3 33Ω
RP4 33Ω
RP6 33Ω
RP5
JMP4
64K-SEL*
(1) LA19
(1) LA18
(1) LA17
(1,3,4) RD*
(1,3,4) WR*
(1) LBHE*
(1) LA0
(1) LS2*
(1,4) BMHE
CASIN
MUXROW
RAS*
CASH*
CASL*
WE*
8/ A0-A7
U44 U43 U42 U41 U40 U39 U38 U37
U31 U30 U29 U28 U27 U26 U25 U24
AD15 AD14 AD13 AD12 AD11 AD10 AD9 AD8
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0
(1,2) LA15
(1,2) LA16
JMP1
JMP2
R4 10K
+5V
(1,3,4) RD*
(1) UMCS*
U10 EPROM HIGH BYTE
U9 EPROM LOW BYTE
AD15(1,4) AD14(1,4) AD13(1,4) AD12(1,4) AD11(1,4) AD10(1,4) AD9(1,4) AD8(1,4)
AD7(1,4) AD6(1,4) AD5(1,4) AD4(1,4) AD3(1,4) AD2(1,4) AD1(1,4) AD0(1,4)
ITEM QTY PART NUMBER DESCRIPTION SPECIFICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
APPLICATION
MATERIAL
CHEMICAL FINISH
DWG Dvlude 1-7-85
CHK
APP'D
REL
SCALE:
DO NOT SCALE DRAWING
SHEET 2 OF 4
AMPRO COMPUTERS INCORPORATED
TITLE: CPU 2A SCHEMATIC
SIZE D DRAWING NUMBER A17005-A


NOTES: UNLESS OTHERWISE SPECIFIED
SEE SHEET 2
SIZE D DRAWING NUMBER A17005-A
REVISIONS
LTR ZONE ECO NO DESCRIPTION APPD DATE
A INITIAL RELEASE
+5V
RP1 4.7K
U32 LS240
SER-IRQ (1)
JMP 12
RP5 4.7K
DUAL SERIAL PORT
INTR
X1
X2
A3
A2
A1
A0
RESET
CS
WR
RD
D7
D6
D5
D4
D3
D2
D1
D0
U15 2681
TXDA
RXDA
TXDB
RXDB
IP6
IP5
IP4
IP3
IP2
IP1
IP0
OP1
OP6
OP5
OP4
OP3
OP2
OP1
OP0
Y1 3.6864 MHz
C14 10pf
C15 5pf
(1,2)LA4
(1,2)LA3
(1,2,4)LA2
(1,2,4)LA1
(1) RESET
(2) SER-CS*
(1,2,4) WR*
(1,2,4) RD*
PAPER* (3)
PRINTER-SEL* (3)
DRIVE-RDY (4)
CTSB
CTSA
PRINTER-INIT (3)
FDC-RESET* (4)
TIMER-IN (1)
RTSB
RTSA
U8 1/2 75188
U7 75189
J3
TXDA
HSIA
HSOA
RXDA
GND
PRT
J4
TXDB
RXDB
HSIB
HSOB
GND
PRT
U34 LS244
U32 LS240
(1) LPT-IRQ
J2
BUSY
ACK*
DATA STB*
DATA 8
DATA 7
DATA 6
DATA 5
DATA 4
DATA 3
DATA 2
DATA 1
PAPER END
SELECT
INIT*
NC
GND (ALL EVEN EXCEPT 24 & 26)
U23 LS374
(1,4) BD7
(1,4) BD6
(1,4) BD5
(1,4) BD4
(1,4) BD3
(1,4) BD2
(1,4) BD1
(1,4) BD0
(1) LPT-CS*
PRINTER PORT
(3) PAPER*
(3) PRINTER-SEL*
(3) PRINTER-INIT
ITEM QTY PART NUMBER DESCRIPTION SPECIFICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE FRACTIONS DECIMALS ANGLES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
APPLICATION
MATERIAL
CHEMICAL FINISH
BY DATE
DWG Dwade 1-7-85
CHK
APPD
REL
SCALE
DO NOT SCALE DRAWING
SHEET 3 OF 4
AMPRO COMPUTERS INCORPORATED
TITLE: CPU 2A SCHEMATIC
SIZE D DRAWING NUMBER A17005-A


NOTES: UNLESS OTHERWISE SPECIFIED
SEE SHEET 2
SIZE D DRAWING NUMBER AI7005-A
REVISIONS
LTR ZONE ECO NO. DESCRIPTION APPD DATE
SEE SHEET 3
FLOPPY CONTROLLER
U33 1770/2
U45 F00
U47 7406
RP10 4.7K
RP9
RP7 330
U6
WRITE GATE-
WRITE DATA-
MOTOR ON-
DIRECTION-
STEP-
READ DATA-
INDEX-
TRACK 0-
WRITE PRT-
SIDE ONE-
READY-
GND
ALL ODD
SPEED SEL
DRIVE-RDY (3)
(1) FDC-CS*
(1,2,3) LA2
(1,2,3) LA1
(1) LS0*
(1) FDC-RESET*
(1) FDC-IRQ
(1) FDC-DRQ
(1) 16 MHz
(1,2) 8 MHz
16/8 MHz
JMP8
8 MHz-BEL*
1/2 U34 LS244
U34 LS244
U46 7438
U47 06
CONTROL REG
U35 LS273
DMA-SEL (1)
(1) RESET*
(1) CREG-CS*
(1,3) BD7
(1,3) BD6
(1,3) BD5
(1,3) BD4
(1,3) BD3
(1,3) BD2
(1,3) BD1
(1,3) BD0
ID REGISTER
U36 LS244
RP8 4.7K
R6 4.7K
U7
ADR 7
ADR 6
ADR 5
ADR 4
ADR 3
ADR 2
ADR 1
ADR 0
(1) ID-CS*
SCSI CONTROLLER
U16 5380
U22 220 330
U17 220 330
U8
RP9 4.7K
(1,2) AD0
(1,2) AD1
(1,2) AD2
(1,2) AD3
(1,2) AD4
(1,2) AD5
(1,2) AD6
(1,2) AD7
(1) SCSI-CS*
(1) RESET*
(1) SCSI-DACK*
(1,2,3) RD*
(1,2,3) WR*
(1) SCSI-RDY
(1) SCSI-IRQ
(1) SCSI-DRQ
(1,2,3) LA3
(1,2,3) LA2
(1,2,3) LA1
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DBP
ATN
BSY
ACK
RST
SEL
MSG
C/D
REQ
I/O
JMP7
INC
TERM PWR
GND
20,22,24, 28,30,34 ALL ODD (1-49) EXCEPT 25
SHT 3
REF IC PINS GND +5V +12V -12V
U5 80186 68 26,60 9,43
U11 AT5508 20 10 20
U13 AT5508 20 10 20
U1 LS373 20 10 20
U2 LS373 20 10 20
U3 LS373 20 10 20
U4 LS245 20 10 20
U9 2764 28 14 28
U10 2764 28 14 28
U24 RAM 16 16 8
U25 RAM 16 16 8
U26 RAM 16 16 8
U27 RAM 16 16 8
U28 RAM 16 16 8
U29 RAM 16 16 8
U30 RAM 16 16 8
U31 RAM 16 16 8
U37 RAM 16 16 8
U38 RAM 16 16 8
U39 RAM 16 16 8
U40 RAM 16 16 8
U41 RAM 16 16 8
U42 RAM 16 16 8
U43 RAM 16 16 8
U44 RAM 16 16 8
U18 LS158 16 8 16
U21 LS158 16 8 16
U15 2681 40 20 40
U23 LS374 20 10 20
U8 75188 14 7 14 1
U7 75189 14 7 14
U34 LS244 20 10 20
U36 LS244 20 10 20
U45 F00 14 7 14
U35 LS273 20 10 20
U46 7438 14 7 14
U47 7406 14 7 14
U32 LS240 20 10 20
U16 5380 40 11 31
U33 1770 28 14 15
U6 OSC 14 7 14
U12 DELAY 14 7 14
ITEM QTY PART NUMBER DESCRIPTION SPECIFICATION
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS DECIMALS ANGLES ± XX? — XX X± XXX± —
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
MODEL QTY NEXT ASSY
APPLICATION
MATERIAL
CHEMICAL FINISH
BY DATE
DWG Dubois 1-7-85
CHK
APPD
REL
SCALE:
DO NOT SCALE DRAWING
SHEET 4 OF 4
AMPRO COMPUTERS INCORPORATED
TITLE: CPU 2A SCHEMATIC
SIZE D DRAWING NUMBER AI7005-A

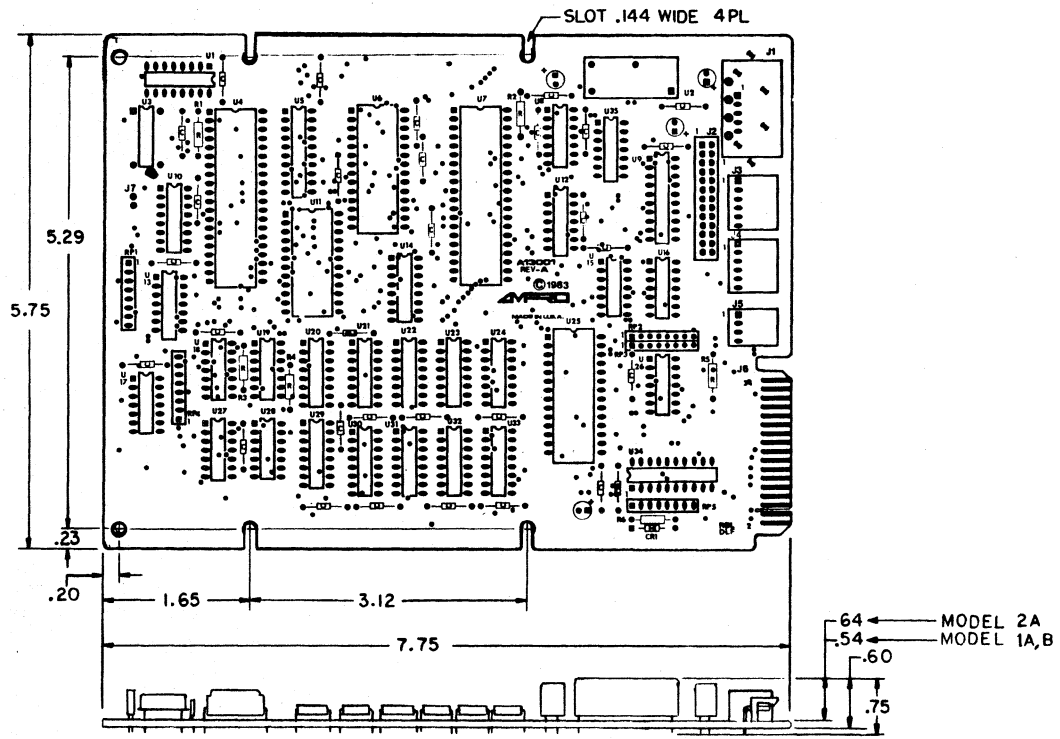
SIZE C	DRAWING NUMBER: A17002-A

REVISIONS

LTR	ZONE	ECO NO.	DESCRIPTION	APP'D	DATE
A			INITIAL RELEASE		1-9-84

NOTES: UNLESS OTHERWISE SPECIFIED

1. COMPONENT LAYOUT SHOWN IS CPU 1A.
2. MOUNTING DIMENSIONS ARE THE SAME FOR ALL MODELS.
3. J1,3,4,5,6 LOCATION IS THE SAME.



ITEM	QTY	PART NUMBER	DESCRIPTION	SPECIFICATION

MODEL	QTY	NEXT ASSY
APPLICATION		

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. TOLERANCES ARE:
FRACTIONS DECIMALS ANGLES
± .XX±.02 XX.X± .XXX±.010
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS.

MATERIAL:

CHEMICAL FINISH:

	BY	DATE
DWG	DJ	2-7-84
CHK		
APP'D		2-8-84
REL		

SCALE: 1 X

DO NOT SCALE DRAWING

SHEET 1 OF 1

AMPRO COMPUTERS, INCORPORATED

TITLE: CPU MOUNTING

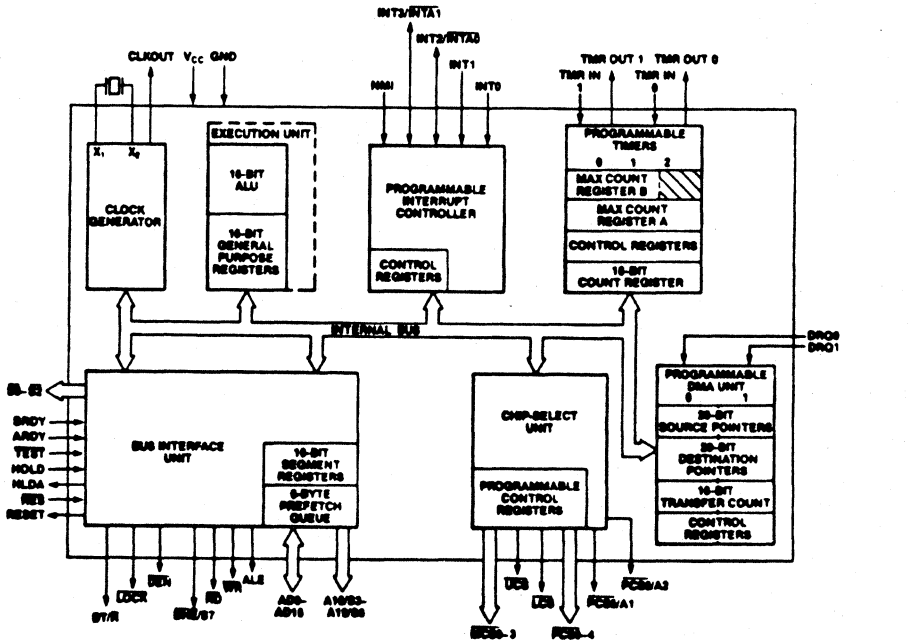
SIZE C | DRAWING NUMBER: A17002-A

APPENDIX C

COMPONENT DATA SHEETS

iAPX 186
HIGH INTEGRATION 16-BIT MICROPROCESSOR

- **Integrated Feature Set**
  - Enhanced 8066-2 CPU
  - Clock Generator
  - 2 Independent, High-Speed DMA Channels
  - Programmable Interrupt Controller
  - 3 Programmable 16-bit Timers
  - Programmable Memory and Peripheral Chip-Select Logic
  - Programmable Wait State Generator
  - Local Bus Controller
- **Available in 10 MHz (80186-10), 8 MHz (80186), and 6 MHz (80186-6) Versions.**
- **High-Performance Processor**
  - 2 Times the Performance of the Standard iAPX 86
  - 4 MByte/Sec Bus Bandwidth Interface
- **Direct Addressing Capability to 1 MByte of Memory**
- **Completely Object Code Compatible with All Existing iAPX 86, 88 Software**
  - 10 New Instruction Types
- **Complete System Development Support**
  - Development Software; Assembler, PL/M, Pascal, Fortan, and System Utilities
  - In-Circuit-Emulator (I²ICE™-186)
  - iRMX™ 86, 88 Compatible (80130 OSF)
- **High Performance Numerical Coprocessing Capability Through 8087 Interface**

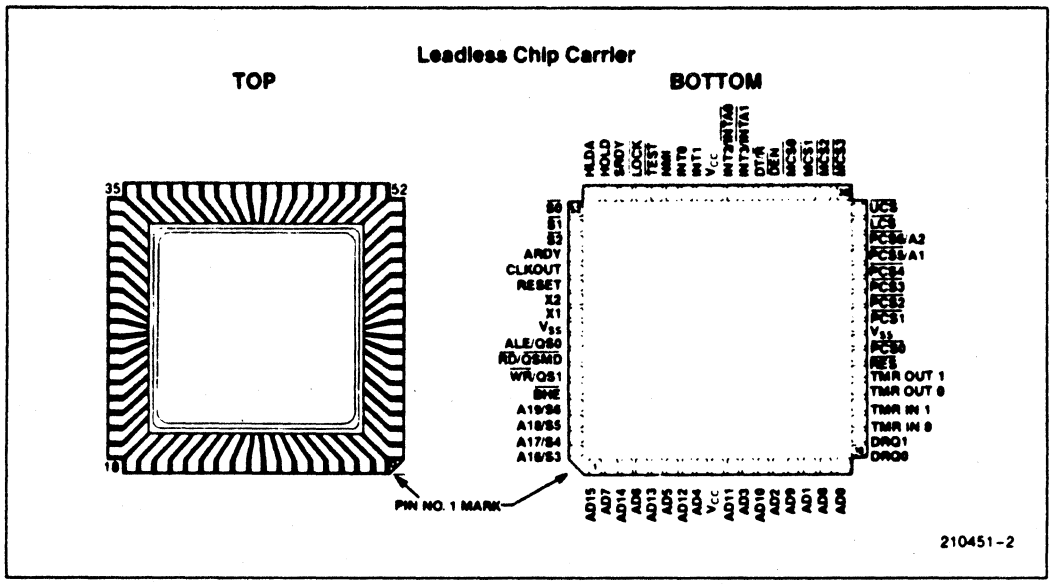


210451-1

Figure 1. iAPx 186 Block Diagram

Order Number: 210451-006

The Intel iAXP 186 (80186 part number) is a highly integrated 16-bit microprocessor. The iAPX 186 effectively combines 15-20 of the most common iAPX 86 system components onto one. The 80186 provides two times greater throughput than the standard 5 MHz iAPX 86. The iAPX 186 is upward compatible with iAPX 86 and 88 software and adds 10 new instruction types to the existing set.



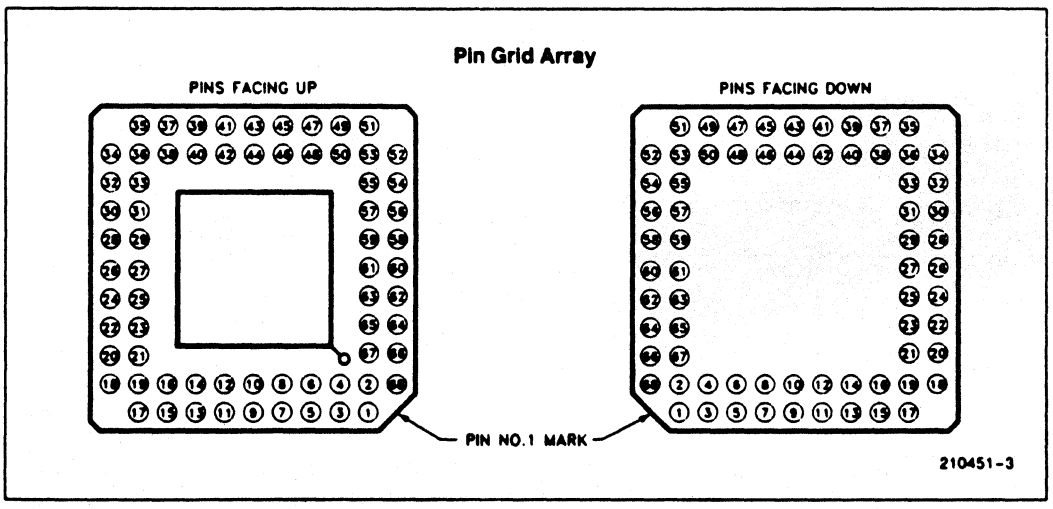


Figure 2. 80186 Pinout Diagram

Table 1. 80186 Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}, V_{CC}	9, 43	I	**System Power:** + 5 volt power supply.
V_{SS}, V_{SS}	26, 60	I	System Ground.
RESET	57	O	Reset Output indicates that the 80186 CPU is being reset, and can be used as a system reset. It is active HIGH, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the $\overline{RES}$ signal.
X1, X2	59, 58	I	Crystal Inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. In this case, minimize the capacitance on X2 or drive X2 with complemented X1. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).
CLKOUT	56	O	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for the 8087 Numeric Processor Extension.
$\overline{RES}$	24	I	System Reset causes the 80186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80186 clock. The 80186 begins fetching instructions approximately 7 clock cycles after $\overline{RES}$ is returned HIGH. $\overline{RES}$ is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of $\overline{RES}$ must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. When $\overline{RES}$ occurs, the 80186 will drive the status lines to an inactive level for one clock, and then tri-state them.
$\overline{TEST}$	47	I	$\overline{TEST}$ is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when "WAIT" execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80186 is waiting for $\overline{TEST}$, interrupts will be serviced. This input is synchronized internally.
TMR IN 0, TMR IN 1	20 21	I I	Timer Inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active HIGH (or LOW-to-HIGH transitions are counted) and internally synchronized.
TMR OUT 0, TMR OUT 1	22 23	O O	Timer outputs are used to provide single pulse or continous waveform generation, depending upon the timer mode selected.
DRQ0 DRQ1	18 19	I I	DMA Request is driven HIGH by an external device when it desires that a DMA channel (Channel 0 or 1) perform a transfer. These signals are active HIGH, level-triggered, and internally synchronized.
NMI	46	I	Non-Maskable Interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
INT0, INT1 INT2/$\overline{INTA0}$ INT3/$\overline{INTA1}$	45, 44 42 41	I I/O I/O	Maskable Interrupt Requests can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge- or level-triggered. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged. When iRMX mode is selected, the function of these pins changes (see Interrupt Controller section of this data sheet).

