5070 CPU card

Reference manual

Manual part #5866, rev. 1106

CONTACT INFORMATION
Front Desk: 303–430–1500
Technical Support: 303–426–4521
FastHelp@octagonsystems.com
www.octagonsystems.com

Copyright

Disclaimer

Technical Support

Carefully recheck your system before calling Technical Support. Run as many tests as possible; the more information you can provide, the easier it will be for Technical Support staff to help you solve the problem. For additional technical assistance, try the following:

Technical Support telephone: 303–426–4521
E-mail Technical Support: fasthelp@octagonsystems.com
Applications Notes (via web): www.octagonsystems.com

IMPORTANT!

Please read the following section before installing your product:

Octagon's products are designed to be high in performance while consuming very little power. In order to maintain this advantage, CMOS circuitry is used.

CMOS chips have specific needs and some special requirements that the user must be aware of. Read the following to help avoid damage to your card from the use of CMOS chips.

Using CMOS circuitry in industrial control

Industrial computers originally used LSTTL circuits. Because many PC components are used in laptop computers, IC manufacturers are exclusively using CMOS technology. Both TTL and CMOS have failure mechanisms, but they are different. Described below are some of the failures that are common to all manufacturers of CMOS equipment. However, much of the information has been put in the context of the Micro PC.

Octagon has developed a reliable database of customer–induced, field failures. The average MTBF of Micro PC cards exceeds 11 years, yet there are failures. Most failures have been identified as customer–induced, but there is a small percentage that cannot be identified. As expected, virtually all the failures occur when bringing up the first system. On subsequent systems, the failure rate drops dramatically.

- Approximately 20% of the returned cards are problem–free. These cards, typically, have the wrong jumper settings or the customer has problems with the software. This causes frustration for the customer and incurs a testing charge from Octagon.
- Of the remaining 80% of the cards, 90% of these cards fail due to customer misuse and accident. Customers often cannot pinpoint the cause of the misuse.
- Therefore, 72% of the returned cards are damaged through some type of misuse. Of the remaining 8%, Octagon is unable to determine the cause of the failure and repairs these cards at no charge if they are under warranty.

The most common failures on CPU control cards are over voltage of the power supply, static discharge, and damage to the serial and parallel ports. On expansion cards, the most common failures are static discharge, over voltage of inputs, over current of outputs, and misuse of the CMOS circuitry with regards to power supply sequencing. In the case of the video cards, the

most common failure is to miswire the card to the flat panel display. Miswiring can damage both the card and an expensive display.

- **Multiple component failures:** The chance of a random component failure is very rare since the average MTBF of an Octagon card is greater than 11 years. In a 7–year study, Octagon has never found a single case where multiple IC failures were not caused by misuse or accident. It is very probable that multiple component failures indicate that they were user–induced.

- **Testing "dead" cards:** For a card that is "completely nonfunctional", there is a simple test to determine accidental over voltage, reverse voltage or other "forced" current situations. Unplug the card from the bus and remove all cables. Using an ordinary digital ohmmeter on the 2,000 ohm scale, measure the resistance between power and ground. Record this number. Reverse the ohmmeter leads and measure the resistance again. If the ratio of the resistances is 2:1 or greater, fault conditions most likely have occurred. A common cause is miswiring the power supply.

- **Improper power causes catastrophic failure:** If a card has had reverse polarity or high voltage applied, replacing a failed component is not an adequate fix. Other components probably have been partially damaged or a failure mechanism has been induced. Therefore, a failure will probably occur in the future. For such cards, Octagon highly recommends that these cards be replaced.

- **Other over–voltage symptoms:** In over–voltage situations, the programmable logic devices, EPROMs and CPU chips, usually fail in this order. The failed device may be hot to the touch. It is usually the case that only one IC will be overheated at a time.

- **Power sequencing:** The major failure of I/O chips is caused by the external application of input voltage while the Micro PC power is off. If you apply 5V to the input of a TTL chip with the power off, nothing will happen. Applying a 5V input to a CMOS card will cause the current to flow through the input and out the 5V power pin. This current attempts to power up the card. Most inputs are rated at 25 mA maximum. When this is exceeded, the chip may be damaged.