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
A19/S6, A18/S5, A17/S4, A16/S3	65 66 67 68	O O O O	Address Bus Outputs (16–19) and Bus Cycle Status (3–6) reflect the four most significant address bits during T_1. These signals are active HIGH. During T_2, T_3, T_W, and T_4, status information is available on these lines as encoded below:
			Low / **High**
			S6: Processor Cycle / DMA Cycle
			S3, S4, and S5 are defined as LOW during T_2–T_4.
AD15–AD0	10–17, 1–8	I/O	Address/Data Bus (0–15) signals constitute the time multiplexed memory or I/O address (T_1) and data (T_2, T_3, T_W, and T_4) bus. The bus is active HIGH. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7 through D_0. It is LOW during T_1 when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations.
$\overline{BHE}$/S7	64	O	During T_1 the Bus High Enable signal should be used to determine if data is to be enabled onto the most significant half of the data bus; pins D_{15}–D_8. $\overline{BHE}$ is LOW during T_1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the higher half of the bus. The S_7 status information is available during T_2, T_3, and T_4. S_7 is logically equivalent to $\overline{BHE}$. The signal is active LOW, and is tristated OFF during bus HOLD.
			$\overline{BHE}$ and A0 Encodings
			$\overline{BHE}$ Value / **A0 Value** / **Function**
			0 / 0 / Word Transfer
			0 / 1 / Byte Transfer on upper half of data bus (D15–D8)
			1 / 0 / Byte Transfer on lower half of data bus (D_7–D_0)
			1 / 1 / Reserved
ALE/QS0	61	O	Address Latch Enable/Queue Status 0 is provided by the 80186 to latch the address into the 8282/8283 address latches. ALE is active HIGH. Addresses are guaranteed to be valid on the trailing edge of ALE. The ALE rising edge is generated off the rising edge of the CLKOUT immediately preceding T_1 of the associated bus cycle, effectively one-half clock cycle earlier than in the standard 8086. The trailing edge is generated off the CLKOUT rising edge in T_1 as in the 8086. Note that ALE is never floated.
$\overline{WR}$/QS1	63	O	Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. $\overline{WR}$ is active for T_2, T_3, and T_W of any write cycle. It is active LOW, and floats during "HOLD." It is driven HIGH for one clock during Reset, and then floated. When the 80186 is in queue status mode, the ALE/QS0 and $\overline{WR}$/QS1 pins provide information about processor/instruction queue interaction.
			QS1 / **QS0** / **Queue Operation**
			0 / 0 / No queue operation
			0 / 1 / First opcode byte fetched from the queue
			1 / 1 / Subsequent byte fetched from the queue
			1 / 0 / Empty the que

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{RD}$/QSMD	62	O	Read Strobe indicates that the 80186 is performing a memory or I/O read cycle. $\overline{RD}$ is active LOW for T_2, T_3, and T_W of any read cycle. It is guaranteed not to go LOW in T_2 until after the Address Bus is floated. $\overline{RD}$ is active LOW, and floats during "HOLD". $\overline{RD}$ is driven HIGH for one clock during Reset, and then the output driver is floated. A weak internal pull-up mechanism of the $\overline{RD}$ line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80186 should provide ALE, $\overline{WR}$ and $\overline{RD}$, or if the Queue-Status should be provided. $\overline{RD}$ should be connected to GND to provide Queue-Status data.
ARDY	55	I	Asynchronous Ready informs the 80186 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input, and is active HIGH. Only the rising edge is internally synchronized by the 80186. This means that the falling edge of ARDY must be synchronized to the 80186 clock. If connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active to terminate a bus cycle. If unused, this line should be tied LOW.
SRDY	49	I	Synchronous Ready must be synchronized externally to the 80186. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated. If unused, this line should be tied LOW.
$\overline{LOCK}$	48	O	$\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. It remains active until the completion of the instruction following the LOCK prefix. No prefetches will occur while $\overline{LOCK}$ is asserted. When executing more than one LOCK instruction, always make sure there are 6 bytes of code between the end of the first LOCK instruction and the start of the second LOCK instruction. $\overline{LOCK}$ is active LOW, is driven HIGH for one clock during RESET, and then floated.
$\overline{S0}$, $\overline{S1}$, $\overline{S2}$	52–54	O	Bus cycle status $\overline{S0}$–$\overline{S2}$ are encoded to provide bus-transaction information:

80186 Bus Cycle Status Information

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Bus Cycle Initiated
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

The status pins float during "HOLD."
$\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator.
The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
HOLD (input) HLDA (output)	50 51	I O	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80186 clock. The 80186 will issue a HLDA (HIGH) in response to a HOLD request at the end of T_4 or T_1. Simultaneous with the issuance of HLDA, the 80186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80186 will lower HLDA. When the 80186 needs to run another bus cycle, it will again drive the local bus and control lines.
$\overline{UCS}$	34	O	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K–256K block) of memory. This line is not floated during bus HOLD. The address range activating $\overline{UCS}$ is software programmable.
$\overline{LCS}$	33	O	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K–256K) of memory. This line is not floated during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.
$\overline{MCS0}$–3	38, 37, 36, 35	O	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined mid-range portion of memory (8K–512K). These lines are not floated during bus HOLD. The address ranges activating $\overline{MCS0}$–3 are software programmable.
$\overline{PCS0}$ $\overline{PCS1}$–4	25 27, 28, 29, 30	O O	Peripheral Chip Select signals 0–4 are active LOW when a reference is made to the defined peripheral area (64K byte I/O space). These lines are not floated during bus HOLD. The address ranges activating $\overline{PCS0}$–4 are software programmable.
$\overline{PCS5}$/A1	31	O	Peripheral Chip Select 5 or Latched. A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating $\overline{PCS5}$ is software programmable. When programmed to provide latched. A1, rather than $\overline{PCS5}$, this pin will retain the previously latched value of A1 during a bus HOLD. A1 is active HIGH.
$\overline{PCS6}$/A2	32	O	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating $\overline{PCS6}$ is software programmable. When programmed to provide latched A2, rather than $\overline{PCS6}$, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active HIGH.
DT/$\overline{R}$	40	O	Data Transmit/Receive controls the direction of data flow through the external 8286/8287 data bus transceiver. When LOW, data is transferred to the 80186. When HIGH the 80186 places write data on the data bus.
$\overline{DEN}$	39	O	Data Enable is provided as an 8286/8287 data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access. $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state.

FUNCTIONAL DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the iAPX 186. This architecture is common to the iAPX 86, 88, and 286 microprocessor families as well. The iAPX 186 is a very high integration 16-bit microprocessor. It combines 15–20 of the most common microprocessor system components onto one chip while providing twice the performance of the standard iAPX 86. The 80186 is object code compatible with the iAPX 86, 88 microprocessors and adds 10 new instruction types to the existing iAPX 86, 88 instruction set.

iAPX 186 BASE ARCHITECTURE

The iAPX 86, 88, 186, and 286 family all contain the same basic set of registers, instructions, and addressing modes. The 80186 processor is upward compatible with the 8086, 8088, and 80286 CPUs.

Register Set

The 80186 base architecture has fourteen registers as shown in Figures 3a and 3b. These registers are grouped into the following categories.

General Registers

Eight 16-bit general purpose registers may be used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers

Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers

Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

Status and Control Registers

Two 16-bit special purpose registers record or alter certain aspects of the 80186 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 3a and 3b).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80186 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 2.

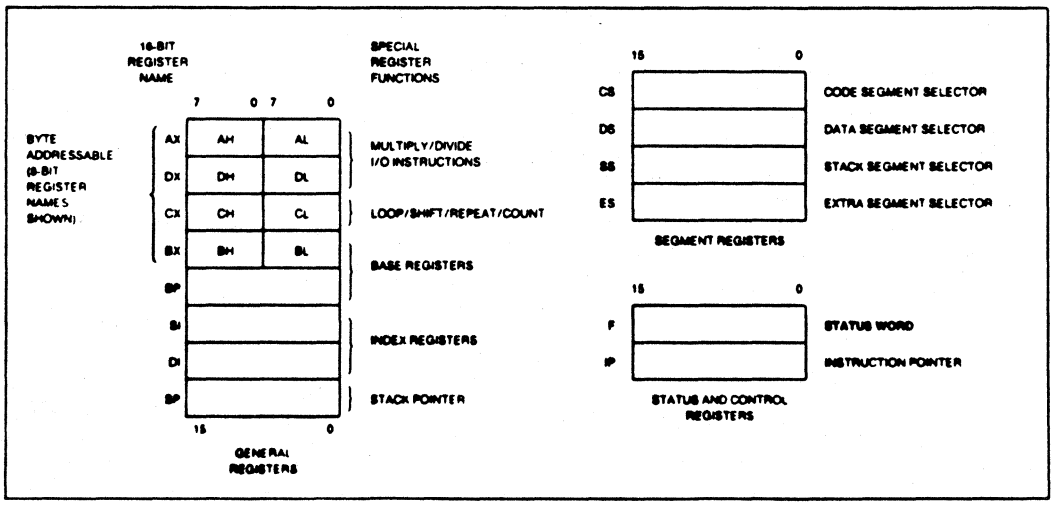


Figure 3a. 80186 General Purpose Register Set

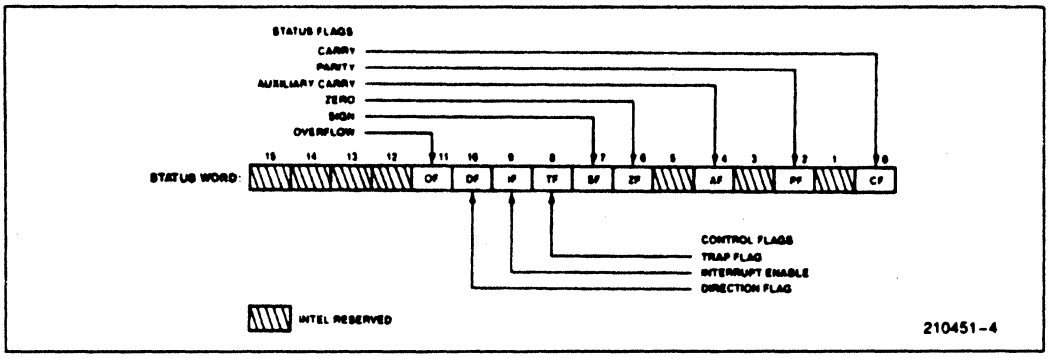


Figure 3b. Status Word Format

Table 2. Status Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1-bits; cleared otherwise
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise
6	ZF	Zero Flag—Set if result is zero; cleared otherwise
7	SF	Sign Flag—Set equal to high-order bit of result (0 if positive, 1 if negative)
8	TF	Single Step Flag—Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag—Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto increment.
11	OF	Overflow Flag—Set if the signed result cannot be expressed within the number of bits in the destination operand; cleared otherwise

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in Figure 4.

An 80186 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this data sheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained in one of four internal segment register (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 5). This allows for a 1 MByte physical address size.

All instructions that address operands in memory must specify the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack
ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero
LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word
FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{TEST}$ pin active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
HIGH LEVEL INSTRUCTIONS	
ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 4. iAPX 186 Instruction Set

CONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal
JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above nor equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less nor equal
JGE/JNL	Jump if greater or equal/not less
JL/JNGE	Jump if less/not greater nor equal
JLE/JNG	Jump if less or equal/not greater
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if sign
UNCONDITIONAL TRANSFERS	
CALL	Call procedure
RET	Return from procedure
JMP	Jump
ITERATION CONTROLS	
LOOP	Loop
LOOPE/LOOPZ	Loop if equal/zero
LOOPNE/LOOPNZ	Loop if not equal/not zero
JCXZ	Jump if register CX = 0
INTERRUPTS	
INT	Interrupt
INTO	Interrupt if overflow
IRET	Interrupt return

Figure 4. IAPX 186 Instruction Set (Continued)

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

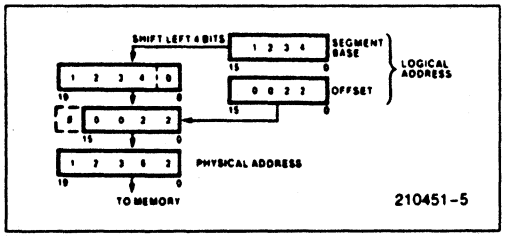


Figure 5. Two Component Address

Table 3. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.

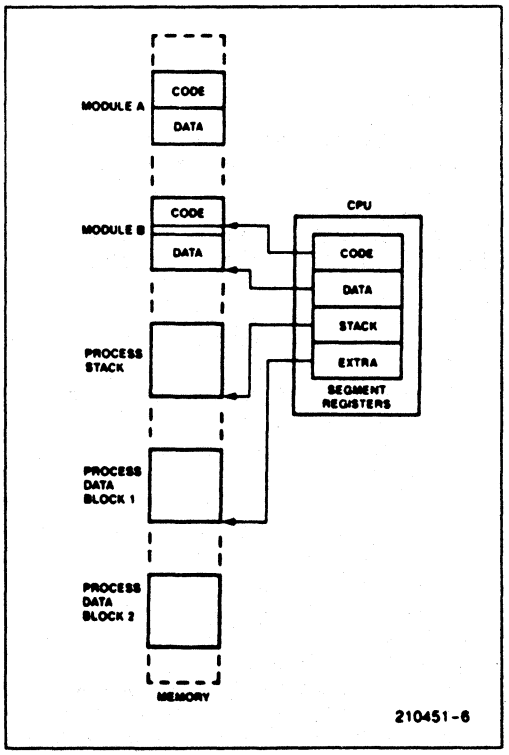


Figure 6. Segmented Memory Helps Structure Software

Addressing Modes

The 80186 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

- *Register Operand Mode:* The operand is located in one of the 8- or 16-bit general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers).

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

- *Direct Mode:* The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- *Register Indirect Mode:* The operand's offset is in one of the registers SI, DI, BX, or BP.
- *Based Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- *Indexed Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- *Based Indexed Mode:* The operand's offset is the sum of the contents of a base register and an Index register.
- *Based indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80186 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32- and 64-bit integers are supported using the iAPX 186/20 Numeric Data Processor.
- *Ordinal:* An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- *Pointer:* A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- *String:* A contiguous sequence of bytes or words. A string may contain from 1 to 64K bytes.
- *ASCII:* A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- *BCD:* A byte (unpacked) representation of the decimal digits 0–9.
- *Packed BCD:* A byte (packed) representation of two decimal digits (0–9). One digit is stored in each nibble (4-bits) of the byte.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using the iAPX 186/20 Numeric Data Processor configuration.)

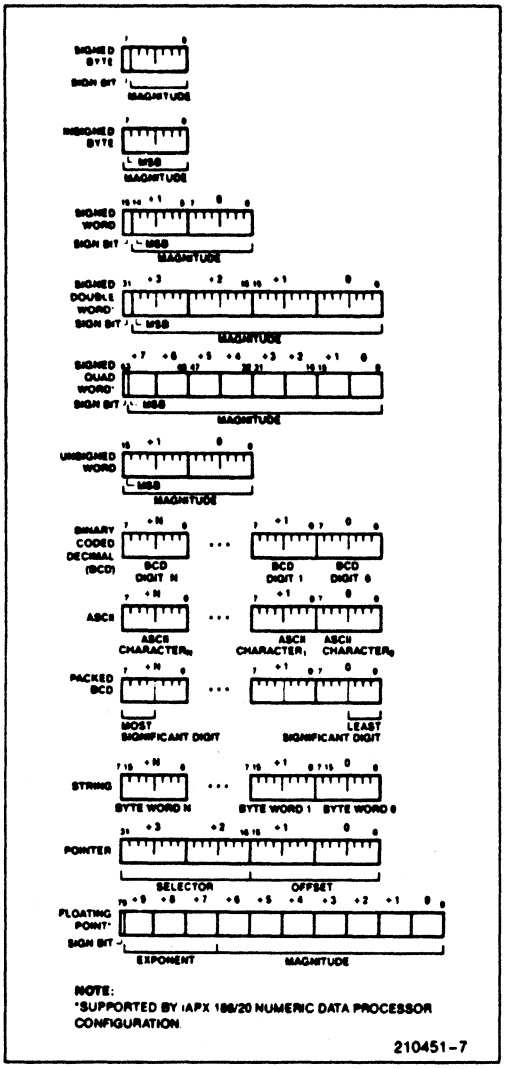
In general, individual data elements must fit within defined segment limits. Figure 7 graphically represents the data types supported by the iAPX 186.

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. 8-bit port addresses are zero extended such that A_{15}–A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.



NOTE:
*SUPPORTED BY iAPX 186/20 NUMERIC DATA PROCESSOR CONFIGURATION

210451-7

Figure 7. iAPX 186 Supported Data Types

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0–31, some of which are used for instruction exceptions, are reserved. Table 4 shows the 80186 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80186 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and noncascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80186 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INT0, DIV, IDIV, etc.). All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 4), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interrupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

The software generated 80186 interrupts are described below.

DIVIDE ERROR EXCEPTION (TYPE 0)

Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of bits in the destination.

SINGLE-STEP INTERRUPT (TYPE 1)

Generated after most instructions if the TF flag is set. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction.

NON-MASKABLE INTERRUPT—NMI (TYPE 2)

An external interrupt source which cannot be masked.

Table 4. 80186 Interrupt Vectors

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV, IDIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INT0 Detected Overflow Exception	4	*1	INT0
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 Interrupt	8	2A****	
Timer 1 Interrupt	18	2B****	
Timer 2 Interrupt	19	2C****	
Reserved	9	3	
DMA 0 Interrupt	10	4	
DMA 1 Interrupt	11	5	
INT0 Interrupt	12	6	
INT1 Interrupt	13	7	
INT2 Interrupt	14	8	
INT3 Interrupt	15	9	

NOTES:
*1. These are generated as the result of an instruction execution.
**2. This is handled as in the 8086.
****3.All three timers constitute one source of request to the interrupt controller. The Timer interrupts all have the same default priority level with respect to all other interrupt sources. However, they have a defined priority ordering amongst themselves. (Priority 2A is higher priority than 2B.) Each Timer interrupt has a separate vector type number.
4. Default priorities for the interrupt sources are used only if the user does not program each source into a unique priority level.
***5. An escape opcode will cause a trap only if the proper bit is set in the peripheral control block relocation register.

BREAKPOINT INTERRUPT (TYPE 3)

A one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

INT0 DETECTED OVERFLOW EXCEPTION (TYPE4)

Generated during an INT0 instruction if the 0F bit is set.

ARRAY BOUNDS EXCEPTION (TYPE 5)

Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

UNUSED OPCODE EXCEPTION (TYPE 6)

Generated if execution is attempted on undefined opcodes.

ESCAPE OPCODE EXCEPTION (TYPE 7)

Generated if execution is attempted of ESC opcodes (D8H–DFH). This exception will only be generated if a bit in the relocation register is set. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Hardware-generated interrupts are divided into two groups: maskable interrupts and non-maskable interrupts. The 80186 provides maskable hardware interrupt request pins INT0–INT3. In addition, maskable interrupts may be generated by the 80186 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this data sheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80186 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt (NMI) is also provided. This interrupt is serviced regardless of the state of the IF bit. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an NMI interrupt to prevent maskable interrupts from being serviced.

Single-Step Interrupt

The 80186 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single-step interrupt and is controlled by the single-step flag bit (TF) in the Status Word. Once this bit is set, an internal single-step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin LOW. $\overline{RES}$ forces the 80186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80186 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 5.

Table 5. 80186 Initial Registerr State after RESET

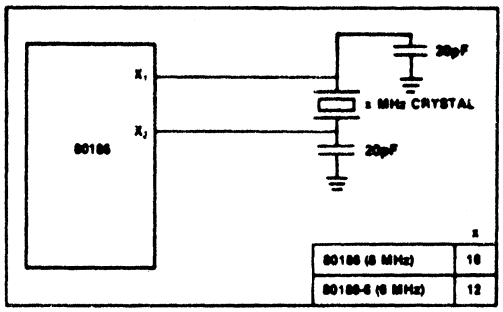
Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

iAPX 186 CLOCK GENERATOR

The iAPX 186 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the iAPX 186 is designed to be used with a parallel resonant fundamental mode crystal. This is used as the time base for the iAPX 186. The crystal frequency selected will be double the CPU clock frequency. Use of an LC or RC circuit is not recommended with this oscillator. If an external oscillator is used, it can be connected directly to input pin X1 in lieu of a crystal. The output of the oscillator is not directly available outside the iAPX 186. The recommended crystal configuration is shown in Figure 8.



210451-8

Figure 8. Recommended iAPX 186 Crystal Configuration

The following parameters may be used for choosing a crystal:

Temperature Range:	0 to 70°C
ESR (Equivalent Series Resistance):	3Ω max
C_0 (Shunt Capacitance of Crystal):	7.0 pf max
C_1 (Load Capacitance):	20 pf ± 2 pf
Dirve Level:	1 mw max

Clock Generator

The iAPX 186 clock generator provides the 50% duty cycle processor clock for the iAPX 186. It does this by dividing the oscillator output by 2 forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the iAPX 186. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The iAPX 186 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T_2, T_3 and again in the middle of each T_W until ARDY is sampled HIGH. One-half CLKOUT cycle of resolution time is used. Full synchronization is performed only on the rising edge of ARDY, i.e., the falling edge of ARDY must be synchronized to the CLKOUT signal if it will occur during T_2, T_3, or T_W. High-to-LOW transitions of ARDY must be performed synchronously to the CPU clock.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T_2, T_3 and again at the end of each T_W until it is sampled HIGH. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated.

This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the iAPX 186, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/Ready Logic description.

RESET Logic

The iAPX 186 provides both a $\overline{RES}$ input pin and a synchronized RESET pin for use with other system components. The $\overline{RES}$ input pin on the iAPX 186 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to two and one-half clocks behind $\overline{RES}$.

Multiple iAPX 186 processors may be synchronized through the $\overline{RES}$ input pin, since this input resets both the processor and divide-by-two internal counter in the clock generator. In order to insure that the divide-by-two counters all begin counting at the same time, the active going edge of $\overline{RES}$ must satisfy a 25 ns setup time before the falling edge of the 80186 clock input. In addition, in order to insure that all CPUs begin executing in the same clock cycle, the reset must satisfy a 25 ns setup time before the rising edge of the CLKOUT signal of all the processors.

LOCAL BUS CONTROLLER

The iAPX 186 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides control lines that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The iAPX 186 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory to the iAPX 186 or to strobe data from the iAPX 186 to memory. The ALE line provides a strobe to address latches for the multiplexed address/data bus. The iAPX 186 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, the user will have to use the $\overline{S2}$ signal (which will require external latching), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

Transceiver Control

The iAPX 186 generates two control signals to be connected to 8286/8287 transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

Table 6. Transceiver Control Signals Description

Pin Name	Function
$\overline{DEN}$ (Data Enable)	Enables the output drivers of the transceivers. It is active LOW during memory, I/O, or INTA cycles.
DT/$\overline{R}$ (Data Transmit/ Receive)	Determines the direction of travel through the transceivers. A HIGH level directs data away from the processor during write operations, while a LOW level directs data toward the processor during a read operation.

Local Bus Arbitration

The iAPX 186 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The iAPX 186 provides a single HOLD/HLDA pair through which all other bus masters may gain control of the local bus. This requires external circuitry to arbitrate which external device will gain control of the bus from the iAPX 186 when there is more than one alternate local bus master. When the iAPX 186 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$-$\overline{S2}$, $\overline{LOCK}$, AD0-AD15, A16-A19, $\overline{BHE}$, and DT/$\overline{R}$ to allow another master to drive these lines directly.

The iAPX 186 HOLD latency time, i.e., the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is the highest-priority activity request which the processor may receive: higher than instruction fetching or internal DMA cycles. However, if a DMA cycle is in progress, the iAPX 186 will complete the transfer before relinquishing the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency time can be as great as 4 bus cycles. This will occur if a DMA word transfer operation is taking place from an odd ad-

dress to an odd address. This is a total of 16 clocks or more, if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

Local Bus Controller and Reset

Upon receipt of a RESET pulse from the $\overline{RES}$ input, the local bus controller will perform the following action:

- Drive $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float.

NOTE:
$\overline{RD}$ is also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.

- Drive $\overline{S0}$–$\overline{S2}$ to the passive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- TRISTATE AD0–15, A16–19, $\overline{BHE}$, DT/$\overline{R}$.
- Drive ALE LOW (ALE is never floated).
- Drive HLDA LOW.

INTERNAL PERIPHERAL INTERFACE

All the iAPX 186 integrated peripherals are controlled via 16-bit registers contained within an internal 256-byte control block. This control block may be mapped into either memory or I/O space. Internal logic will recognize the address and respond to the bus cycle. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but D_{15-0}, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all zeros). All of the defined registers within this control block may be read or written by the 80186 CPU at any time. The location of any register contained within the 256-byte control block is determined by the current base address of the control block.

The control block base address is programmed via a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 9). It provides the upper 12 bits of the base address of the control block. The control block is effectively an internal chip select range and must abide by all the rules concerning chip selects (the chip select circuitry is discussed later in this data sheet). Any access to the 256 bytes of the control block activates an internal chip select. Other chip selects may overlap the control block only if they are programmed to zero wait states and ignore external ready. In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space, whereas if the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into iRMX mode, and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH. This causes the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 10.

The integrated iAPX 1866 peripherals operate semi-autonomously from the CPU. Access to them for the most part is via software read/write of the control block. Most of these registers can be both read and written. A few dedicated lines, such as interrupts and DMA request provide real-time communication between the CPU and peripherals as in a more conventional system utilizing discrete peripheral blocks. The overall interaction and function of the peripheral blocks has not substantially changed.

CHIP-SELECT/READY GENERATION LOGIC

The iAPX 186 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also povide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The iAPX 186 provides 6 memory chip select outputs for 3 address areas; upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address

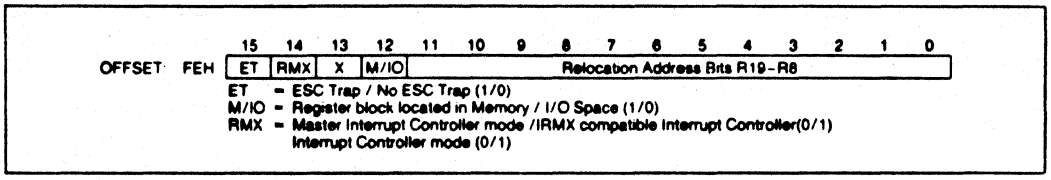


Figure 9. Relocation Register

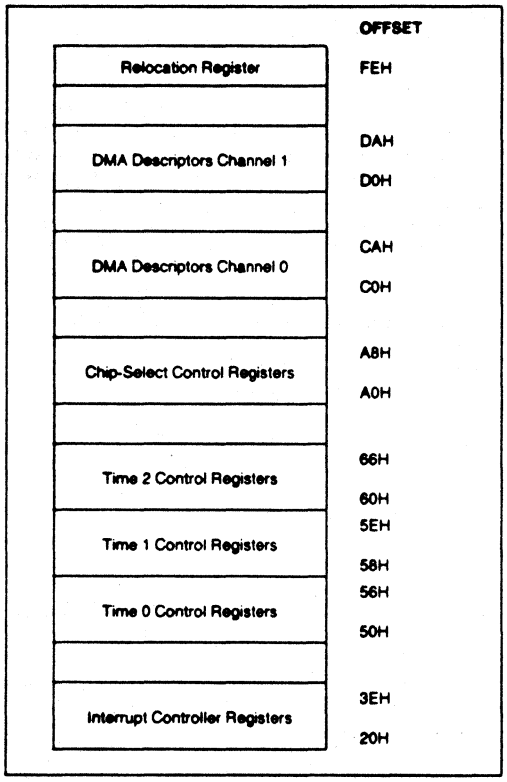


Figure 10. Internal Register Map

of the midrange memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas iAPX 186 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.

Upper Memory $\overline{CS}$

The iAPX 186 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the iAPX 186 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 7 shows the relationship between the base address selected and the size of the memory block obtained.

Table 7. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0 = R1 = R2 = 0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

The lower limit of this memory block is defined in the UMCS register (see Figure 11). This register is at offset A0H in the internal control block. The legal values for bits 6–13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6–13 not shown in Table 7 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

Any internally generated 20-bit address whose upper 16 bits are greater than or equal to UMCS (with bits 0–5 "0") will cause UCS to be activated. UMCS bits R2–R0 are used to specify READY mode for the area of memory defined by this chip-select register, as explained below.

Lower Memory $\overline{CS}$

The iAPX 186 provides a chip select for low memory called $\overline{LCS}$. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is also defined. Table 8 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 8. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0 = R1 = R2 = 0)
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

The upper limit of this memory block is defined in the LMCS register (see Figure 12). This register is at offset A2H in the internal control block. The legal values for bits 6–15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6–15 not shown in Table 8 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 0–5 "1") will cause $\overline{LCS}$ to be active. LMCS register bits R2–R0 are used to specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The iAPX 186 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located anywhere within the iAPX 186 1M byte memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the mid-range select lines, as shown in Table 9, is determined by bits 8–14 of the MPCS register (see Figure 13). This register is at location A8H in the internal control block. One and only one of bits 8–14 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. Thus, if the total block size is 32K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range and $\overline{MCS3}$ being active for the last range.

The EX and MS in MPCS relate to peripheral functionally as described in a later section.

Table 9. MPCS Programming Values

Total Block Size	Individual Select Size	MPCS Bits 14–8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

The base address of the mid-range memory block is defined by bits 15–9 of the MMCS register (see Figure 14). This register is at offset A6H in the internal control block. These bits correspond to bits A19–A13 of the 20-bit memory address. Bits A12–A0 of the base address are always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the mid-range block size is 32K (or the size of the block for which each $\overline{MCS}$ line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both of these registers is undefined. However, none of the $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.

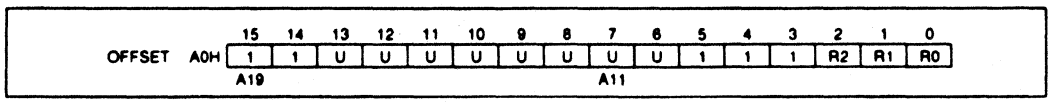


Figure 11. UMCS Register

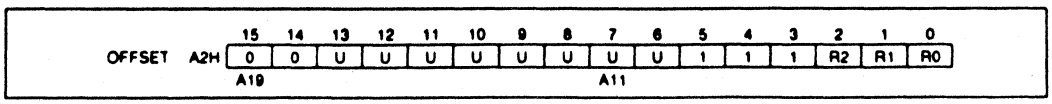


Figure 12. LMCS Register

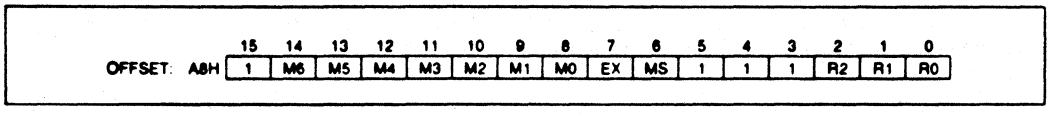


Figure 13. MPCS Register

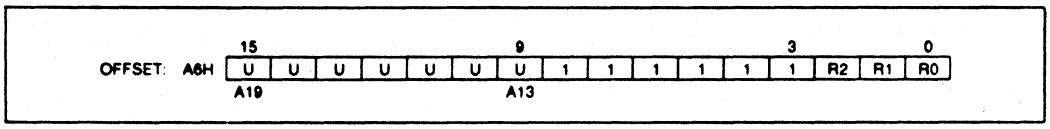


Figure 14. MMCS Register

MMCS bits R2–R0 specify READY mode of operation for all mid-range chip selects. All devices in mid-range memory must use the same number of WAIT states.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

Peripheral Chip Selects

The iAPX 186 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. This base address may be located in either memory or I/O space.

Seven $\overline{CS}$ lines called $\overline{PCS0}$–6 are generated by the iAPX 186. The base address is user-programmable; however it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

$\overline{PCS5}$ and $\overline{PCS6}$ can also be programmed to provide latched address bits A1, A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0, A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the hardware interface because the 8-bit registers of peripherals are simply treated as 16-bit registers located on even boundaries in I/O space or memory space where only the lower 8-bits of the register are significant: the upper 8-bits are "don't cares."

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 15). This register is located at offset A4H in the internal control block. Bits 15–6 of this register correspond to bits 19–10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9–0 of the PBA of the peripheral chip-select block are all zeros. If the chip-select block is located in I/O space, bits 12–15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

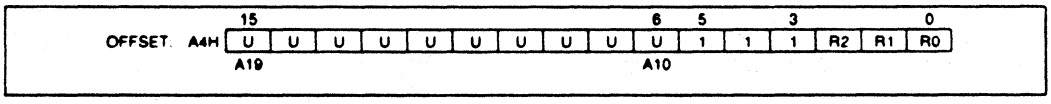


Figure 15. PACS Register

The user should program bits 15-6 to correspond to the desired peripheral base location. PACS bits 0-2 are used to specify READY mode for $\overline{PSC0}$-$\overline{PCS3}$.

Table 10. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA —PBA + 127
PCS1	PBA + 128—PBA + 255
PCS2	PBA + 256—PBA + 383
PCS3	PBA + 384—PBA + 511
PCS4	PBA + 512—PBA + 639
PCS5	PBA + 640—PBA + 767
PCS6	PBA + 768—PBA + 895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 16). This register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined, however none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 11. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space.
	0 = Peripherals mapped into I/O space.
EX	0 = 5 $\overline{PCS}$ lines. A1, A2 provided.
	1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided.

MPCS bits 0-2 are used to specify READY mode for $\overline{PCS4}$-$\overline{PCS6}$ as outlined below.

READY Generation Logic

The iAPX 186 can generate a "READY" signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0-3 wait states for all accesses to the area for which the chip select is active. In addition, the iAPX 186 may be programmed to either ignore external READY for each chip-select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each $\overline{CS}$ line or group of lines generated by the iAPX 186. The interpretation of the ready bits is shown in Table 12.

Table 12. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). This means, for example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

R2-R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2-R0 of PACS set the $\overline{PCS0}$-3 READY mode, R2-R0 of MPCS set the $\overline{PCS4}$-6 READY mode.

Chip Select/Ready Logic and Reset

Upon reset, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven HIGH.
- Upon leaving RESET, the $\overline{UCS}$ line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to allow the maximum number of internal wait states in conjunction with external Ready consideration (i.e., UMCS resets to FFFBH).
- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

OFFSET: A8H	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Figure 16. MPCS Register

DMA CHANNELS

The 80186 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data. This provides a maximum data transfer rate of one Mword/sec or 2 MBytes/sec.

DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit destination pointer (2 words), a 16-bit Transfer Counter, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 13. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 18). All registers may be modified or altered during any DMA activity. Any changes made to these registers will be reflected immediately in DMA operation.

Table 13. DMA Control Block Format

Register Name	Register Address	
	Ch. 0	Ch. 1
Control Word	CAH	DAH
Transfer Count	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	C0H	D0H

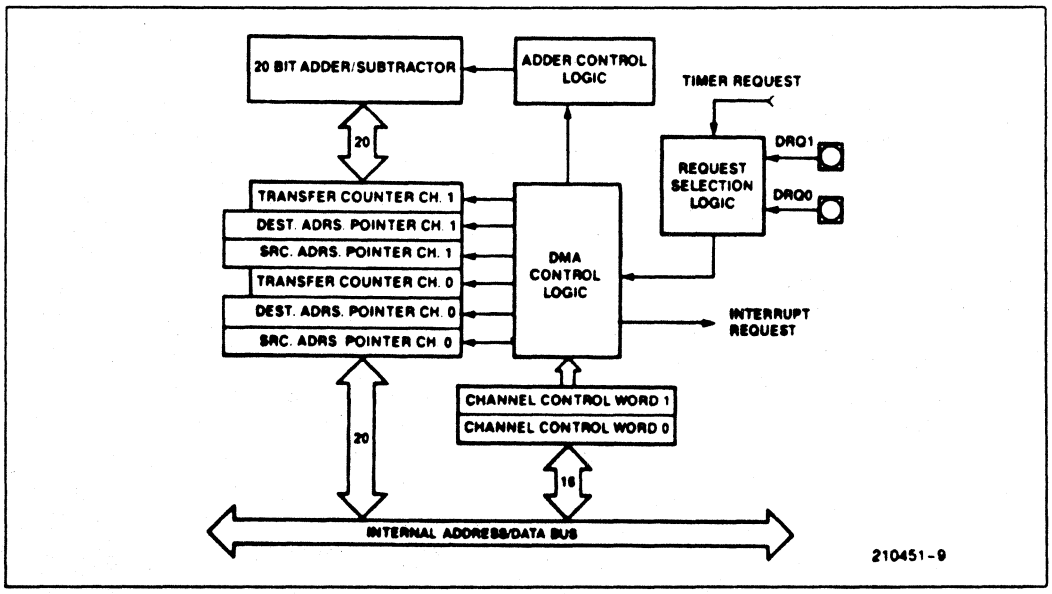


Figure 17. DMA Unit Block Diagram

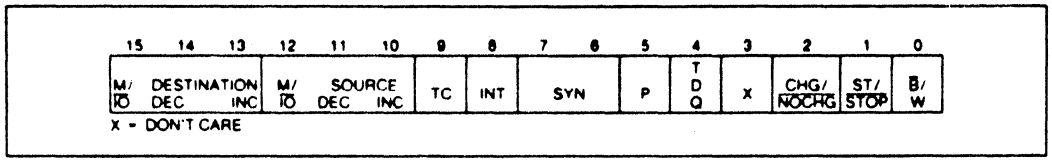


Figure 18. DMA Control Register

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 81086 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

DMA Control Word Bit Descriptions

$\overline{B}$/W: Byte/Word (0/1) Transfers.

ST/$\overline{STOP}$: Start/stop (1/0) Channel.

CHG/$\overline{NOCHG}$: Change/Do not change (1/0) ST/$\overline{STOP}$ bit. If this bit is set when writing to the control word, the ST/$\overline{STOP}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{STOP}$ bit will not be altered. This bit is not stored; it will always be a 0 on read.

INT: Enable Interrupts to CPU on Transfer Count termination.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach zero. The ST/$\overline{STOP}$ bit will also be reset at this point if TC is set. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach zero.

SYN (2 bits)
00 No synchronization.

NOTE: The ST bit will be cleared automatically when the contents of the TC register reach zero regardless of the state of the TC bit.

01 Source synchronization.

10 Destination synchronization.

11 Unused.

SOURCE:INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Source pointer is in M/IO space (1/0).

DEC Decrement source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

DEST: INC Increment destination pointer by 1 or 2 ($\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Destination pointer is in M/IO space (1/0).

DEC Decrement destination pointer by 1 or 2 (depending on $\overline{B}$/W) after each transfer.

P Channel priority—relative to other channel.

0 low priority.

1 high priority.

Channels will alternate cycles if both set at same priority level.

TDRQ 0: Disable DMA requests from timer 2.

1: Enable DMA requests from timer 2.

Bit 3 Bit 3 is not used.

If both INC and DEC are specified for the same pointer, the pointer will remain constant after each cycle.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 18a). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed the pointer is incremented or decremented by two. Each pointer may point into either memory or I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be obtained if all word transfers are performed to even addresses, since this will allow data to be accessed in a single memory access.

DMA Transfer Count Register

Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set or unsynchronized transfers are programmed, however, DMA activity will terminate when the transfer count register reaches zero.

DMA Requests

Data transfers may be either source or destination synchronized, that is either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows a complete transfer to take place every 2 bus cycles or eight clock cycles (assuming no wait states). No prefetching occurs when destination synchronization is performed, however. Data will not be fetched from the source address until the destination device signals that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This is done to allow the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination synchronized transfers. These lead to the maximum DMA transfer rates shown in Table 14.

Table 14. Maximum DMA Transfer Rates

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	2MBytes/sec	2MBytes/sec
Source Synch	2MBytes/sec	2MBytes/sec
Destination Synch	1.3MBytes/sec	1.5MByte/sec

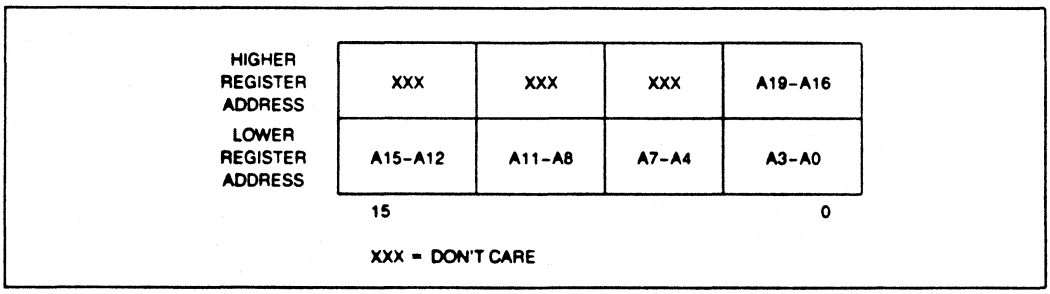


Figure 18a. DMA Memory Pointer Register Format

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses the odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/STOP bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also have been generated. Therefore the source and destination transfer pointers, and the transfer count register (if used) must be programmed before this bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/NOCHG bit is cleared when the control register is written, the ST/STOP bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

Upon RESET, the DMA channels will perform the following actions:

- The Start/Stop bit for each channel will be reset to STOP.
- Any transfer in progress is aborted.

TIMERS

The 80186 provides three internal 16-bit programmable timers (see Figure 19). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

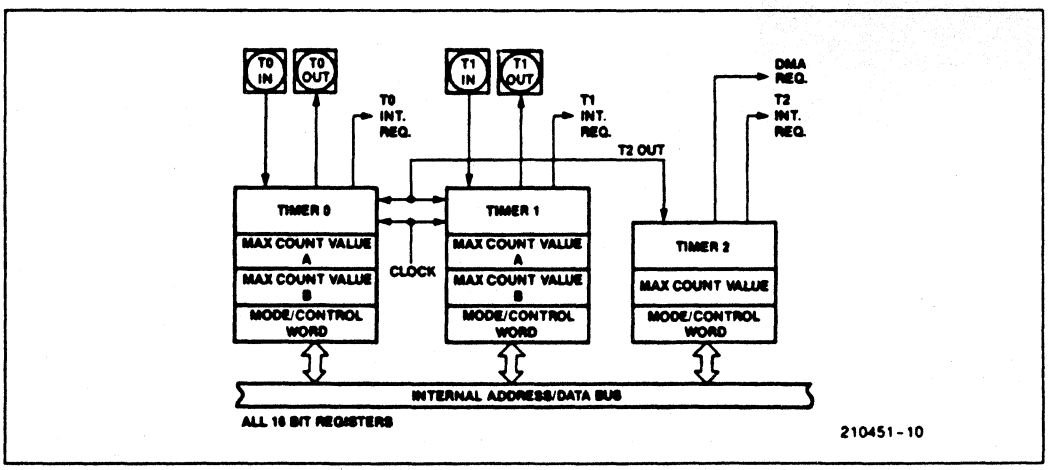


Figure 19. Timer Block Diagram

Timer Operation

The timers are controlled by 11 16-bit registers in the internal peripheral control block. The configuration of these registers is shown in Table 15. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values programmed by the user. If a single MAX COUNT register is used, the timer output pin will switch LOW for a single clock, 1 clock after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate (2 MHz for an 8 MHz CPU clock). Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to 6 clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16 bits wide, 16 bits of resolution are provided. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle, however. This is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

The timers have several programmable options.

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

Timer Mode/Control Register

The mode/control register (see Figure 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 15. Timer Control Block Format

Register Name	Register Offset		
	Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

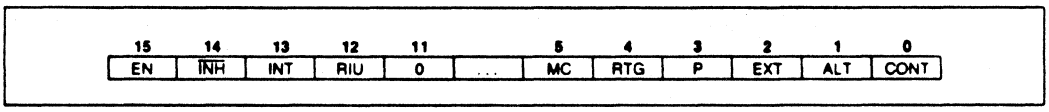


Figure 20. Timer Mode/Control Register

ALT:

The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating non-repetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is zero, the output pin will go LOW for one clock, the clock after the maximum count is reached. If ALT is one, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT:

Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If COUNT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

EXT:

The external bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80186 clock. If this bit is set, the timer will count LOW-to-HIGH transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as 6 clocks. However, clock inputs may be pipelined as closely together as every 4 clocks without losing clock pulses.

P:

The prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, the output of timer 2 will be used as a clock for the timer. Note that the user must initialize and start timer 2 to obtain the prescaled clock.

RTG:

Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is HIGH, the timer will count; if the input pin is LOW, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80186 clock.

When RTG = 1, the input pin detects LOW-to-HIGH transitions. The first such transition starts the timer running, clearing the timer value to zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

EN:

The enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transistions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

INH:

The inhibit bit allows for selective updating of the enable (EN) bit. If INH is a one during the write to the mode/control word, then the state of the EN bit will be modified by the write. If INH is a zero during the write, the EN bit will be unaffected by the operation.This bit is not stored; it will always be a 0 on a read.

INT:

When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller).

MC:

The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set

regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

Programmer intervention is required to clear this bit.

RIU:

The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A zero value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is zero.

Not all mode bits are provided for timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

Count Registers

Each of the three timers has a 16-bit count register. The current contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting,the new value will take effect in the current count cycle.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two max count values whenever the current maximum count is reached. The condition which causes a timer to reset is equivalent between the current count value and the max count being used. This means that if the count is changed to be above the max count value, or if the max count value is changed to be below the current value, the timer will not reset to zero, but rather will count to its maximum value, "wrap around" to zero, then count until the max count is reached.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

INTERRUPT CONTROLLER

The 80186 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU.

Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80186 interrupt controller has its own control register that set the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The interrupt controller has a special iRMX 86 compatibility mode that allows the use of the 80186 within the iRMX 86 operating system interrupt structure. The conroller is set in this mode by setting bit 14 in the peripheral control block relocation register (see iRMX 86 Compatibility Mode section). In this mode, the internal 80186 interrupt controller functions as a "slave" controller to an external "master" controller. Special initialization software must be included to properly set up the 80186 interrupt controller in iRMX 86 mode.

MASTER MODE OPERATION

Interrupt Controller External Interface

For external interrupt sources, five dedicated pins are provided. One of these pins is dedicated to NMI, non-maskable interrupt. This is typically used for power-fail interrupts, etc. The other four pins may function either as four interrupt input lines with internally generated interrupt vectors, as an interrupt line and an interrupt acknowledge line (called the "cascade mode") along with two other input lines with internally generated interrupt vectors, or as two interrupt input lines and two dedicated interrupt acknowledge output lines. When the interrupt lines are configured in cascade mode, the 80186 interrupt controller will not generate internal interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the vector is read into the 80186 on the second cycle. The capability to interface to external 8259A programmable interrupt controllers is thus provided when the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in master mode are similar to the 8259A. The interrupt controller responds indentically to internal interrupts in all three modes: the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

Fully Nested Mode

When in the fully nested mode four pins are used as direct interrupt requests. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the in-service bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled without being themselves interrupted by lower-priority interrupts. Since interrupts are enabled, higher-priority interrupts will be serviced.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is issued at the end of the service routine just before the issuance of the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode

The 80186 has four interrupt pins and two of them have dual functions. In the fully nested mode the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 22. INT0 is an interrupt input interfaced to an 8259A, while INT2/$\overline{\text{INTA0}}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{\text{INTA1}}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{\text{INTA}}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 8259As. Three levels of priority are created, requiring priority resolution in the 80186 interrupt controller, the master 8259As, and the slave 8259As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

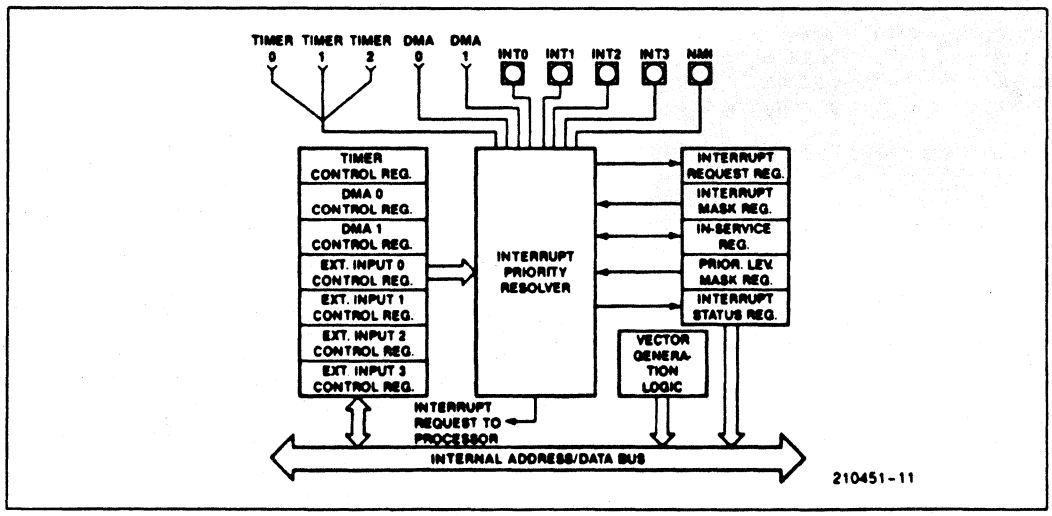


Figure 21. Interrupt Controller Block Diagram

Special Fully Nested Mode

This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 8259A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80186 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80186 controller until the 80186 in-service bit is reset. In special fully nested mode, the 80186 interrupt controller will allow interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80186 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the external master's IS register is required to determine if there is more than one bit set. If so, the IS bit in the 80186 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 31). Bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Bits 0–4 indicate to the processor the type vector of the highest-priority source requesting service. Reading the Poll Word causes the In-Service bit of the highest priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, i.e., not set the indicated in-service bit. The 80186 provides a Poll Status Word in addition to the conventional Poll Word to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

Master Mode Features

Programmable Priority

The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority). All interrupt sources have preprogrammed default priority levels (see Table 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, it allows other requests to be serviced.

End-of-Interrupt Command

The end-of-interrupt (EOI) command is used by the programmer to reset the In-Service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and nonspecific. The nonspecific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode

The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active HIGH. In the edge sense mode or the level-trigger mode, the interrupt request must remain active (HIGH) until the interrupt request is acknowledged by the 80186 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go LOW for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made: holding the interrupt input HIGH will cause continuous interrupt requests.

Interrupt Vectoring

The 80186 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 23. It contains 15 registers. All registers can both be read or written unless specified otherwise.

In-Service Register

This register can be read from or written into. The format is shown in Figure 24. It contains the In-Service bit for each of the interrupt sources. The In-Service bit is set to indicate that a source's service routine is in progress. When an In-Service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the In-Service bit for all three timers; the D0 and D1 bits are the In-Service bits for the two DMA channels; the I0–I3 are the In-Service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or bv reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command issued by the CPU.

Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 24. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller, therefore the external interrupt bits cannot be written. The external interrupt request bits show exactly when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be HIGH only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Mask Register

This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 24. A one in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

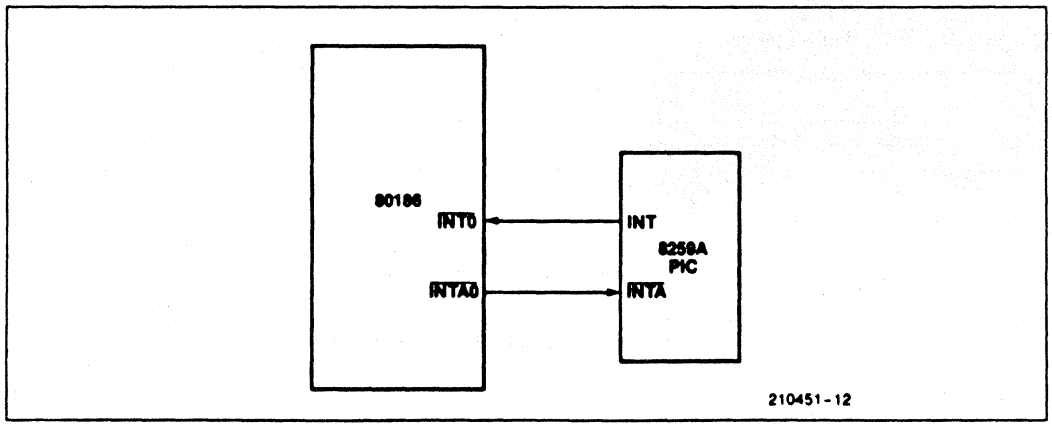


Figure 22. Cascade Mode Interrupt Connection

	OFFSET
INT3 CONTROL REGISTER	3EH
INT2 CONTROL REGISTER	3CH
INT1 CONTROL REGISTER	3AH
INT0 CONTROL REGISTER	38H
DMA 1 CONTROL REGISTER	36H
DMA 0 CONTROL REGISTER	34H
TIMER CONTROL REGISTER	32H
INTERRUPT STATUS REGISTER	30H
INTERRUPT REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY MASK REGISTER	2AH
MASK REGISTER	28H
POLL STATUS REGISTER	26H
POLL REGISTER	24H
EOI REGISTER	22H

Figure 23. Interrupt Controller Registers (Non-iRMX™ 86 Mode)

Priority Mask Register

This register is used to mask all interrupts below particular interrupt priority levels. The format of this register is shown in Figure 25. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so all interrupts are unmasked.

Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 26. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. The purpose of this bit is to allow prompt service of all non-maskable interrupts. This bit may also be set by the CPU.

IRTx: These three bits represent the individual timer interrupt request bits. These bits are used to differentiate the timer interrupts, since the timer IR bit in the interrupt request register is the "OR" function of all timer interrupt request. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

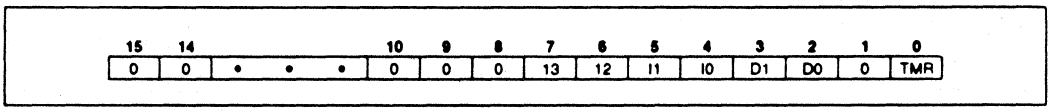


Figure 24. In-Service, Interrupt Request, and Mask Register Formats

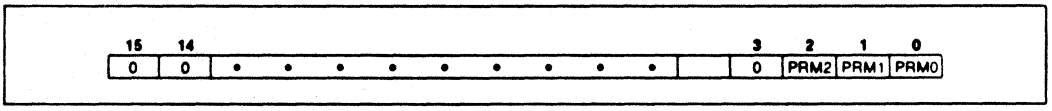


Figure 25. Priority Mask Register Format

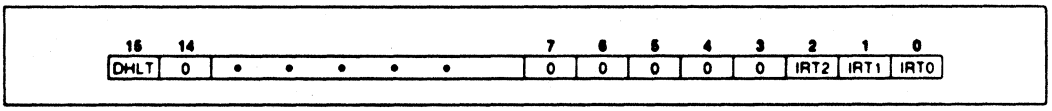


Figure 26. Interrupt Status Register Format

Timer, DMA 0, 1; Control Register

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 27. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

INT0-INT3 Control Registers

These registers are the control words for the four external input pins. Figure 28 shows the format of the INT0 and INT1 Control registers; Figure 29 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PRO-2: Priority programming information. Highest Priority = 000, Lowest Priority = 111

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt Input levels are active high. In level-triggered mode, an interrupt is generated whenever the external line is high. In edge-triggered mode, an interrupt will be generated only when this level is proceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit, 1 = mask; 0 = non-mask.

C: Cascade mode bit, 1 = cascade; 0 = direct

SFNM: Special fully nested mode bit, 1 = SFNM

EOI Register

The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 30. It initiates an EOI command when written to by the 80186 CPU.

The bits in the EOI register are encoded as follows:

S_X: Encoded information that specifies an interrupt source vector type as shown in Table 4. For example, to reset the In-Service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10. Note that to reset the single In-Service bit for any of the three timers, the vector type for timer 0 (8) should be written in this register.

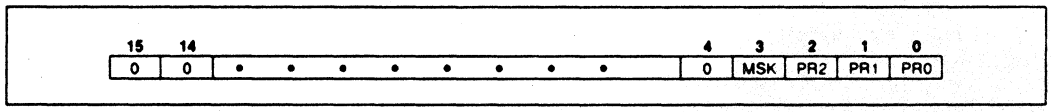


Figure 27. Timer/DMA Control Registers Formats

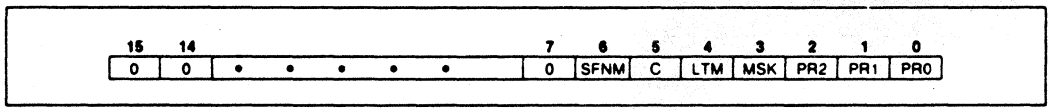


Figure 28. INT0/INT1 Control Register Formats

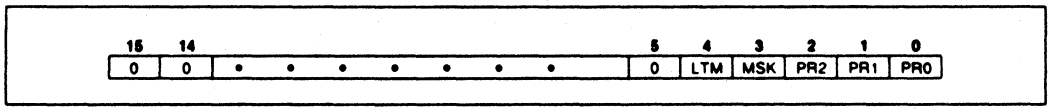


Figure 29. INT2/INT3 Control Register Formats

NSPEC/SPEC: A bit that determines the type of EOI command. Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 31. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_x: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

iRMX™ 86 COMPATIBILITY MODE

This mode allows iRMX 86-80186 compatibility. The interrupt model of iRMX 86 requires one master and multiple slave 8259As in cascaded fashion. When iRMX mode is used, the internal 80186 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80186 resources will be monitored through the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80186 interrupt controller will be in the non-iRMX 86 mode of operation. To set the controller in the iRMX 86 mode, bit 14 of the Relocation Register should be set.

Because of pin limitations caused by the need to interface to an external 8259A master, the internal interrupt controller will no longer accept external inputs. There are however, enough 80186 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The iRMX 86 operating system requires peripherals to be assigned fixed priority levels. This is incompatible with the normal operation of the 80186 interrupt controller. Therefore, the initialization software must program the proper priority levels for each source. The required priority levels for the internal interrupt sources in iRMX mode are shown in Table 16.

Table 16. Internal Source Priority Level

Priority Level	Interrupt Source
0	Timer 0
1	(reserved)
2	DMA 0
3	DMA 1
4	Timer 1
5	Timer 2

These level assignments must remain fixed in the iRMX 86 mode of operation.

iRMX™ 86 Mode External Interface

The configuration of the 80186 with respect to an external 8259A master is shown in Figure 32. The INT0 input is used as the 80186 CPU interrupt input. INT3 functions as an output to send the 80186 slave-interrupt-request to one of the 8 master-PIC-inputs.

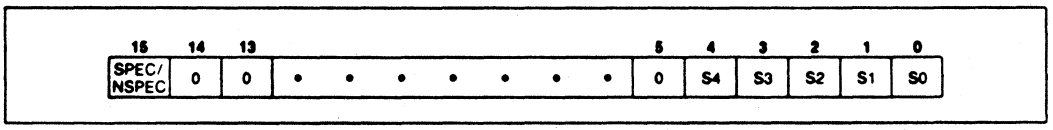


Figure 30. EOI Register Format

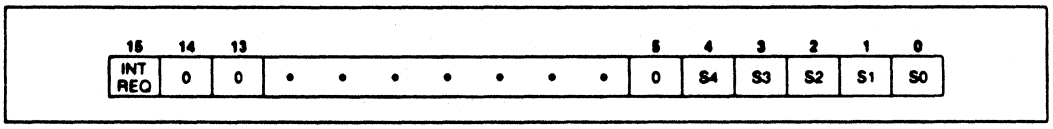


Figure 31. Poll Register Format

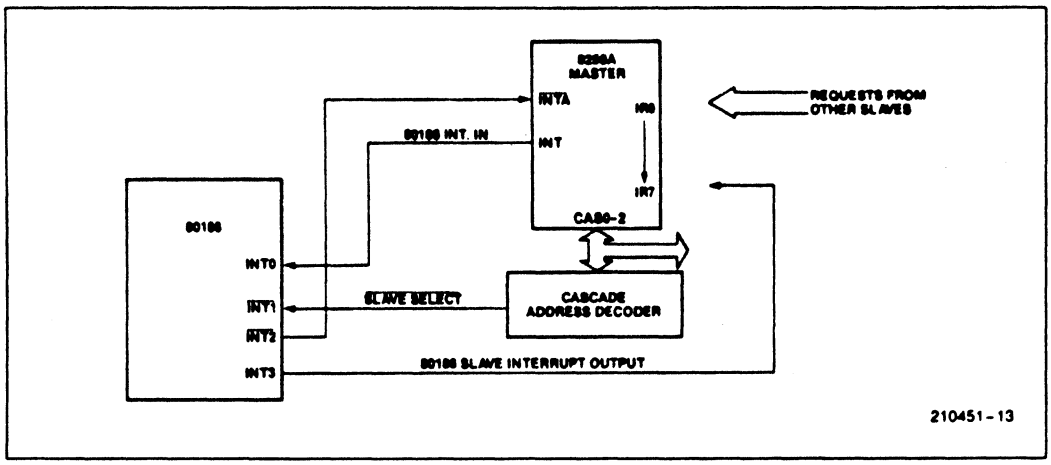


Figure 32. iRMX™ 86 Interrupt Controller Interconnection

Correct master-slave interface requires decoding of the slave addresses (CAS0-2). Slave 8259As do this internally. Because of pin limitations, the 80186 slave address will have to be decoded externally. $\overline{INT1}$ is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

$\overline{INT2}$ is used as an acknowledge output, suitable to drive the $\overline{INTA}$ input of an 8259A.

Interrupt Nesting

iRMX 86 mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the iRMX™ 86 Mode

Vector generation in iRMX mode is exactly like that of an 8259A slave. The interrupt controller generates an 8-bit vector which the CPU multiplies by four and uses as an address into a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In iRMX mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the iRMX™ 86 Mode

All control and command registers are located inside the internal peripheral control block. Figure 33 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 34. It initiates an EOI command when written by the 80186 CPU.

The bits in the EOI register are encoded as follows:

L_X: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal interrupt sources. The format for this register is shown in Figure 35. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register

This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 35. The interrupt request bits are set when a request arrives from an internal source, and are reset when the processor acknowledges the request.

Mask Register

The register contains a mask bit for each interrupt source. The format for this register is shown in Figure 35. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers, i.e., changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

Control Registers

These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 36. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_x: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

msk: mask bit for the priority level indicated by pr_x bits.

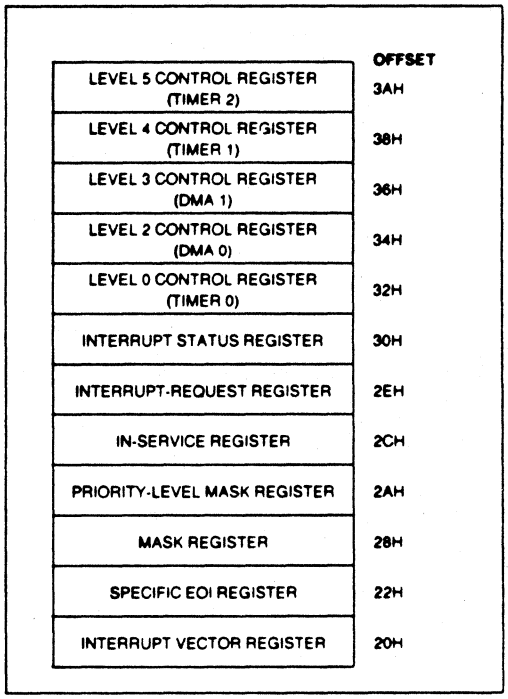


Figure 33. Interrupt Controller Registers (iRMX™ 86 Mode)

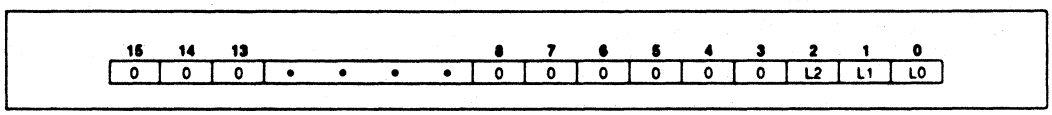


Figure 34. Specific EOI Register Format

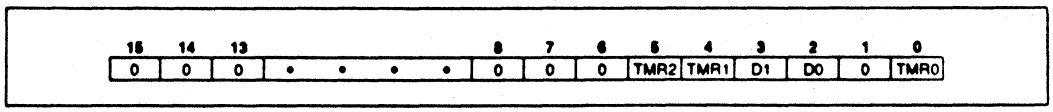


Figure 35. In-Service, Interrupt Request, and Mask Register Format

Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 37. The interrupt controller itself provides the lower three bits of the interrupt vector as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register

This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

Interrupt Status Register

This register is defined exactly as in Non-iRMX Mode. (See Fig. 26.)

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. This places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to non-iRMX 86 mode.

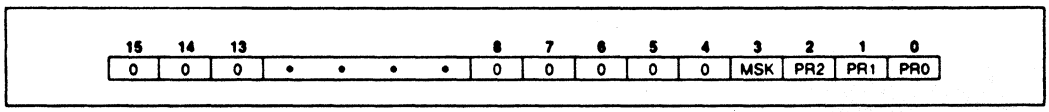


Figure 36. Control Word Format

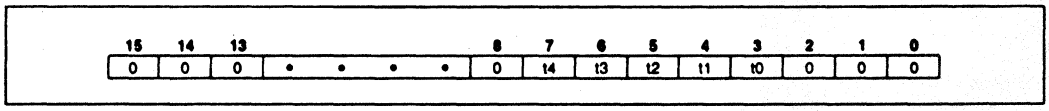


Figure 37. Interrupt Vector Register Format

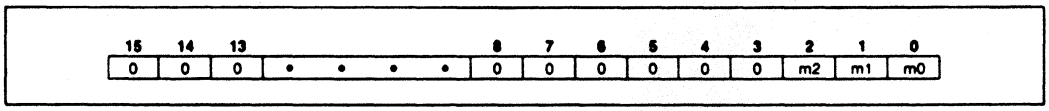


Figure 38. Priority Level Mask Register

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

DESCRIPTION

The Signetics SCN2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SCN2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2681 is available in three package versions to satisfy various system requirements: 40-pin and 28-pin, both 0.6" wide DIPs, and a compact 24-pin, 0.4" wide, DIP.

FEATURES

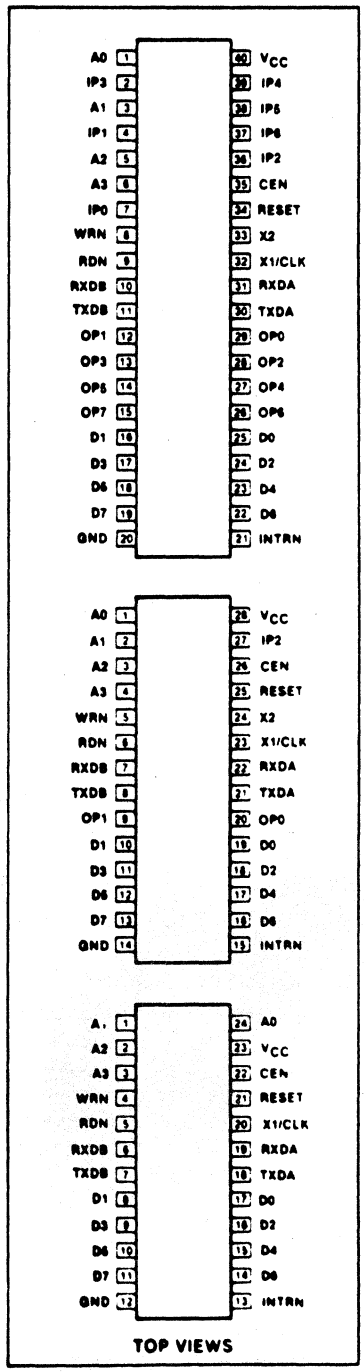
- Dual full-duplex asynchronous receiver/transmiter
- Quadruple buffered receiver data registers
- Programmable data format
  - —5 to 8 data bits plus parity
  - —Odd, even, no parity or force parity
  - —1, 1.5 or 2 stop bits programmable in 1/16 bit increments
- Programmable baud rate for each receiver and transmiter selectable from:
  - —18 fixed rates: 50 to 38.4K baud
  - —One user defined rate derived from programmable timer/counter
  - —External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - —Normal (full duplex)
  - —Automatic echo
  - —Local loopback
  - —Remote loopback
- Multi-function programmable 16-bit counter/timer
- Multi-function 7-bit input port
  - —Can serve as clock or control inputs
  - —Change of state detection on four inputs
- Multi-function 8-bit output port
  - —Individual bit set/reset capability
  - —Outputs can be programmed to be status/interrupt signals
- Versatile interrupt system
  - —Single interrupt output with eight maskable interrupting conditions
  - —Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs
- Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec
- Automatic wake-up mode for multidrop applications
- Start-end break interrupt/status
- Detects break which originates in the middle of a character
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply

ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C		
	24 Pin[1]	28 Pin[2]	40 Pin[2]
Ceramic DIP	Not available	SCN2681AC1I28	SCN2681AC1I40
Plastic DIP	SCN2681AC1N24	SCN2681AC1N28	SCN2681AC1N40

[1] 400 mil wide DIP
[2] 600 mil wide DIP

PIN CONFIGURATION

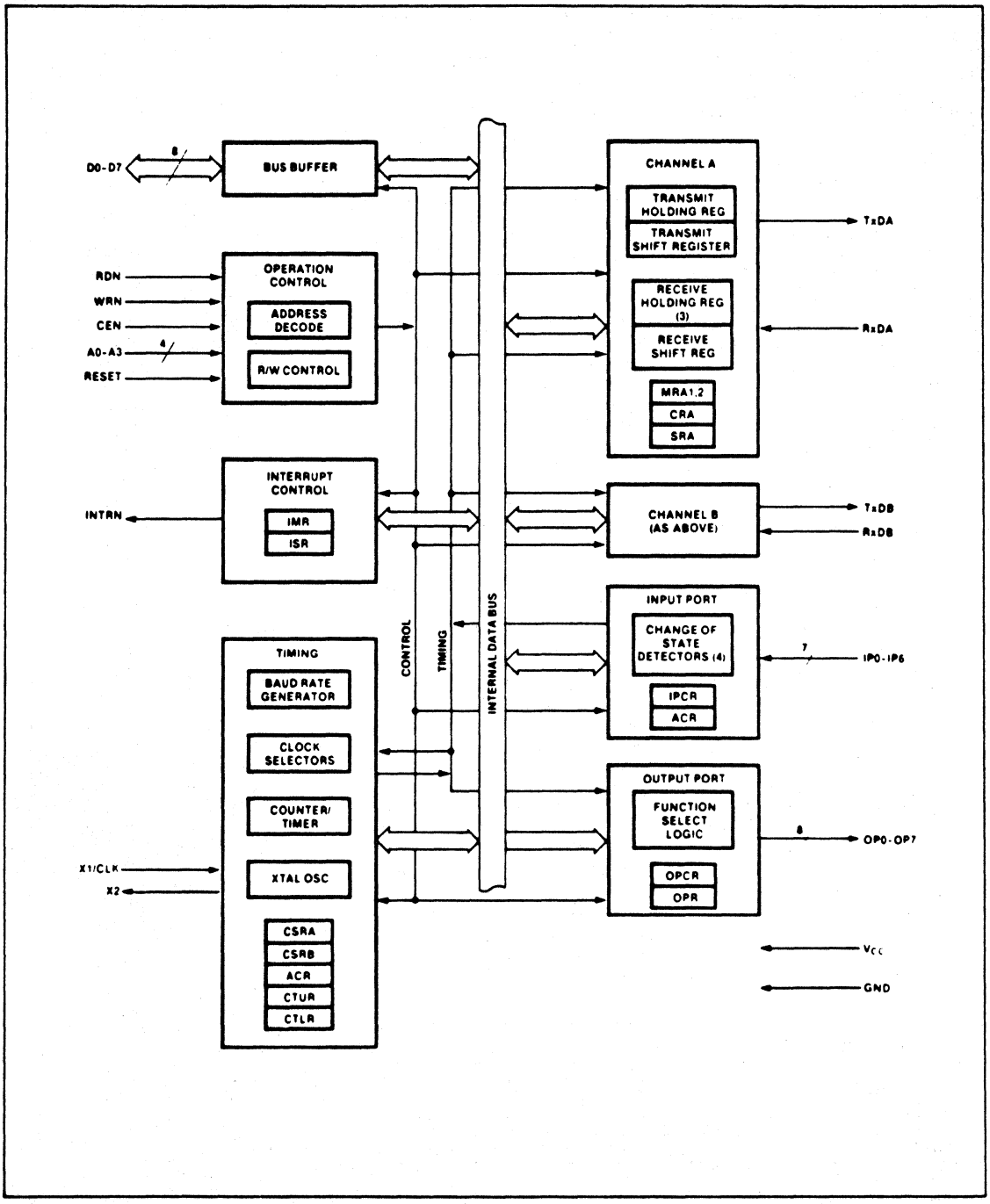


TOP VIEWS

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

BLOCK DIAGRAM



DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

PIN DESIGNATION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0-D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the WRN, RDN and A0-A3 inputs. When high, places the D0-D7 lines in the 3-state condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0-A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0-OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
X2	X	X		O	**Crystal 2:** Connection for other side of the crystal. Should be connected to ground if a crystal is not used. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN)
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN)
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

PIN DESIGNATION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** +5V supply input
GND	X	X	X	I	**Ground**

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3-OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the 2681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. D_7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50μs will set the corresponding bit in the input port will change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Preliminary

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. If it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least sigificant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled)

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected. The character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the

Preliminary

receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the ac-

Table 1 2681 REGISTER ADDRESSING

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

Preliminary

Table 2 REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA CSRB	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA CRB	not used—must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes*	0 = no 1 = yes*	0 = no 1 = yes*	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/FFULLB	0 = OPR[4] 1 = RxRDY/FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Table 2 REGISTER BIT FORMATS (Continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

cumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.

3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

	Baud Rate CLOCK = 3.6864MHz	
CSRA[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRA[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[7:4]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP6—16X	IP6—16X
1 1 1 1	IP6—1X	IP6—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

	Baud Rate	
CSRB[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

Preliminary

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received. further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock).

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled. Characters loaded into the THR while the transmitter is disabled will not be transmitted.

Preliminary

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng.