- **Failure on power–up:** Even when there is not enough current to destroy an input described above, the chip may be destroyed when the power to the card is applied. This is due to the fact that the input current biases the IC so that it acts as a forward biased diode on power–up. This type of failure is typical on serial interface chips but can apply to any IC on the card.

- **Under rated power supply:** The board may fail to boot due to an under rated power supply. It is important that a quality power supply be used with the 5070 that has sufficient current capacity, line and load

regulation, hold up time, current limiting, and minimum ripple. The power supply for the 5070 must meet the startup risetime requirements specified in the ATX Power Design Guide, version 1.1, section 3.3.5. This assures that all the circuitry on the CPU control card sequences properly and avoids system lockup.

- **Excessive signal lead lengths:** Another source of failure that was identified years ago at Octagon was excessive lead lengths on digital inputs. Long leads act as an antenna to pick up noise. They can also act as unterminated transmission lines. When 5V is switch onto a line, it creates a transient waveform. Octagon has seen sub-microsecond pulses of 8V or more. The solution is to place a capacitor, for example 0.1 µF, across the switch contact. This will also eliminate radio frequency and other high frequency pickup.

Note Any physical damage to the CPU control card is **not** covered under warranty.

Table of Contents

List of Figures

List of Tables

Overview: Section 1 – Installation

Section 1 provides installation and programming instructions, startup options, and system configuration program examples. The following chapters are included:

Chapter 1:	Overview
Chapter 2:	Quick start
Chapter 3:	Setup programs
Chapter 4:	Save and run programs
Chapter 5:	Installing a different OS

Chapter 1: Overview

Description

The 5070 CPU control card is a single board computer in the Octagon Micro PC form factor. It is intended for higher–performance, low–power embedded control applications. The 5070 integrates serial communications, IDE hard disk port, floppy disk port, CompactFlash socket, a multifunctional parallel port, digital I/O, USB, keyboard, mouse and speaker ports, video, and a 10/100BaseT Ethernet port.

The 5070 comes with Datalight ROM–DOS installed and it is compatible with Windows NT, Windows 98, Windows CE, Linux, QNX, and DOS. Since the 5070 uses the same functional blocks as the other Octagon Micro PC™ cards, the circuitry has been fully proven as reliable and the software is compatible with the other software in the Micro PC series.

The 5070 can be used in a stand–alone mode or expanded through the Micro PC card cage or the PC/104 interface.

5070 major hardware features

CPU processor

The CPU is a ZF Micro ZFx86 128 MHz processor. It is designed for low–power applications. It can be configured to run at 33, 50, 66, 99, 100 or 128 MHz.

32 MB Surface Mount SDRAM

The 5070 comes with 32 MB of surface mount SDRAM. In OEM quantities it can be ordered with 16 MB surface mounted SDRAM.

Solid–state disk SSD1

SSD1 is a 2MB SMT boot flash that contains a 128K BIOS. In OEM quantities it can be ordered with 4 or 8MB SMT flash. The BIOS in SSD1 can be reprogrammed through the Z-tag interface.

CompactFlash socket

The CompactFlash socket accepts a Type 1 CompactFlash card. The CompactFlash appears as an IDE device to the system.

Hard disk and floppy disk ports

The hard drive and floppy drive are routed through an 80–pin connector on the top side of the board. The Octagon HDC-18-HDD/FDD cable breaks out the 80 pins into a standard 40–pin IDE hard drive connector and a standard 34–pin floppy drive connector. Note that the IDE connector does not supply +5V to a hard drive. The BIOS supports up to two IDE drives and two floppy drives. However, the HDD/FDD cable only supports one hard drive and one floppy drive.

Digital I/O

The 4–bit digital I/O port provides two input lines and two output lines. These lines will interface with logic devices, switch inputs, LEDs and industry standard opto module racks. The I/O lines are 0–5V logic compatible.

USB

The 5070 provides two USB ports, which are available when using an operating system that supports USB. Both channels are open HCI compliant.

Ethernet

- The 5070 provides a 10/100 BaseT Ethernet port and supports the IEEE 802.3 Ethernet standard.