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

Table 3 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE
Duty cycle of 16X clock is 50% ± 1%

Preliminary

ACR[6:4] — Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IP0 Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

Table 4 ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

Preliminary

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		4.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

NOTES

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6,7]

PARAMETER		TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset Timing (figure 1)					
t_{RES}	RESET pulse width	1.0			µs
Bus Timing (figure 2)[8]					
t_{AS}	A0-A3 setup time to RDN, WRN low	10			ns
t_{AH}	A0-A3 hold time from RDN, WRN high	0			ns
t_{CS}	CEN setup time to RDN, WRN low	0			ns
t_{CH}	CEN hold time from RDN, WRN high	0			ns
t_{RW}	WRN, RDN pulse width	225			ns
t_{DD}	Data valid after RDN low			175	ns
t_{DF}	Data bus floating after RDN high			100	ns
t_{DS}	Data setup time before WRN high	100			ns
t_{DH}	Data hold time after WRN high	20			ns
t_{RWD}	High time between READs and/or WRITEs[9,10]	200			ns
Port Timing (figure 3)[8]					
t_{PS}	Port input setup time before RDN low	0			ns
t_{PH}	Port input hold time after RDN high	0			ns
t_{PD}	Port output valid after WRN high			400	ns
Interrupt Timing (figure 4)					
t_{IR}	INTRN (or OP3-OP7 when used as interrupts) high from:				
	Read RHR (RXRDY/FFULL interrupt)			300	ns
	Write THR (TXRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (clear of interrupt mask bit)			300	ns
Clock Timing (figure 5)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC}	CTCLK (IP2) high or low time	100			ns
f_{CTC}	CTCLK (IP2) frequency	0		4.0	MHz
t_{RX}	RxC high or low time	220			ns
f_{RX}	RxC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
t_{TX}	TxC high or low time	220			ns
f_{TX}	TxC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
Transmitter Timing (figure 6)					
t_{TXD}	TxD output delay from TxC low			350	ns
t_{TCS}	TxC output skew from TxD output data	0		150	ns
Receiver Timing (figure 7)					
t_{RXS}	RxD data setup time to RXC high	240			ns
t_{RXH}	RxD data hold time from RXC high	200			ns

NOTES

4 Parameters are valid over specified temperature range

5 All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate

6 Typical values are at +25°C, typical supply voltages, and typical processing parameters

7 Test condition for outputs $C_L = 150pF$ except interrupt outputs. Test condition for interrupt outputs $C_L = 50pF$, $R_L = 2.7K$ ohm to V_{CC}

8 Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the strobing input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN

9 If CEN is used as the strobing input, this parameter defines the minimum high time between one CEN and the next

10 Consecutive write operations to the same command register require at least three edges of the X1 clock between writes

Preliminary

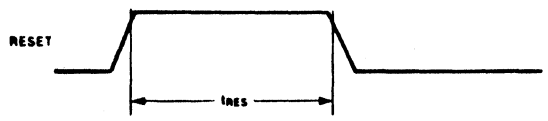


Figure 1. Reset Timing

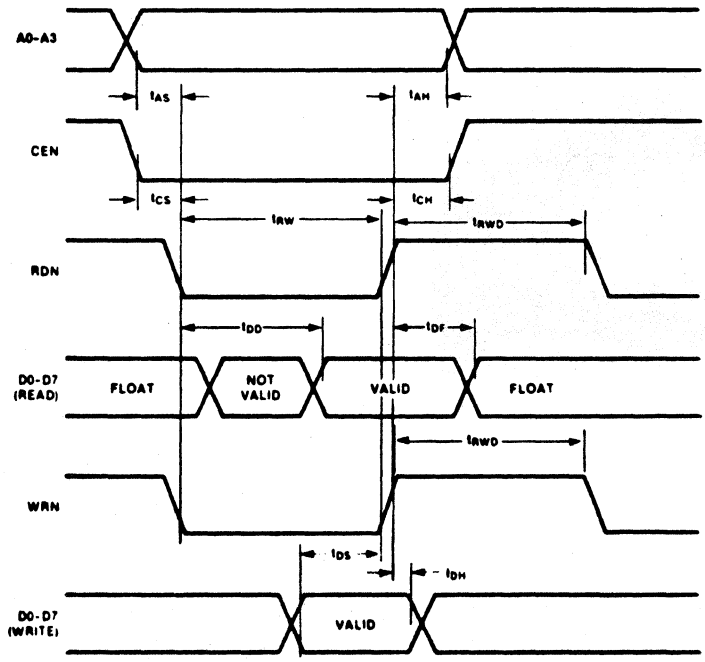


Figure 2. Bus Timing

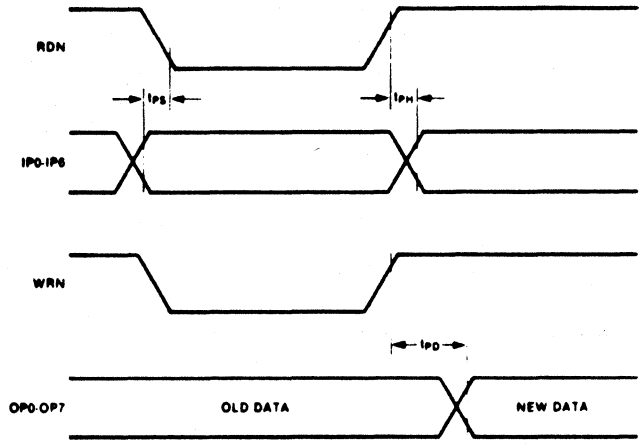


Figure 3. Port Timing

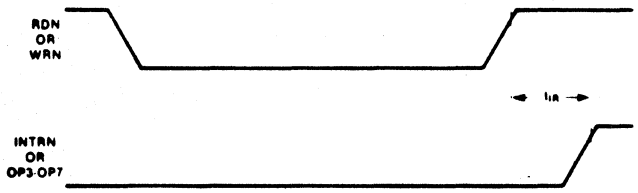
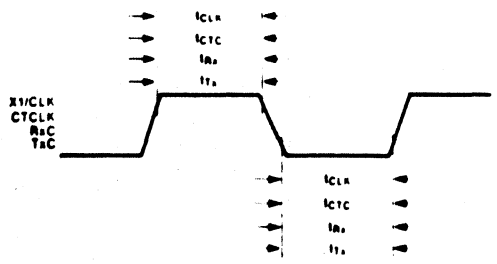


Figure 4. Interrupt Timing



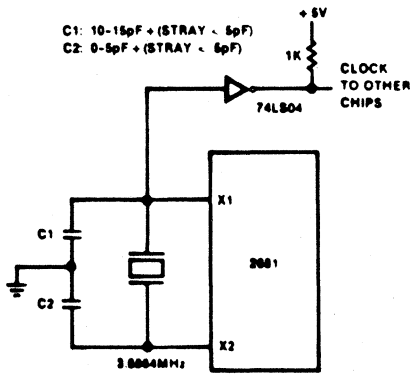


Figure 5. Clock Timing

Preliminary

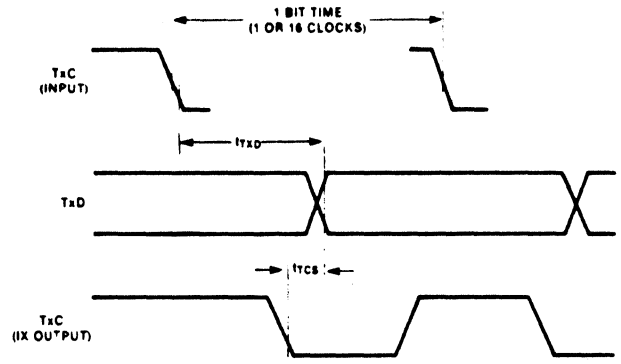


Figure 6. Transmit

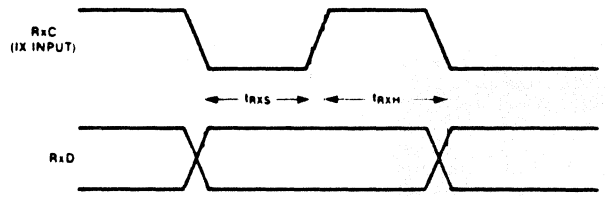


Figure 7. Receive

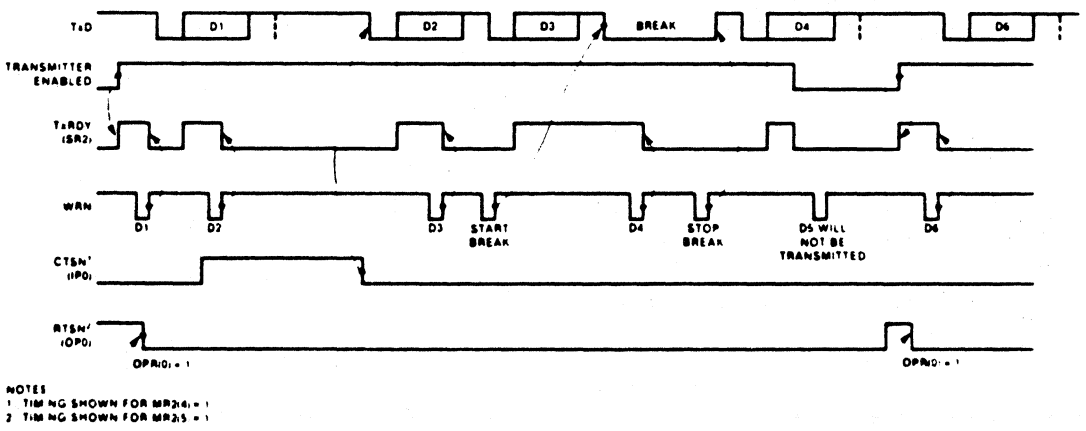
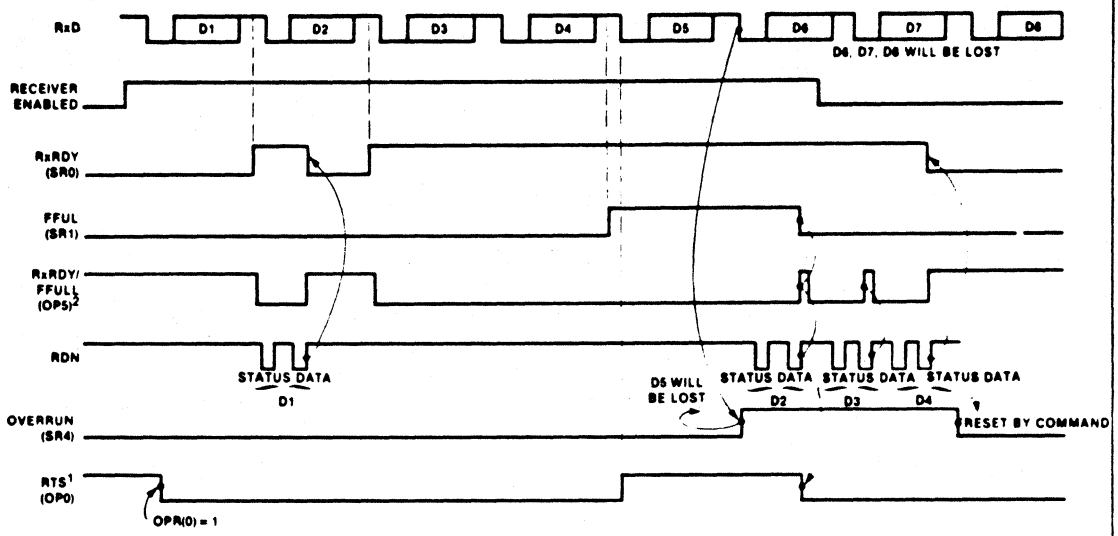


Figure 8. Transmitter Timing



NOTES
1. TIMING SHOWN FOR MR1(7) = 1.
2. SHOWN FOR OPCR(4) = 1 AND MR(6) = 0.

Figure 9. Receiver Timing

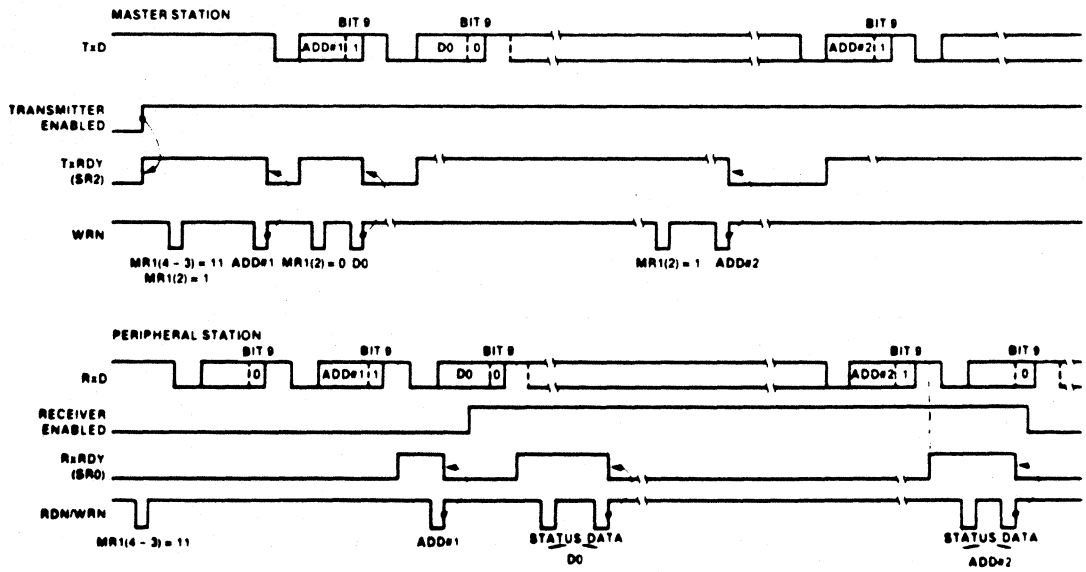


Figure 10. Wake Up Mode

NCR 5380 SCSI INTERFACE
PRODUCT BRIEF

SCSI INTERFACE

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

MPU INTERFACE

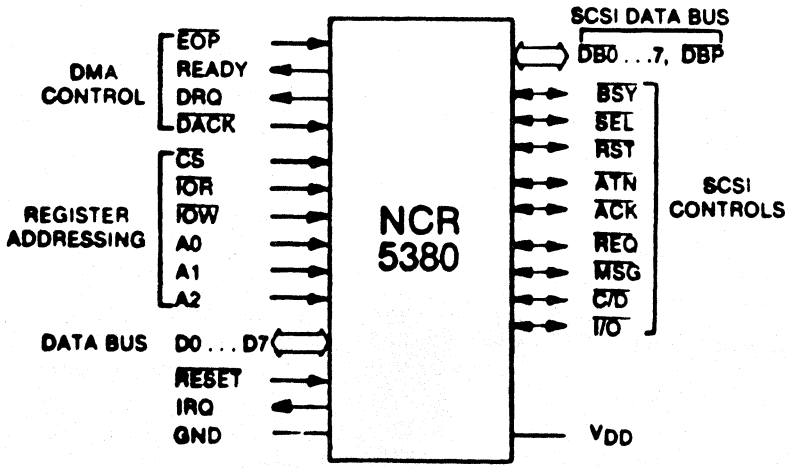
- Memory or I/O mapped interface
- DMA or programmed I/O
- Normal or block mode DMA
- Optional MPU interrupts

The NCR 5380 is designed to accommodate the Small Computer Systems Interface (SCSI) as defined by the ANSI X3T9.2 committee. The 5380 operates in both the Initiator and Target roles and can therefore be used in host adapter and control unit designs. This device supports arbitration, including reselection, and is intended to be used in systems that require either open collector or differential pair transceivers.* It has special high current outputs for driving the SCSI bus directly in the open collector mode.

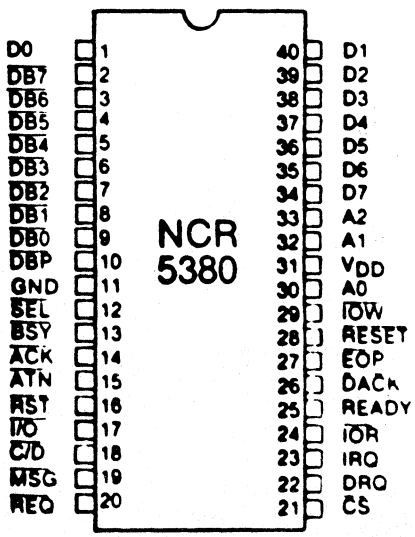
The NCR 5380 communicates with the system microprocessor as a peripheral device. The chip is controlled by reading and writing several internal registers which may be addressed as standard or memory mapped I/O. Minimal processor intervention is required for DMA transfers because the 5380 controls the necessary handshake signals. The NCR 5380 interrupts the MPU when it detects a bus condition that requires attention. Normal and block mode DMA is provided to match many popular DMA controllers.

* Differential pair operation is supported in the NCR 5381 (48 PIN).

FUNCTIONAL PIN GROUPING



PINOUT



NCR

PIN DESCRIPTIONS

MICROPROCESSOR INTERFACE SIGNALS

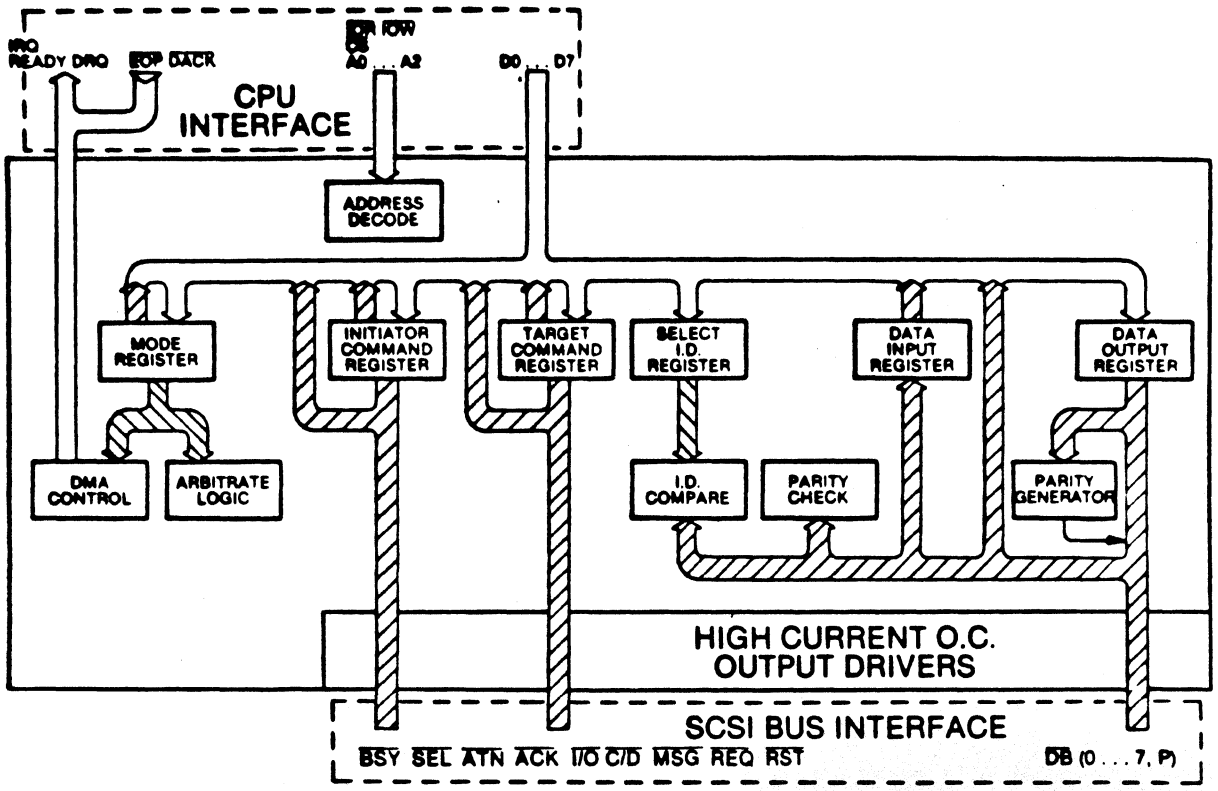
Pin Name	Pin Number	Description
A0 . . . A2	30, 32, 33	INPUTS This address is used with $\overline{CS}$, $\overline{IOR}$ or $\overline{IOW}$ to address all internal registers.
$\overline{CS}$	21	INPUT Chip Select enables a read or write of the internal register selected by A0 . . . A2. $\overline{CS}$ is a low active signal.
$\overline{DACK}$	26	INPUT DMA Acknowledge resets DRQ and selects the data register for input or output. $\overline{DACK}$ is a low active signal.
DRQ	22	OUTPUT DMA Request indicates that the data register is ready to be read or written. DRQ occurs only if DMA MODE is true in the command register. It is cleared by $\overline{DACK}$.
D0 . . . D7	34 . . . 40, 1	BI-DIRECTIONAL, TRI-STATE Microprocessor data bus Active high
$\overline{EOP}$	27	INPUT The End of Process signal is true during the last byte of a DMA transfer. This stops additional transfers but allows the current transfer to finish. $\overline{EOP}$ is a low active signal.
$\overline{IOR}$	24	INPUT I/O Read is used to read an internal register selected by $\overline{CS}$ and A0 . . . A2. It also selects the data register when used with $\overline{DACK}$. $\overline{IOR}$ is a low active signal.
$\overline{IOW}$	29	INPUT I/O Write is used to write an internal register selected by $\overline{CS}$ and A0 . . . A2. It also selects the data register when used with $\overline{DACK}$. $\overline{IOW}$ is a low active signal.
IRQ	23	OUTPUT Interrupt Request alerts the microprocessor of an error condition or an event completion.
READY	25	OUTPUT Ready can be used to control the speed of block mode DMA transfers
$\overline{RESET}$	28	INPUT Reset clears all registers. It does not force the SCSI signal $\overline{RST}$ to the active state. $\overline{RESET}$ is a low active signal

POWER SIGNALS

Pin Name	Pin Number	Description
V_{DD}	31	+5 VOLTS
GND	11	GROUND

SCSI INTERFACE SIGNALS

Pin Name	Pin Number	Description
$\overline{ACK}$	14	BI-DIRECTIONAL, OPEN COL INITIATOR ROLE: The chip asserts this signal in response to $\overline{REQ}$ for a byte transfer on the SCSI bus. TARGET ROLE: $\overline{ACK}$ is received as a response to the $\overline{REQ}$ signal. $\overline{ACK}$ is an active low signal.
$\overline{ATN}$	15	BI-DIRECTIONAL, OPEN COL. INITIATOR ROLE: The chip asserts this signal when the microprocessor requests the attention condition. TARGET ROLE: $\overline{ATN}$ is a received signal. $\overline{ATN}$ is an active low signal.
$\overline{BSY}$	13	BI-DIRECTIONAL, OPEN COL. The SCSI $\overline{BSY}$ signal can be driven and received concurrently. $\overline{BSY}$ is an active low signal
$\overline{C/D}$	18	BI-DIRECTIONAL, OPEN COL Command/Data is an input for an initiator, an output for a target. It indicates a command when asserted $\overline{C/D}$ is an active low signal
$\overline{I/O}$	17	BI-DIRECTIONAL, OPEN COL Input/Output is an input for an initiator, an output for a target. It indicates an input to the initiator when asserted. $\overline{I/O}$ is an active low signal
$\overline{MSG}$	19	BI-DIRECTIONAL, OPEN COL Message is an input for an initiator, an output for a target. It indicates a message when asserted $\overline{MSG}$ is an active low signal
$\overline{REQ}$	20	BI-DIRECTIONAL, OPEN COL The target asserts $\overline{REQ}$ to request a byte transfer from the initiator. The transfer may be in either direction. $\overline{REQ}$ is an active low signal
$\overline{RST}$	16	BI-DIRECTIONAL, OPEN COL. SCSI BUS reset signal $\overline{RST}$ is an active low signal
$\overline{SBO}$ $\overline{SB7}$, $\overline{SBP}$	2 . . 10	BI-DIRECTIONAL OPEN COL SCSI DATA BUS and PARITY These signals are low active
$\overline{SEL}$	12	BI-DIRECTIONAL, OPEN COL Select is used for selection and reselect operations $\overline{SEL}$ is an active low signal

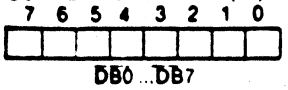


REGISTER SUMMARY

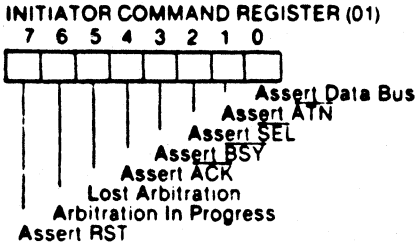
A2	A1	A0	R/W	REGISTER NAME
0	0	0	R	Current SCSI Data
0	0	0	W	Output Data Register
0	0	1	R/W	Initiator Command Reg.
0	1	0	R/W	Mode Register
0	1	1	R/W	Target Command Reg.
1	0	0	R	SCSI Bus Status
1	0	0	W	Select Enable Register
1	0	1	R	Bus & Status Register
1	0	1	W	Start DMA Send
1	1	0	R	Input Data Reg.
1	1	0	W	Start Target Rec. DMA
1	1	1	R	Reset Parity/Interrupts
1	1	1	W	Start Init. Rec. DMA

READ

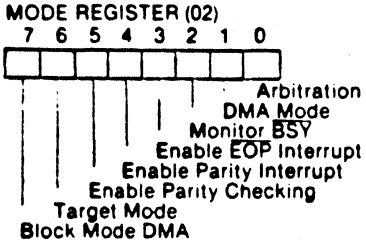
CURRENT SCSI DATA (00)



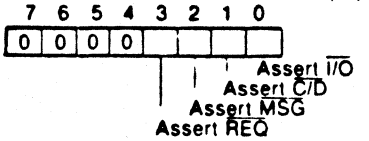
INITIATOR COMMAND REGISTER (01)



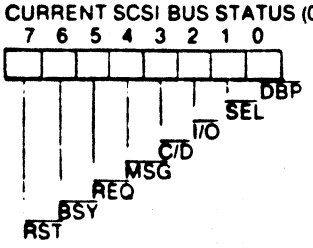
MODE REGISTER (02)



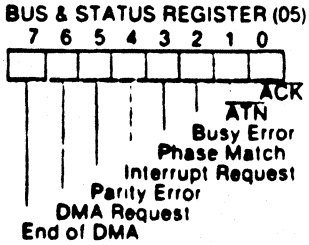
TARGET COMMAND REGISTER (03)



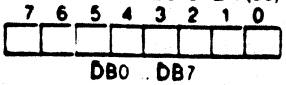
CURRENT SCSI BUS STATUS (04)



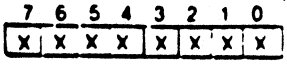
BUS & STATUS REGISTER (05)



INPUT DATA REGISTER (06)

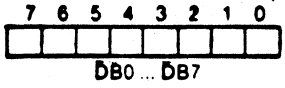


RESET PARITY/INTERRUPT (07)

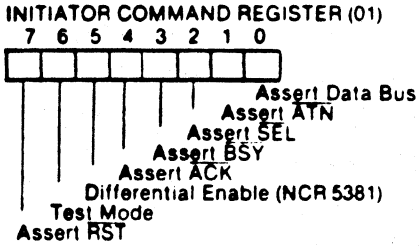


WRITE

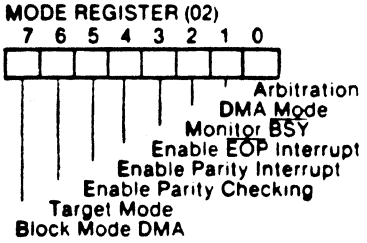
OUTPUT DATA REGISTER (00)



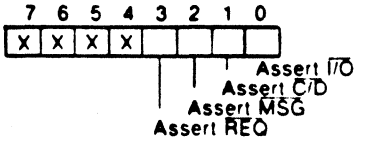
INITIATOR COMMAND REGISTER (01)



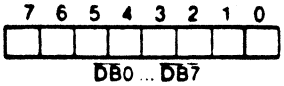
MODE REGISTER (02)



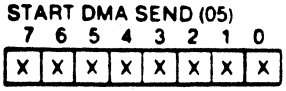
TARGET COMMAND REGISTER (03)



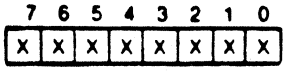
SELECT ENABLE REGISTER (04)



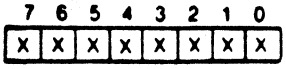
START DMA SEND (05)



START DMA TARGET RECEIVE (06)



START DMA INITIATOR RECEIVE (07)



NOTE X = DON'T CARE

ELECTRICAL CHARACTERISTICS
OPERATING CONDITIONS

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

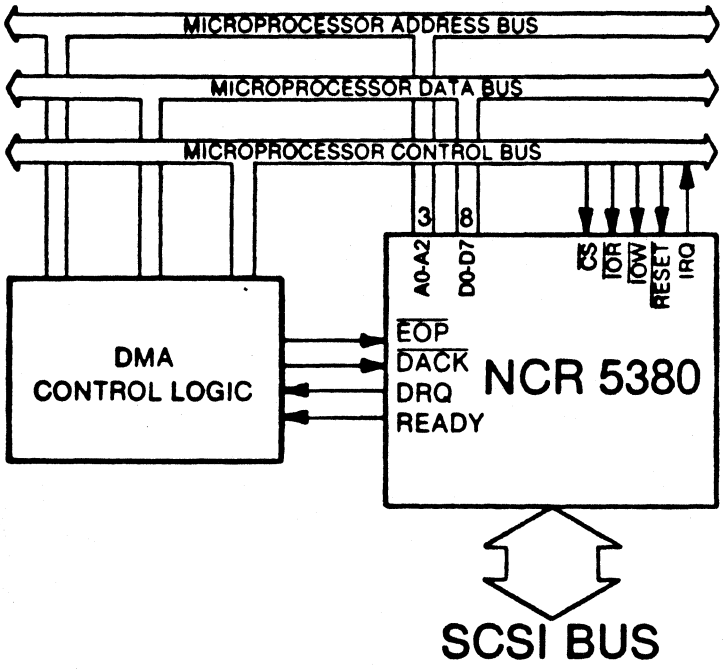
INPUT SIGNAL REQUIREMENTS

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

OUTPUT SIGNAL REQUIREMENTS

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

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



NCR MICROELECTRONICS DIVISION
1635 Aeroplaza Drive
Colorado Springs, Colorado 80916
Phone: 800/525-2252
Telex: 45 2457 NCR MICRO CSP

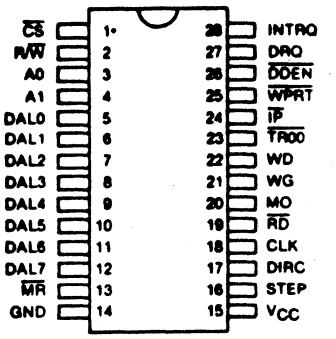
PRELIMINARY

WESTERN DIGITAL
CORPORATION

WD1770/1772 5¼" Floppy Disk Controller/Formatter

FEATURES

- 28 PIN DIP
- SINGLE 5V SUPPLY
- BUILT-IN DATA SEPARATOR
- BUILT-IN WRITE PRECOMPENSATION
- 5¼" SINGLE AND DOUBLE DENSITY
- MOTOR CONTROL
- 128, 256, 512 OR 1024 SECTOR LENGTHS
- TTL COMPATIBLE
- 8 BIT BIDIRECTIONAL DATA BUS
- TWO VERSIONS AVAILABLE
  WD1770 = STANDARD 179X STEP RATES
  WD1772 = FASTER STEP RATES



PIN DESIGNATION

DESCRIPTION

The WD1770 is a MOS/LSI device which performs the functions of a 5¼" Floppy Disk Controller/Formatter. It is similar to its predecessor, the WD179X, but also contains a digital data separator and write precompensation circuitry. The drive side of the interface needs no additional logic except for buffers/receivers. Designed for 5¼" single or double density operation, the device contains a programmable Motor On signal.

The WD1770 is implemented in NMOS silicon gate technology and is available in a 28 pin dual-in-line.

The WD1770 is a low cost version of the FD179X Floppy Disk Controller/Formatter. It is compatible with the 179X, but has a built-in digital data separator and write precompensation circuits. A single read line (RD, Pin 19) is the only input required to recover serial FM or MFM data from the disk drive. The device has been specifically designed for control of 5¼" floppy disk drives with data rates of 125 KBits/Sec (single density) and 250 KBits/Sec (double density). In addition, write precompensation of 125 Nsec from nominal can be enabled at any point through simple software commands. Another programmable feature, Motor On, has been incorporated to enable the spindle motor automatically prior to operating a selected drive.

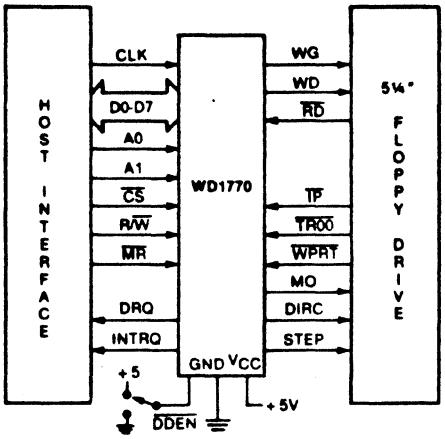
Two versions of the WD1770 are available. The standard version is compatible with the 179X stepping rates, while the WD1772 offers stepping rates of 2, 3, 5 and 6 msec.

The processor interface consists of an 8-bit bidirectional bus for transfer of status, data, and commands. All host communication with the drive occurs through these data lines. They are capable of driving one standard TTL load or three "LS" loads.

June, 1983

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	CHIP SELECT	$\overline{CS}$	A logic low on this input selects the chip and enable Host communication with the device.
2	READ/WRITE	R/$\overline{W}$	A logic high on this input controls the placement of data on the $\overline{D0}$-$\overline{D7}$ lines from a selected register, while a logic low causes a write operation to a selected register.
3,4	ADDRESS 0,1	A0, A1	These two inputs select a register to Read/Write data: $\overline{CS}$ A1 A0 R/$\overline{W}$ = 1 R/$\overline{W}$ = 0 0 0 0 Status Reg Command Reg 0 0 1 Track Reg Track Reg 0 1 0 Sector Reg Sector Reg 0 1 1 Data Reg Data Reg
5-12	DATA ACCESS LINES 0 THROUGH 7	DAL0-DAL7	Eight bit bidirectional bus used for transfer of data, control, or status. This bus is enabled by $\overline{CS}$ and R/$\overline{W}$. Each line will drive one TTL load.
13	MASTER RESET	$\overline{MR}$	A logic low pulse on this line resets the device and initializes the status register (internal pull-up).
14	GROUND	GND	Ground.
15	POWER SUPPLY	V_{CC}	+5V ± 5% power supply input.
16	STEP	STEP	The Step output contains a pulse for each step of the drive's R/$\overline{W}$ head. The WD1770 and WD1772 offer different step rates.
17	DIRECTION	DIRC	The Direction output is high when stepping in towards the center of the diskette, and low when stepping out.
18	CLOCK	CLK	This input requires a free-running 50% duty cycle clock (for internal timing) at 8 MHZ ± 1%.
19	READ DATA	$\overline{RD}$	This active low input is the raw data line containing both clock and data pulses from the drive.
20	MOTOR ON	MO	Active high output used to enable the spindle motor prior to read, write or stepping operations.
21	WRITE GATE	WG	This output is made valid prior to writing on the diskette.
22	WRITE DATA	WD	FM or MFM clock and data pulses are placed on this line to be written on the diskette.
23	TRACK 00	$\overline{TR00}$	This active low input informs the WD1770 that the drive's R/$\overline{W}$ heads are positioned over Track zero (internal pull-up).
24	INDEX PULSE	$\overline{IP}$	This active low input informs the WD1770 when the physical index hole has been encountered on the diskette (internal pull-up).
25	WRITE PROTECT	$\overline{WPRT}$	This input is sampled whenever a Write Command is received. A logic low on this line will prevent any Write Command from executing (internal pull-up).
26	DOUBLE DENSITY ENABLE	$\overline{DDEN}$	This input pin selects either single (FM) or double (MFM) density When $\overline{DDEN}$ = 0, double density is selected (internal pull-up).

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
27	DATA REQUEST	DRQ	This active high output indicates that the Data Register is full (on a Read) or empty (on a Write operation).
28	INTERRUPT REQUEST	INTRQ	This active high output is set at the completion of any command or reset a read of the Status Register.



WD1770 SYSTEM BLOCK DIAGRAM

ARCHITECTURE

The Floppy Disk Formatter block diagram is illustrated on page 4. The primary sections include the parallel processor interface and the Floppy Disk Interface.

Data Shift Register — This 8-bit register assembles serial data from the Read Data input ($\overline{RD}$) during Read operations and transfers serial data to the Write Data output during Write operations.

Data Register — This 8-bit register is used as a holding register during Disk Read and Write operations. In Disk Read operations, the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations, information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command, the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

Track Register — This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write, and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.

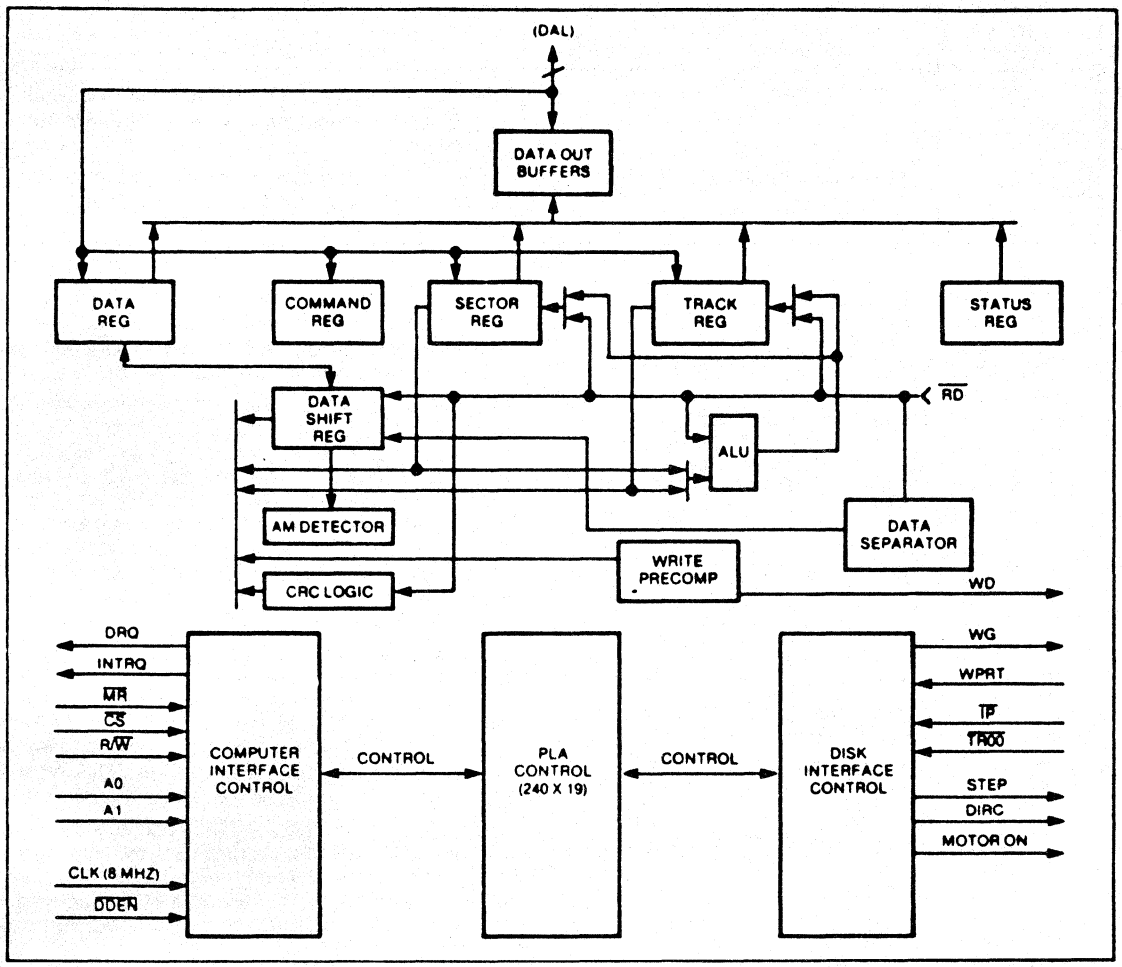
Sector Register (SR) — This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

Command Register (CR) — This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

Status Register (STR) — This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

CRC Logic — This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is:
$G(x) = x^{16} + x^{12} + x^5 + 1$.
The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic/Logic Unit (ALU) — The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.



WD1770 BLOCK DIAGRAM

Timing and Control — All computer and Floppy Disk interface controls are generated through this logic. The internal device timing is generated from an external crystal clock. The FD1770 has two different modes of operation according to the state of $\overline{DDEN}$. When $\overline{DDEN}$ = 0, double density (MFM) is enabled. When $\overline{DDEN}$ = 1, single density is enabled.

AM Detector — The address mark detector detects ID, data and index address marks during read and write operations.

Data Separator — A digital data separator consisting of a ring shift register and data window detection logic provides read data and a recovery clock to the AM detector.

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines (DAL) and associated control signals. The DAL are used to transfer Data, Status, and Control words out of, or into the WD1770. The DAL are three state buffers that are enabled as output drivers when Chip Select ($\overline{CS}$) and $R/\overline{W}$ = 1 are active or act as input receivers when $\overline{CS}$ and $R/\overline{W}$ = 0 are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signal $R/\overline{W}$ during a Read operation or Write operation are interpreted as selecting the following registers:

A1	A0	READ (R/$\overline{W}$ = 1)	WRITE (R/$\overline{W}$ = 0)
0	0	Status Register	Command Register
0	1	Track Register	Track Register
1	0	Sector Register	Sector Register
1	1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the WD1770 and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operations continues until the end of sector is reached.

On Disk Write operations the Data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

The WD1770 has two modes of operation according to the state $\overline{DDEN}$ (Pin 26). When $\overline{DDEN}$ = 1, single density is selected. In either case, the CLK input (Pin 18) is at 8 MHZ.

GENERAL DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{DDEN}$ should be placed to logical "1." For MFM formats, $\overline{DDEN}$ should be placed to a logical "0." Sector lengths are determined at format time by the fourth byte in the "ID" field.

SECTOR LENGTH TABLE	
SECTOR LENGTH FIELD (HEX)	NUMBER OF BYTES IN SECTOR (DECIMAL)
00	128
01	256
02	512
03	1024

The number of sectors per tract as far as the WD1770 is concerned can be from 1 to 255 sectors. The number of tracks as far as the WD1770 is concerned is from 0 to 255 tracks.

GENERAL DISK WRITE OPERATION

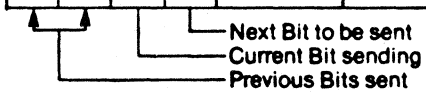
When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the device before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{Write\ Protect}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set.

For Write operations, the WD1770 provides Write Gate (Pin 21) to enable a Write condition, and Write Data (Pin 22) which consists of a series of active high pulses. These pulses contain both Clock and Data information in FM and MFM. Write Data provides the unique missing clock patterns for recording Address Marks.

The Precomp Enable bit in Write commands allow automatic Write precompensation to take place. The outgoing Write Data stream is delayed or advanced from nominal by 125 nanoseconds according to the following table:

PATTERN				MFM	FM
X	1	1	0	Early	N/A
X	0	1	1	Late	N/A
0	0	0	1	Early	N/A
1	0	0	0	Late	N/A



Precompensation is typically enabled on the innermost tracks where bit shifts usually occur and bit density is at its maximun.

COMMAND DESCRIPTION

The WD1770 will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force Interrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 1.

COMMAND SUMMARY

TYPE	COMMAND	BITS 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	u	h	V	r_1	r_0
I	Step-in	0	1	0	u	h	V	r_1	r_0
I	Step-out	0	1	1	u	h	V	r_1	r_0
II	Read Sector	1	0	0	m	h	E	0	0
II	Write Sector	1	0	1	m	h	E	P	a_0
III	Read Address	1	1	0	0	h	E	0	0
III	Read Track	1	1	1	0	h	E	0	0
III	Write Track	1	1	1	1	h	E	P	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

FLAG SUMMARY

TYPE I COMMANDS

h = Motor On Flag (Bit 3)

h = 0, Enable Spin-Up Sequence
h = 1, Disable Spin-Up Sequence

V = Verify Flag (Bit 2)

V = 0, No Verify
V = 1, Verify on Destination Track

r_1, r_0 = Stepping Rate (Bits 1, 0)

r_1	r_0	WD1770	WD1772
0	0	6 ms	2 ms
0	1	12 ms	3 ms
1	0	20 ms	5 ms
1	1	30 ms	6 ms

u = Update Flag (Bit 4)

u = 0, No Update
u = 1, Update Track Register

TYPE II & III COMMANDS

m = Multiple Sector Flag (Bit 4)

m = 0, Single Sector
m = 1, Multiple Sector

a_0 = Data Address Mark (Bit 0)

a_0 = 0, Write Normal Data Mark
a_0 = 1, Write Deleted Data Mark

E = 30ms Settling Delay (Bit 2)

E = 0, No Delay
E = 1, Add 30ms Delay

P = Write Precompensation (Bit 1)

P = 0, Enable Write Precomp
P = 1, Disable Write Precomp

TYPE IV COMMANDS

I_3-I_0 Interrupt Condition (Bits 3-0)

I_0 = 1, Don't Care
I_1 = 1, Don't Care
I_2 = 1, Interrupt on Index Pulse
I_3 = 1, Immediate Interrupt
I_3-I_0 = 0, Terminate without Interrupt

TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field (r_0,r_1), which determines the stepping motor rate.

A 4μs (MFM) or 8 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output. The chip will step the drive in the same direction it last stepped unless the command changes the direction.

The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 24μs before the first stepping pulse is generated.

After the last directional step an additional 30 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. There is also a 30 ms head settling time if the E flag is set in any Type II or III command.

When a Seek, Step or Restore command is executed, an optional verification of Read/Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 30 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. If there is a match but not a valid CRC, the CRC error status bit is set (Status Bit 3), and the next encountered ID field is read from the disk for the verification operation.

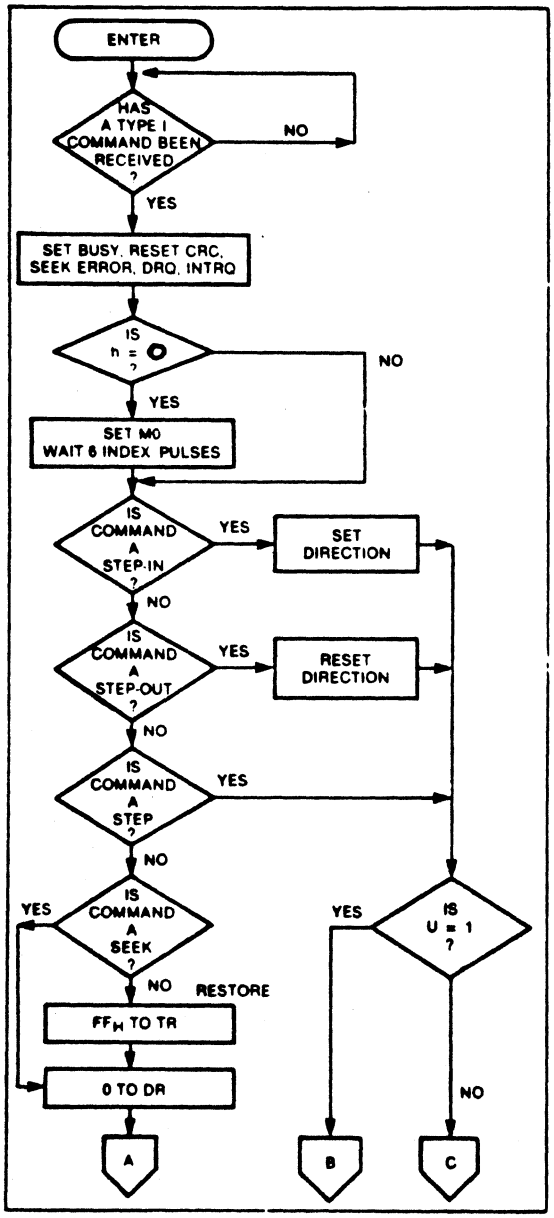
The WD1770 must find an ID field with correct track number and correct CRC within 5 revolutions of the media, otherwise the seek error is set and an INTRQ is generated. If V = 0, no verification is performed.