Serial ports protected against ESD

The 5070 has two serial ports with combinations of RS–232C and RS–422/485 interfaces. These serial ports have the following common specifications:

- IEC1000, level 3, ESD protection specification
  - — Contact discharge ±6 kV
  - — Air–gap discharge ±8 kV
- Backdrive protection
- 16550 compatible

- Up to 115.2K baud
- 16–byte FIFO buffers
- Jumper-selectable terminations for RS–422/485
- Enabled and disabled in Setup

Multifunctional printer port

The 5070 incorporates the latest enhanced parallel port and includes unidirectional, bi-directional, ECP and EPP modes.

The following represent applications in the multifunctional parallel port:

- LPT1 for PC compatible printers
- 17 general purpose digital I/O lines
- Up to a 4 x 4 matrix keypad
- 4–line alphanumeric display

Multipurpose connectors

Most of the peripherals on the 5070 are routed through two 80–pin connectors. The HDC-18-HDD/FDD drive cable breaks out one 80–pin connector into a floppy connector and a hard drive connector. The 5070 HDC-18-MPC-MULTIPORT interface cable breaks out the other 80–pin connector into the serial, parallel, keyboard and mouse, digital I/O, USB, and speaker ports as well as the reset and AT battery interfaces.

Hardware reset

A hardware reset ensures complete reset of the system and all attached peripherals. A hardware reset can be done by any of the following methods:

- An expired watchdog timer cycle
- Depressing the reset switch
- Cycling power
- Power supervisor reset

Video

The 5070 supports VGA, SVGA, SXGA monitors, and flat panel displays.

Real time calendar/clock with battery–backup

The real time clock is fully AT compatible. An optional off–card battery powers the real time clock when the 5 volt supply is removed.

Setup information stored in EEPROM for high reliability

Loss of Setup data is serious in industrial applications. Most PCs store Setup information in battery-backed CMOS RAM. If the battery fails or is replaced during routine maintenance, this information is lost. Without a keyboard and monitor in embedded applications, time consuming re-initialization is required. The 5070 stores the Setup information in EEPROM with 512 bytes available to the user. Software routines to use this available memory come with the 5070.

Watchdog timer added for safety

The watchdog timer resets the system or generates an NMI (nonmaskable interrupt) if the program stops unexpectedly. The watchdog is enabled, disabled and strobed under software control. The time–out period is programmable from 0.5 to 2 seconds. Using INT17 functions in DOS, the timeout period can be extended to 64 seconds.

PC/104 16–bit interface

The PC/104 interface accepts an 8– or 16–bit PC/104 expansion board. Up to two PC/104 expansion boards may be stacked on the 5070.

5 Volt only operation lowers system cost

- 5V ±5%
- +12V and –12V supplied to PC/104 connector from the power connector or the ISA bus; not required for 5070 operation

Rugged environmental operation

■ Operating temperature	–40° to 85°C, 99MHz and below –40° to 70°C, 128MHz
Nonoperating temperature	–55° to 95°C, nonoperating
■ Relative humidity	5% to 95% noncondensing
■ Altitude	–100 to 10,000 m
■ Shock	40g, 3 axis
■ Vibration	5g, 3 axis

Size

4.5 in. x 4.9 in. x .92 in.

5070 major software features

Diagnostic software verifies system integrity automatically

The 5070 has built-in diagnostic software that can be used to verify on-card I/O and memory functions. On power-up, a series of tests is performed. If a problem occurs, the failed test can be identified by a flashing LED or a beep code. The test is performed automatically every time the system is reset or powered up. Memory verification does not require software, test equipment, monitor, keyboard, disks, or test fixtures. See the "*Troubleshooting"* chapter for a listing of tests and failures and their descriptions.

"Instant DOS" operating system

Datalight ROM–DOS is in flash. This means that this version is always present on power-up. The system boots and operates the same way as a desktop PC. Since all software and hardware are included, the system is fully operational "out of the box."

Phoenix software BIOS

The 5070 has a Phoenix Software BIOS with Octagon BIOS extensions. The BIOS extensions include Datalight's FlashFX and Octagon's INT17 functions.

Octagon BIOS extensions

On–board BIOS extensions allow easy access to watchdog timer functions, CMOS memory, etc.

Boot sequence

A 5070 can be configured to boot from on-card flash, CompactFlash, a floppy, hard disk, or CD-ROM.