All commands, except the Force Interrupt command, may be programmed via the h Flag to delay for spindle motor start up time. If the h Flag is set and the Motor On line (Pin 20) is low when a command is received, the WD1770 will force Motor On to a logic 1 and wait 6 revolutions before executing the command. At 300 RPM, this guarantees a one second spindle start up time. If after finishing the command, the device remains idle for 10 revolutions, the Motor

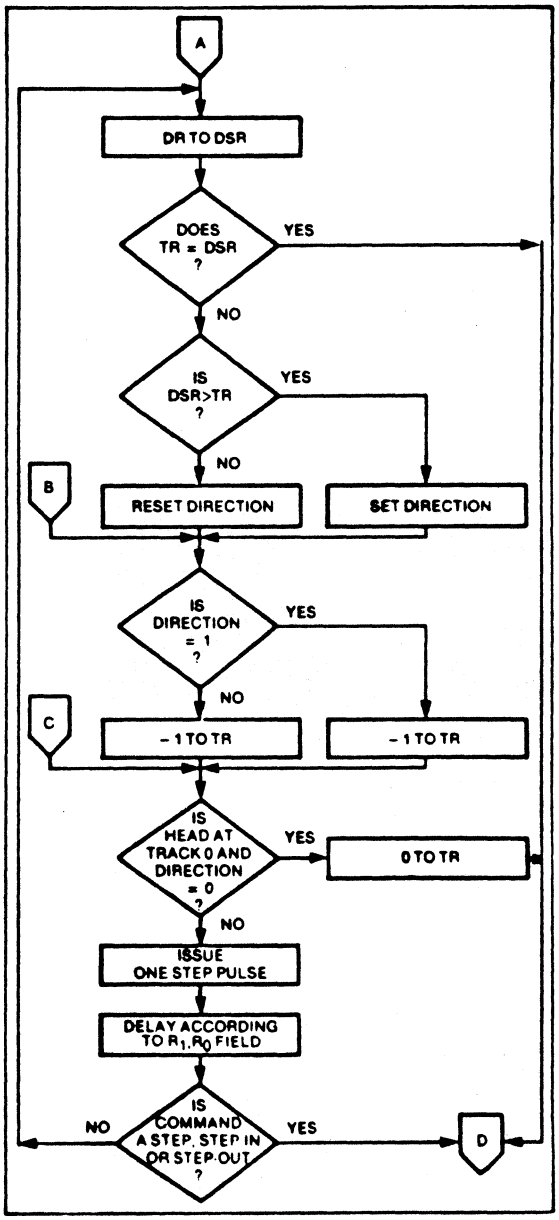
On line will go back to a logic 0. If a command is issued while Motor On is high, the command will execute immediately, defeating the 6 revolution start up. This feature allows consecutive Read or Write commands without waiting for motor start up each time; the WD1770 assumes the spindle motor is up to speed.

RESTORE (SEEK TRACK 0)

Upon receipt of this command, the Track 00 ($\overline{TR00}$) input is sampled. If $\overline{TR00}$ is active low indicating the Read/Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses (Pin 16) at a rate specified by the r_1,r_0 field are issued until the $\overline{TR00}$ input is activated.



TYPE I COMMAND FLOW

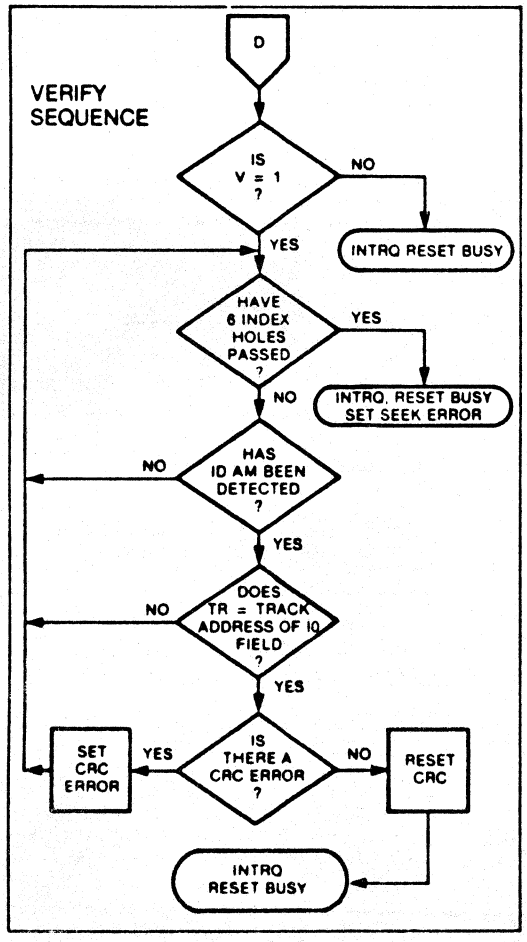


TYPE I COMMAND FLOW

At this time, the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TR00}$ input does not go active low after 255 stepping pulses, the WD1770 terminates operation, interrupts, and sets the Seek error status bit, providing the V flag is set. A verification operation also takes place if the V flag is set. The h bit allows the Motor On option at the start of command.

SEEK

This command assumes that the Track Register contains the track number of the current position of the Read/Write head and the Data Register contains the desired track number. The WD1770 will update the Track Register and issue stepping pulses in the appropriate direction until the contents of the Track Register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command. Note: When using multiple drives, the track register must be updated for the drive selected before seeks are issued.



TYPE I COMMAND FLOW

STEP

Upon receipt of this command, the WD1770 issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the r_1,r_0 field, a verification takes place if the V flag is on. If the U flag is on, the Track Register is updated. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the WD1770 issues one stepping pulse in the direction towards track 76. If the U flag is on, the Track Register is incremented by one. After a delay determined by the r_1,r_0 field, a verification takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

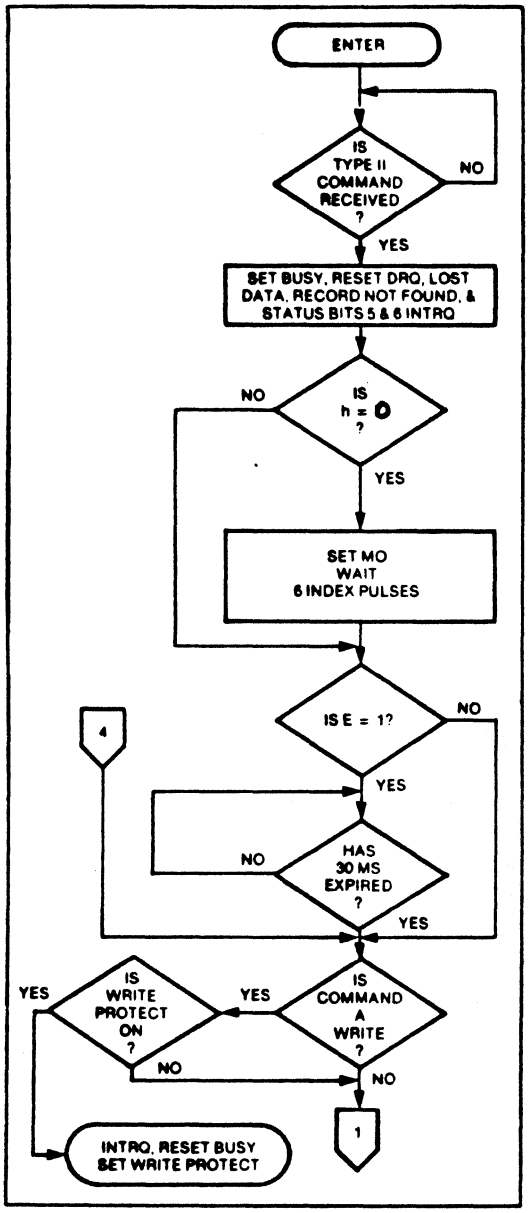
STEP-OUT

Upon receipt of this command, the WD1770 issues one stepping pulse in the direction towards track 0. If the U flag is on, the Track Register is decremented by one. After delay determined by the r_1,r_0 field, a verification takes place if the V flag is on. The h bit allows the Motor On option at the start of the command. An interrupt is generated at the completion of the command.

TYPE II COMMANDS

The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status bit is set. If the E flag = 1 the command will execute after a 30 msec delay.

When an ID field is located on the disk, the WD1770 compares the Track Number on the ID field with the Track Register. If there is not a match, the next encountered ID field is read and a comparison is again made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The WD1770 must find an ID field with a Track number, Sector number, and CRC within four revolutions of the disk, other-



TYPE II COMMAND

wise, the Record not found status bit is set (Status Bit 4) and the command is terminated with an interrupt (INTRQ).

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The WD1770 will continue to read or write multiple records and update the sector register in numerical ascending sequence until the sector register exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

For example: If the WD1770 is instructed to read sector 27 and there are only 26 on the track, the sector register exceeds the number available. The WD1770 will search for 5 disk revolutions, interrupt out, reset busy, and set the record not found status bit.

READ SECTOR

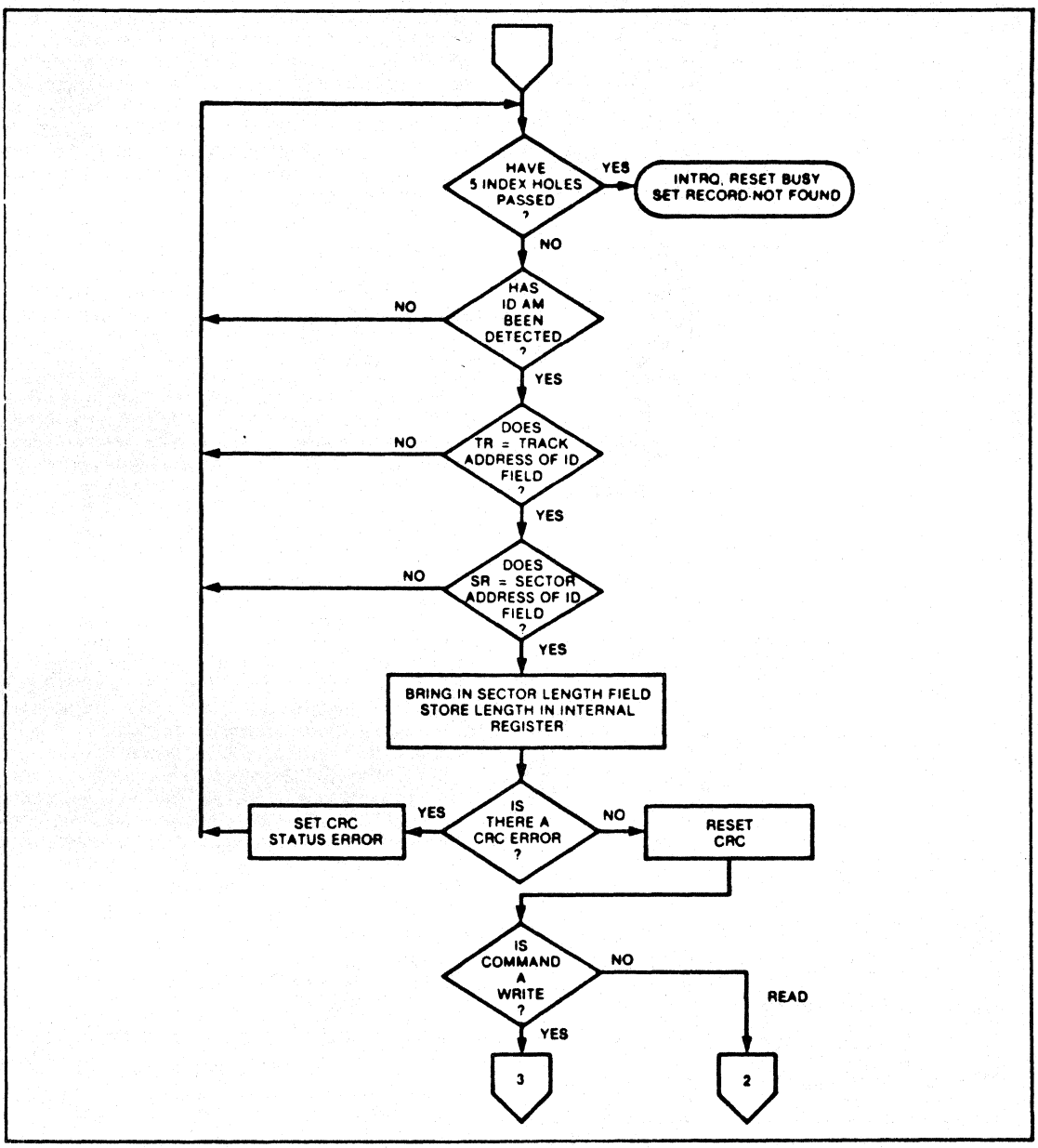
Upon receipt of the Read Sector command, the Busy status bit is set, and when a ID field is encountered that has the correct track number, correct sector number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the ID field is searched for and verified again followed by the Data Address Mark search. If after 5 revolutions the DAM cannot be found, the Record Not Found status bit is set and the operation is terminated. When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the computer has not read the previous contents of the DR before a new character is transferred that character is lost and the Lost Data Status bit is set. This sequence continues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown:

STATUS BIT 5	
1	Deleted Data Mark
0	Data Mark

WRITE SECTOR

Upon receipt of the Write Sector command, the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, and correct CRC, a DRQ is generated. The WD1770 counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer) If DRQ has not been serviced, the command is terminated

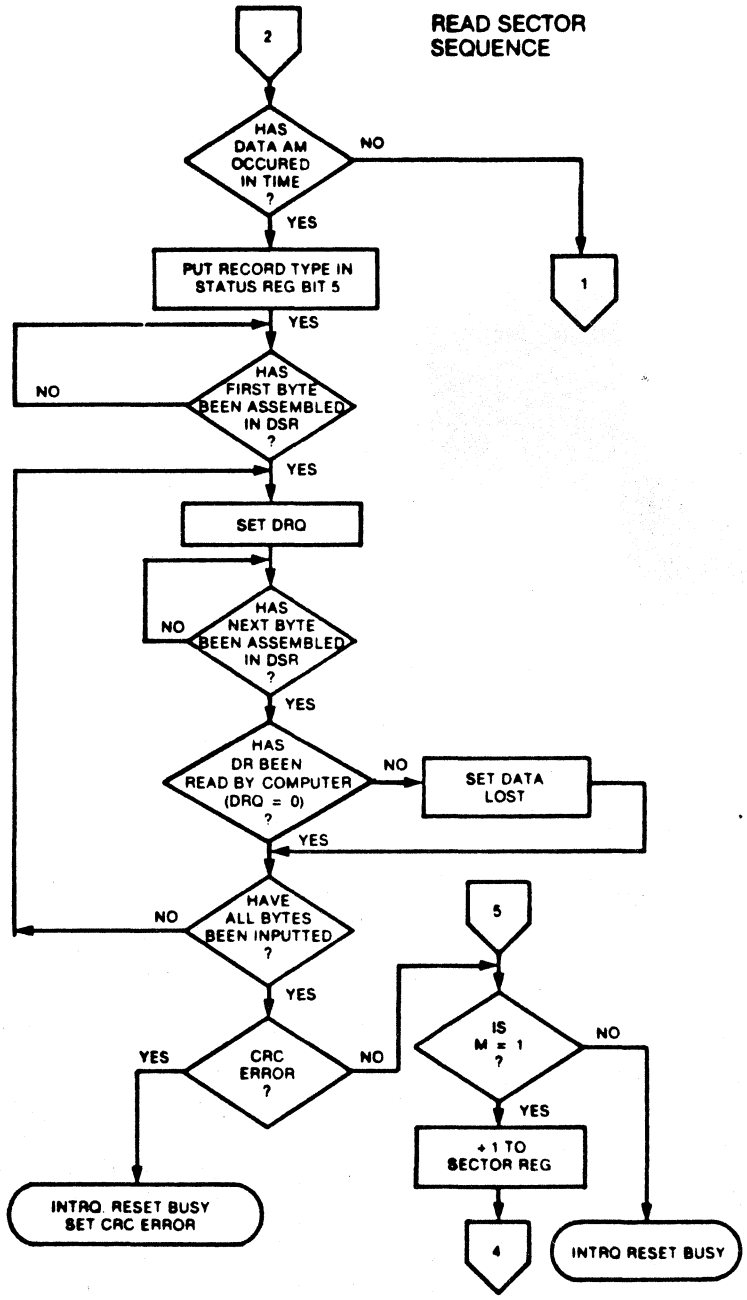


TYPE II COMMAND

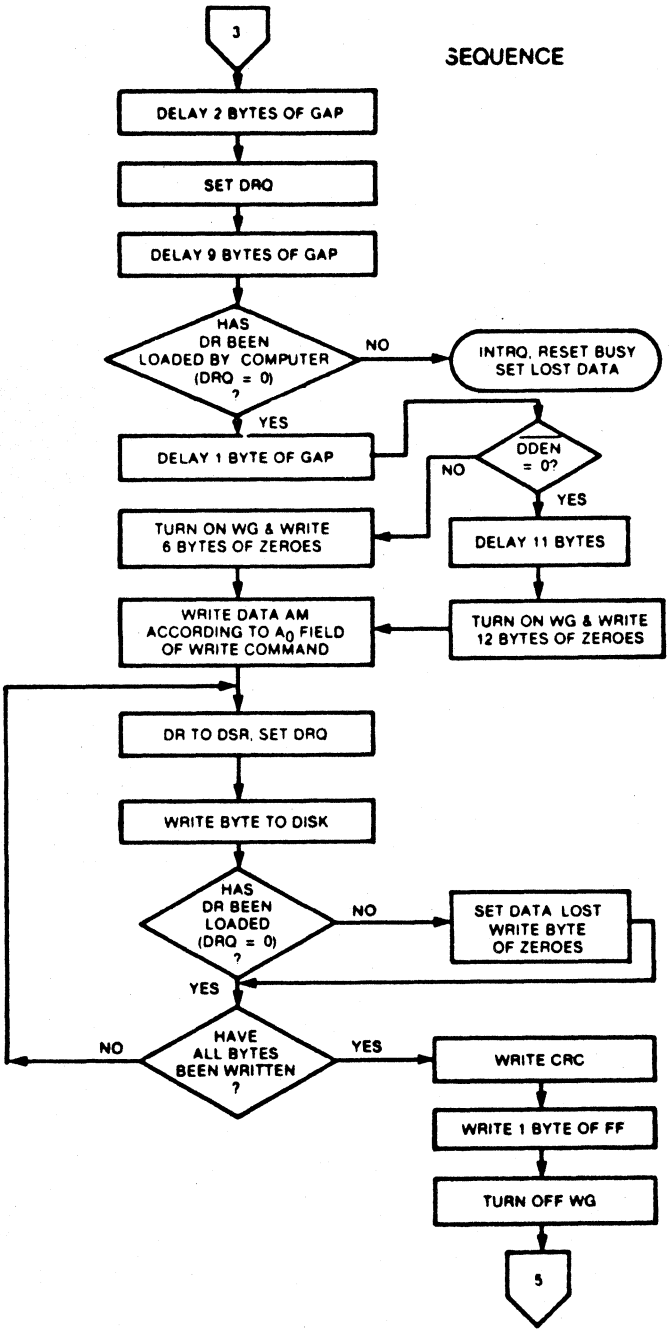
and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeroes in single density and 12 bytes in double density are then written on the disk. At this time, the Data Address Mark is then written on the disk as determined by the a_0 field of the command as shown below:

a_0	DATA ADDRESS MARK (BIT 0)
1	Deleted Data Mark
0	Data Mark

The WD1770 then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit



TYPE II COMMAND



TYPE II COMMAND

is set and a byte of zeroes is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated. INTRQ will set 24μsec (MFM) after the last CRC byte is written. For partial sector writing, the proper method is to write data and fill the balance with zeroes.

TYPE III COMMANDS

Read Address

Upon receipt of the Read Address command, the Busy Status Bit is set. The next encountered ID field is then read in from the disk, and six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

TRACK ADDR	SIDE NUMBER	SECTOR ADDRESS	SECTOR LENGTH	CRC 1	CRC 2
1	2	3	4	5	6

Although the CRC characters are transferred to the computer, the WD1770 checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register so that a comparison can be made by the user. At the end of the operation an interrupt is generated and the Busy Status is reset.

Read Track

Upon receipt of the READ track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. All Gap, Header, and data bytes are assembled and transferred to the data register and DRQ's are generated for each byte. The accumulation of bytes is synchronized to each address mark encountered. An interrupt is generated at the completion of the command.

This command has several characteristics which make it suitable for diagnostic purposes. They are: no CRC checking is performed; gap information is included in the data stream; and the address mark detector is on for the duration of the command. Because the AM detector is always on, write splices or noise may cause the chip to look for an AM.

The ID AM, ID field, ID CRC bytes, DAM, Data, and Data CRC Bytes for each sector will be correct. The Gap Bytes may be read incorrectly during write-splice time because of synchronization.

WRITE TRACK FORMATTING THE DISK

(Refer to section on Type III commands for flow diagrams.)

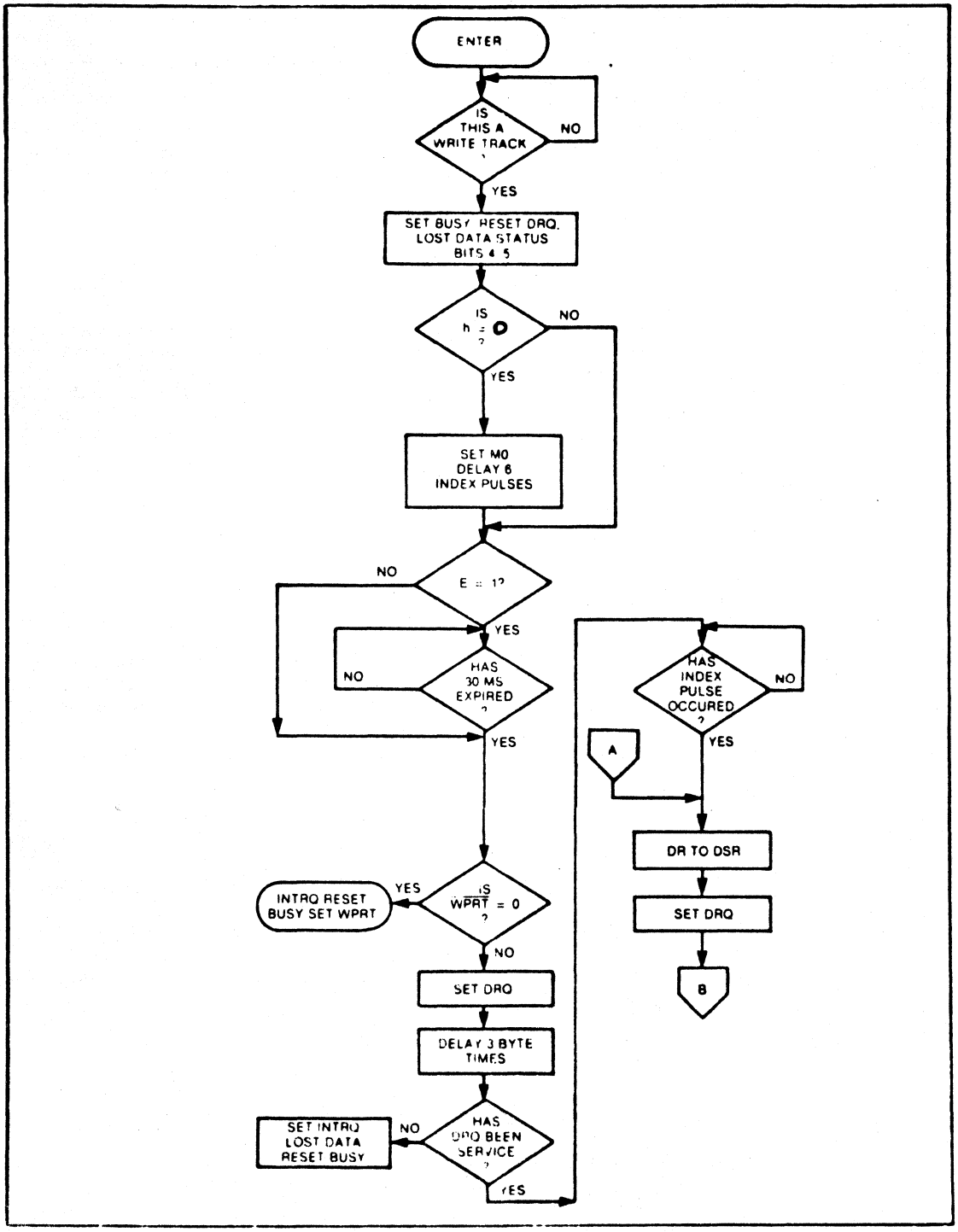
Formatting the disk is a relatively simple task when operating programmed I/O or when operating under DMA with a large amount of memory. Data and gap information must be provided at the computer interface. Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command.

Upon receipt of the Write Track command, the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded within 3 byte times, the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the interrupt is activated. If a byte is not present in the DR when needed, a byte of zeroes is substituted.

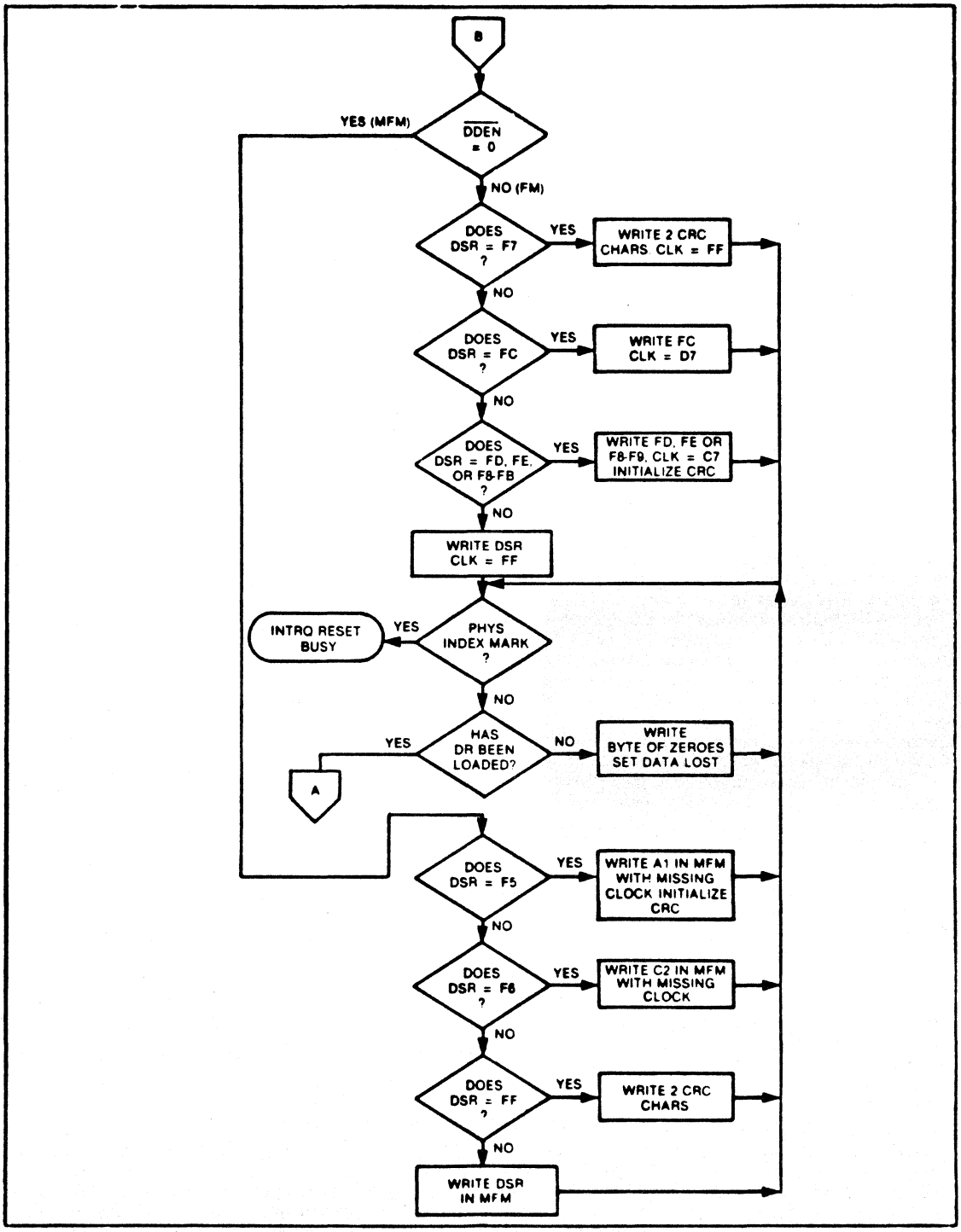
This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the WD1770 detects a data pattern of F5 through FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation.

DATA PATTERN IN DR (HEX)	IN FM ($\overline{DDEN}$ = 1)	IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Present CRC
F6	Not Allowed	Write C2** in MFM
F7	Generate 2 CRC bytes	Generate 2 CRC bytes
F8 thru FB	Write F8 thru FB, CLK = C7, Preset CRC	Write F8 thru FB, in MFM
FC	Write FC with CLK = D7	Write FC in MFM
FD	Write FD with CLK = FF	Write FD in MFM
FE	Write FE, CLK = C7, Preset CRC	Write FE in MFM
FF	Write FF with CLK = FF	Write FF in MFM

*Missing clock transition between bits 4 and 5.
**Missing clock transition between bits 3 and 4.



TYPE III COMMAND WRITE TRACK



TYPE III COMMAND WRITE TRACK

The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 through FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

TYPE IV COMMANDS

The Forced Interrupt command is generally used to terminate a multiple sector read or write command or to insure Type I status in the status register. This command can be loaded into the command register at any time. If there is a current command under execution (busy status bit set) the command will be terminated and the busy status bit reset.

The lower four bits of the command determine the conditional interrupt as follows:

I_0 = Don't Care
I_1 = Don't Care
I_2 = Every Index Pulse
I_3 = Immediate Interrupt

The conditional interrupt is enabled when the corresponding bit positions of the command (I_3-I_0) are set to a 1. Then, when the condition for interrupt is met, the INTRQ line will go high signifying that the condition specified has occurred. If I_3-I_0 are all set to zero (HEX D0), no interrupt will occur but any command presently under execution will be immediately terminated. When using the immediate interrupt condition (I_3 = 1) an interrupt will be immediately generated and the current command terminated. Reading the status or writing to the command register will not automatically clear the interrupt. The HEX D0 is the only command that will enable the immediate interrupt (HEX D8) to clear on a subsequent load command register or read status register operation. Follow a HEX D8 with D0 command.

Wait 16 micro sec (double density) or 32 micro sec (single density) before issuing a new command after issuing a forced interrupt. Loading a new command sooner than this will nullify the forced interrupt.

Forced interrupt stops any command at the end of an internal micro-instruction and generates INTRQ when the specified condition is met. Forced interrupt will wait until ALU operations in progress are complete (CRC calculations, compares, etc.).

Status Register

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The user has the option of reading the status register through program control or using the DRQ line with DMA or interrupt methods. When the Data register is read the DRQ bit in the status register and the DRQ line are automatically reset. A write to the Data register also causes both DRQ's to reset.

The busy bit in the status may be monitored with a user program to determine when a command is complete, in lieu of using the INTRQ line. When using the INTRQ, a busy status check is not recommended because a read of the status register to determine the condition of busy will reset the INTRQ line.

The format of the Status Register is shown below:

(BITS)							
7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

RECOMMENDED — 128 BYTES/SECTOR

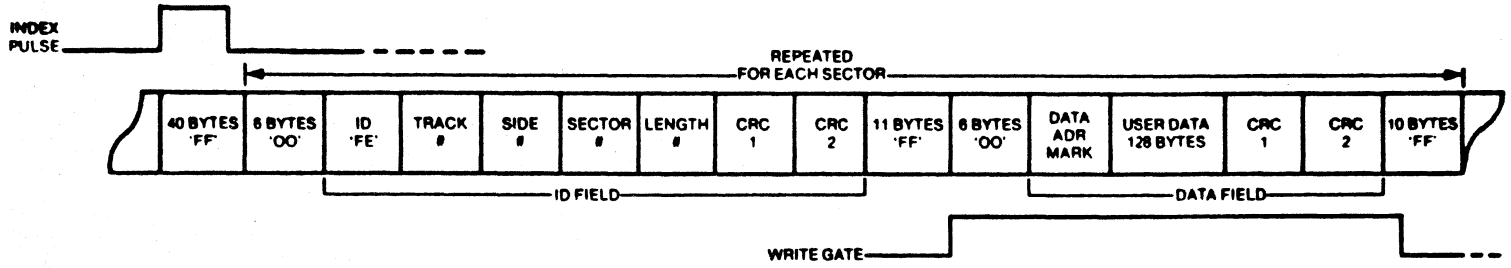
Shown below is the recommended single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one Data Request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00 (Sector Length)
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
10	FF (or 00)
369**	FF (or 00)

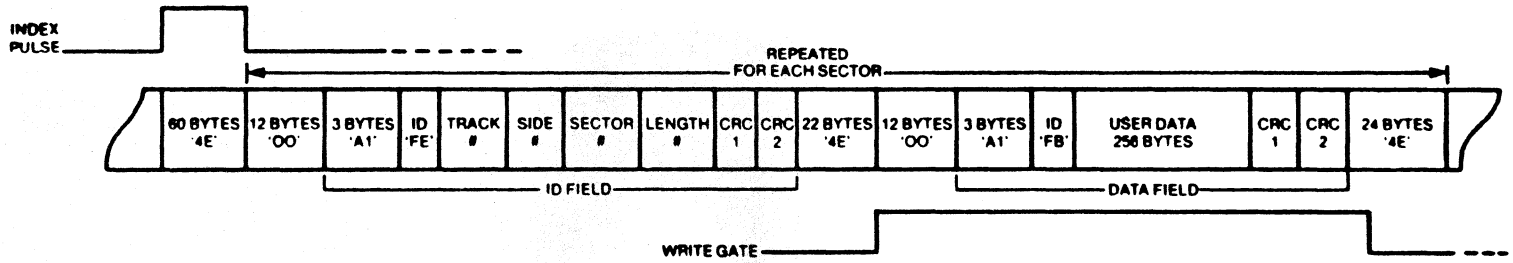
*Write bracketed field 16 times.
**Continue writing until WD1770 interrupts out. Approx. 369 bytes.

256 BYTES/SECTOR

Shown below is the recommended dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.



SINGLE DENSITY FORMAT



DOUBLE DENSITY FORMAT

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
60	4E
12	00
3	F5 (Writes A1)
1	FE (ID Address Mark)
1	Track Number (0 thru 4C)
1	Side Number (0 or 1)
1	Sector Number (1 thru 1A)
1	01 (Sector Length)
1	F7 (2 CRC's written)
22	4E
12	00
3	F5 (Writes A1)
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRC's written)
24	4E
668**	4E

*Write bracketed field 16 times.
**Continue writing until WD1770 interrupts out. Approx. 668 bytes.

1. Non-Standard Formats

Variations in the recommended formats are possible to a limited extent if the following requirements are met:

1) Sector size must be 128, 256, 512 of 1024 bytes.
2) Gap 2 cannot be varied from the recommended format.
3) 3 bytes of A1 must be used in MFM.

In addition, the Index Address Mark is not required for operation by the WD1770 Gap 1, 3, and 4 lengths can be as short as 2 bytes for WD1770 operation, however PLL lock up time, motor speed variation, write-splice area, etc. will add more bytes to each gap to achieve proper operation. It is recommended that the recommended format be used for highest system reliability.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
•	6 bytes 00	12 bytes 00
•		3 bytes A1
Gap III**	10 bytes FF •	24 bytes 4E
	4 bytes 00	8 bytes 00
		3 bytes A1
Gap IV	16 bytes FF	16 bytes 4E

*Byte counts must be exact.
**Byte counts are minimum, except exactly 3 bytes of A1 must be written.

STATUS REGISTER DESCRIPTION

BIT NAME	MEANING
S7 MOTOR ON	This bit reflects the status of the Motor On output.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/SPIN-UP	When set, this bit indicates that the Motor Spin-Up sequence has completed (6 revolutions) on Type I commands. Type 2 & 3 commands, this bit indicates record Type. 0 = Data Mark. 1 = Deleted Data Mark.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA/ TRACK 00	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when update. On Type I commands, this bit reflects the status of the TRACK 00 Pin.
S1 DATA REQUEST/ INDEX	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated. On Type I commands, this bit indicates the status of the Index Pin.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.

DC ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Storage Temperature ..... −55°C to +125°C
Operating Temperature ..... 0°C to 70°C Ambient

Maximum Voltage to Any Input with Respect to V_{SS} ..... (−15 to −0.3V)

DC OPERATING CHARACTERISTICS

TA = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ±.25V

SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNITS	CONDITIONS
I_{IL}	Input Leakage		10	μA	$V_{IN} = V_{CC}$
I_{OL}	Output Leakage		10	μA	$V_{OUT} = V_{CC}$
V_{IH}	Input High Voltage	2.0		V	
V_{IL}	Input Low Voltage		0.8	V	
V_{OH}	Output High Voltage	2.4		V	$I_O = -100\,\mu A$
V_{OL}	Output Low Voltage		0.40	V	$I_O = 1.6$ mA
P_D	Power Dissipation		.75	W	
R_{PU}	Internal Pull-Up	100	1700	μA	$V_{IN} = 0V$
I_{CC}	Supply Current	75 (Typ)	150	mA	

AC TIMING CHARACTERISTICS

TA = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ±.25V

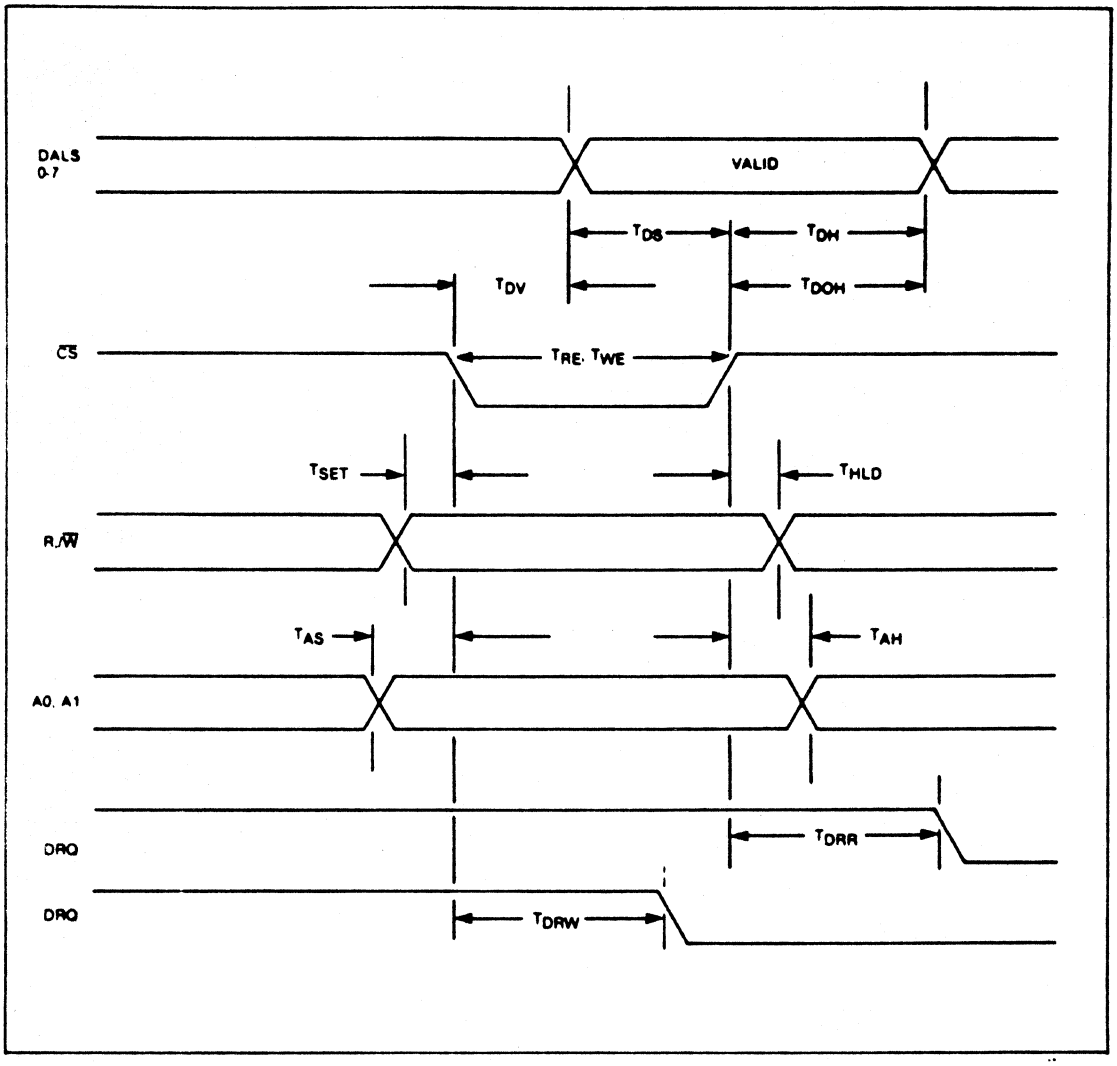
READ ENABLE TIMING — $\overline{RE}$ such that: $R/\overline{W} = 1, \overline{CS} = 0$.

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TRE	RE Pulse Width of $\overline{CS}$	150			nsec	C_L = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		25	100	nsec	
TIRR	INTRQ Reset from $\overline{RE}$			8000	nsec	
TDV	Data Valid from $\overline{RE}$		100	200	nsec	C_L = 50 pf
TDOH	Data Hold from $\overline{RE}$	50		150	nsec	C_L = 50 pf

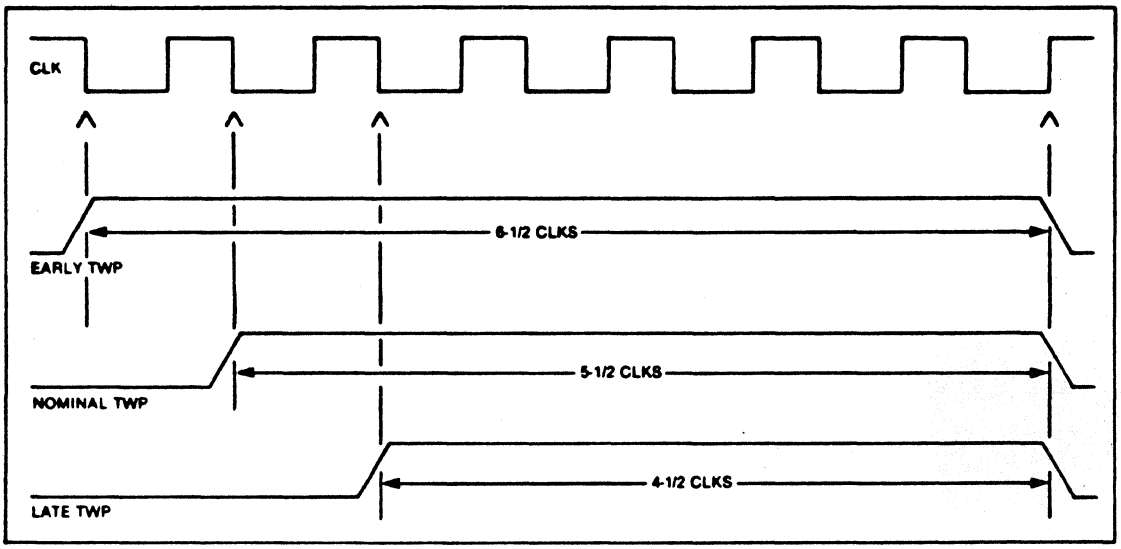
Note: DRQ and INTRQ reset are from rising edge (lagging) of $\overline{RE}$, whereas resets are from falling edge (leading) of $\overline{WE}$.

WRITE ENABLE TIMING — $\overline{WE}$ such that: $R/\overline{W} = 0, \overline{CS} = 0$.

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TAS	Setup ADDR to $\overline{CS}$	50			nsec	
TSET	Setup $R/\overline{W}$ to $\overline{CS}$	0			nsec	
TAH	Hold ADDR from $\overline{CS}$	20			nsec	
THLD	Hold $R/\overline{W}$ from $\overline{CS}$	0			nsec	
TWE	$\overline{WE}$ Pulse Width	150			nsec	
TDRW	DRQ Reset from $\overline{WE}$		100	200	nsec	
TIRW	INTRQ Reset from $\overline{WE}$			8000	nsec	
TDS	Data Setup to $\overline{WE}$	150			nsec	
TDH	Data Hold from $\overline{WE}$	0			nsec	



REGISTER TIMINGS



WRITE DATA TIMING

WRITE DATA TIMING:

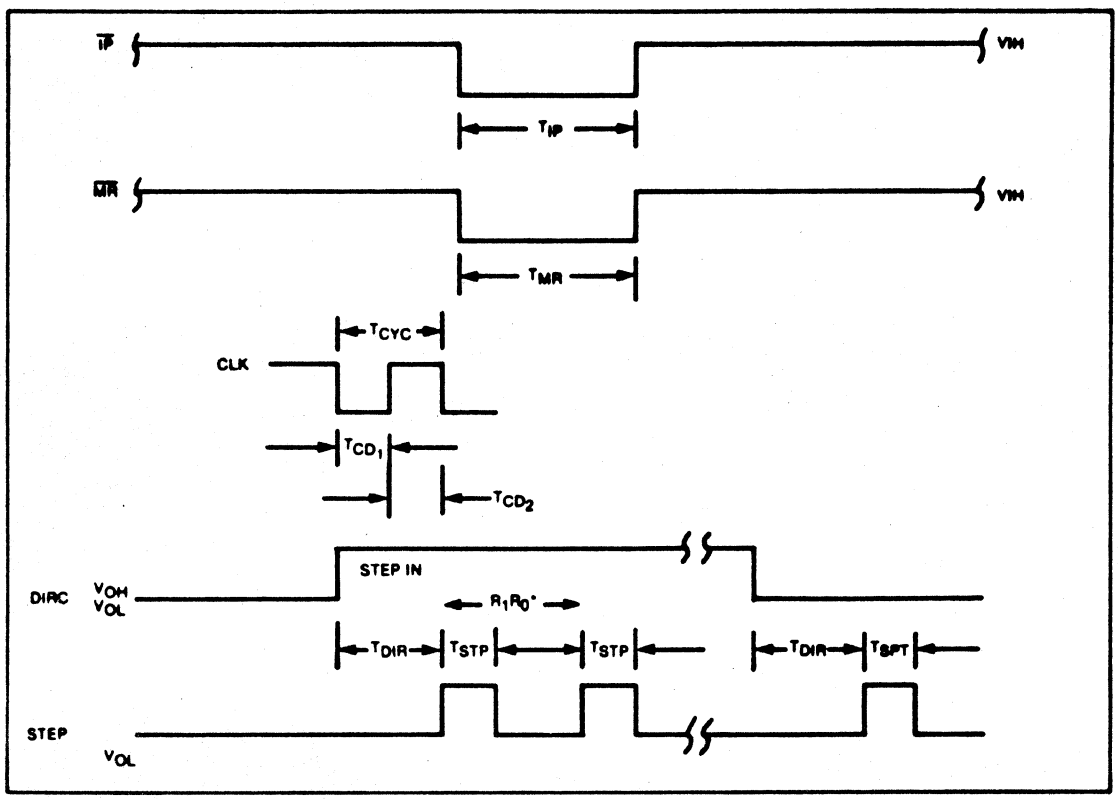
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Twg	Write Gate to Write Data		4		µsec	FM
			2		µsec	MFM
Tbc	Write Data Cycle Time		4,6,8		µsec	
Twf	Write Gate off from WD		4		µsec	FM
			2		µsec	MFM
Twp	Write Data Pulse Width		820		nsec	Early MFM
			690		nsec	Nominal MFM
			570		nsec	Late MFM
			1380		nsec	FM

INPUT DATA TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TPW	$\overline{\text{Raw Read}}$ Pulse Width	200			nsec	
TBC	$\overline{\text{Raw Read}}$ Cycle Time	3000			nsec	

MISCELLANEOUS TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TCD_1	Clock Duty (low)	50	67		nsec	
TCD_2	Clock Duty (high)	50	67		nsec	
TSTP	Step Pulse Output		4		µsec	MFM
			8			FM
TDIR	Dir Setup to Step		24		µsec	MFM
			48			FM
TMR	Master Reset Pulse Width	50			µsec	
TIP	Index Pulse Width	20			µsec	



MISCELLANEOUS TIMING