Chapter 2: *Quick start*

This chapter covers the basics of setting up a 5070 CPU card. Refer to the 5070 component diagrams, Figures 2–1 and 2–2, for the location of the various connectors. The following topics are discussed:

- Panel mounting, stacking, or installing the 5070 into an Octagon Micro PC card cage
- Connecting a monitor and keyboard
- Running a demo program

WARNING!

The 5070 can not be installed in a PC. These cards are designed to be independent CPU cards only, not accelerators or coprocessors.

Hardware installation

WARNING!

The 5070 CPU card contains static–sensitive CMOS components. The card is most susceptible to damage when it is plugged into a card cage. To avoid damaging your card and its components:

- **Ground yourself before handling the card and observe proper ESD precautions**
- **Disconnect power before removing or inserting the card in a card cage, or before removing or inserting a PC/104 expansion board**

Figure 2–1 5070 component diagram – top

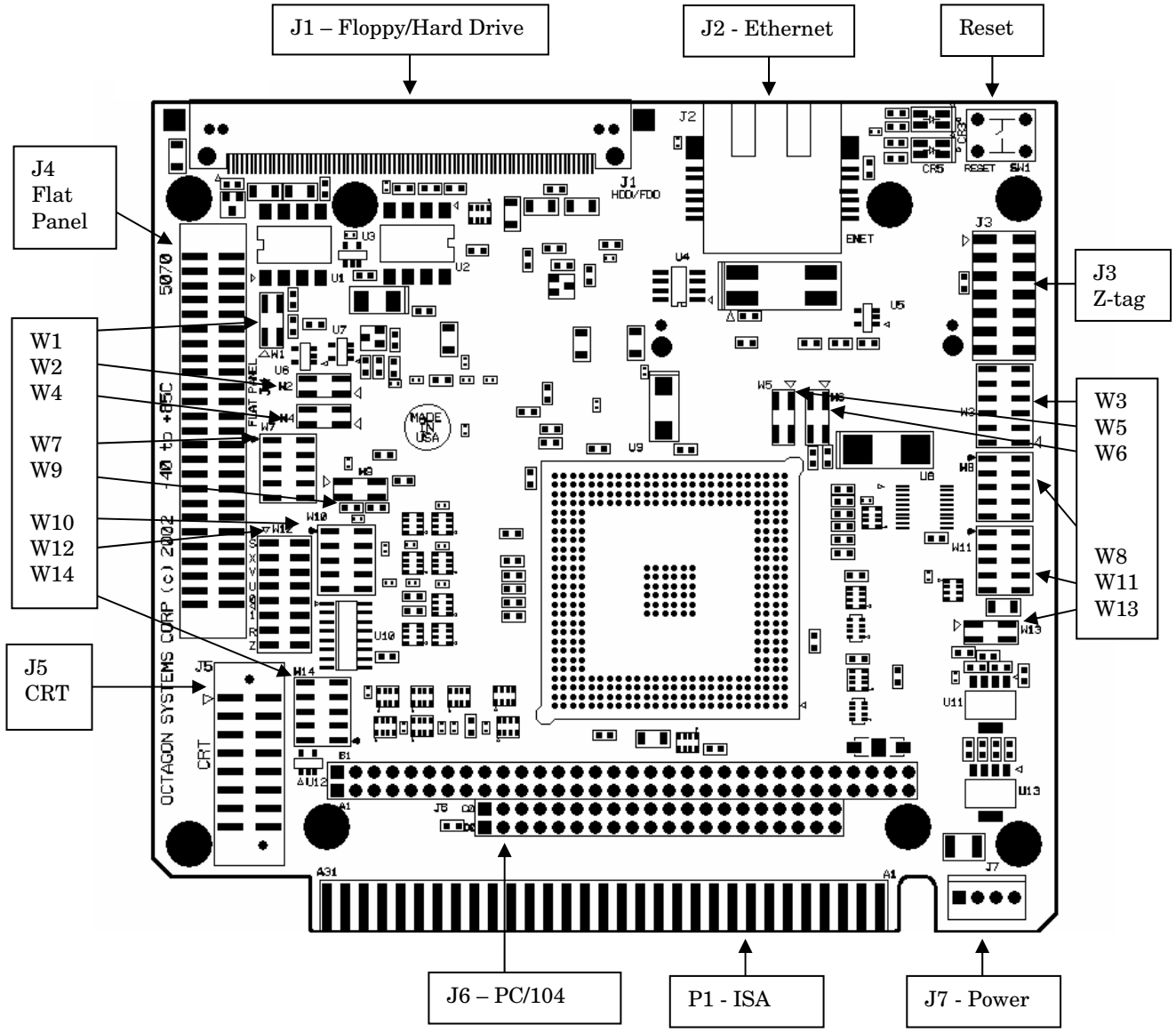


Figure 2–2 5070 component diagram – bottom

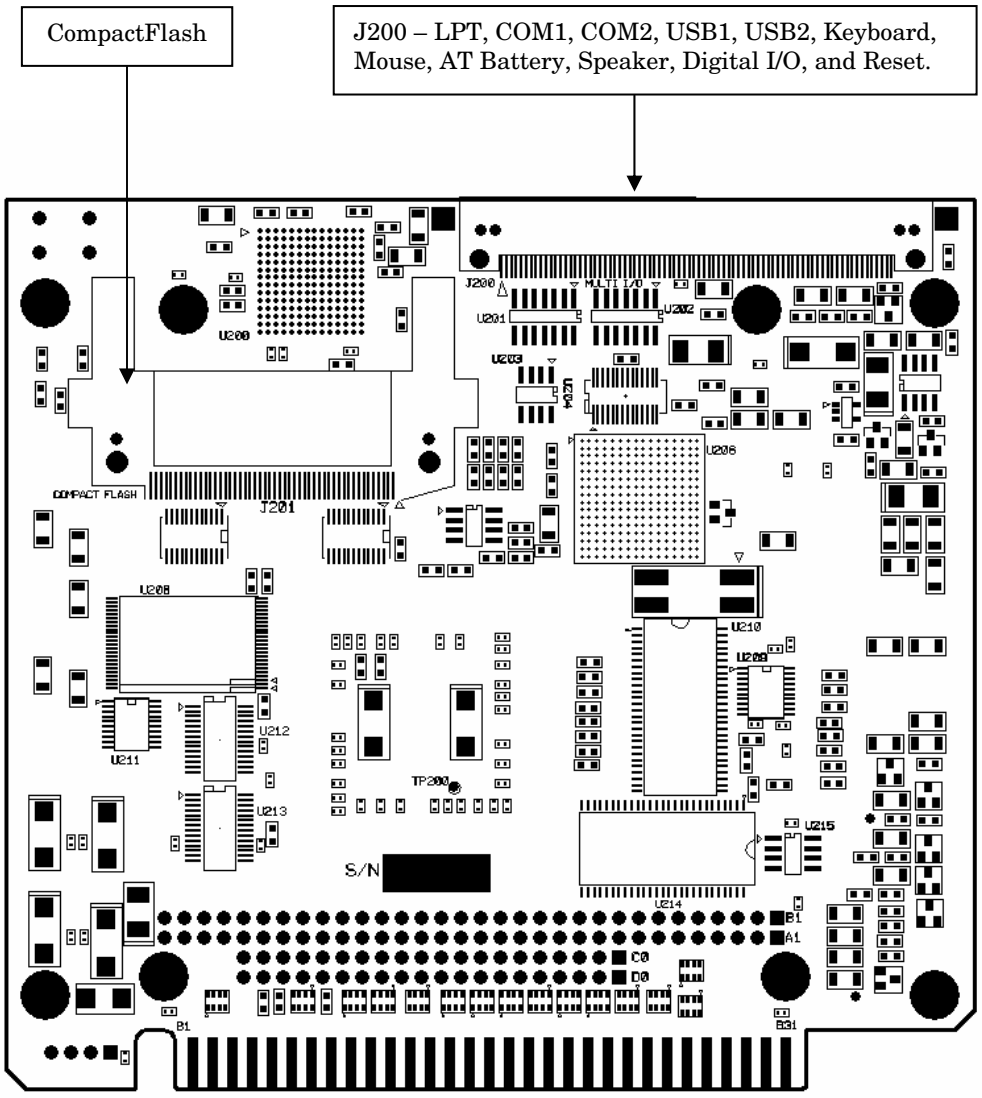


Figure 2–3 5070 dimensions (inches)

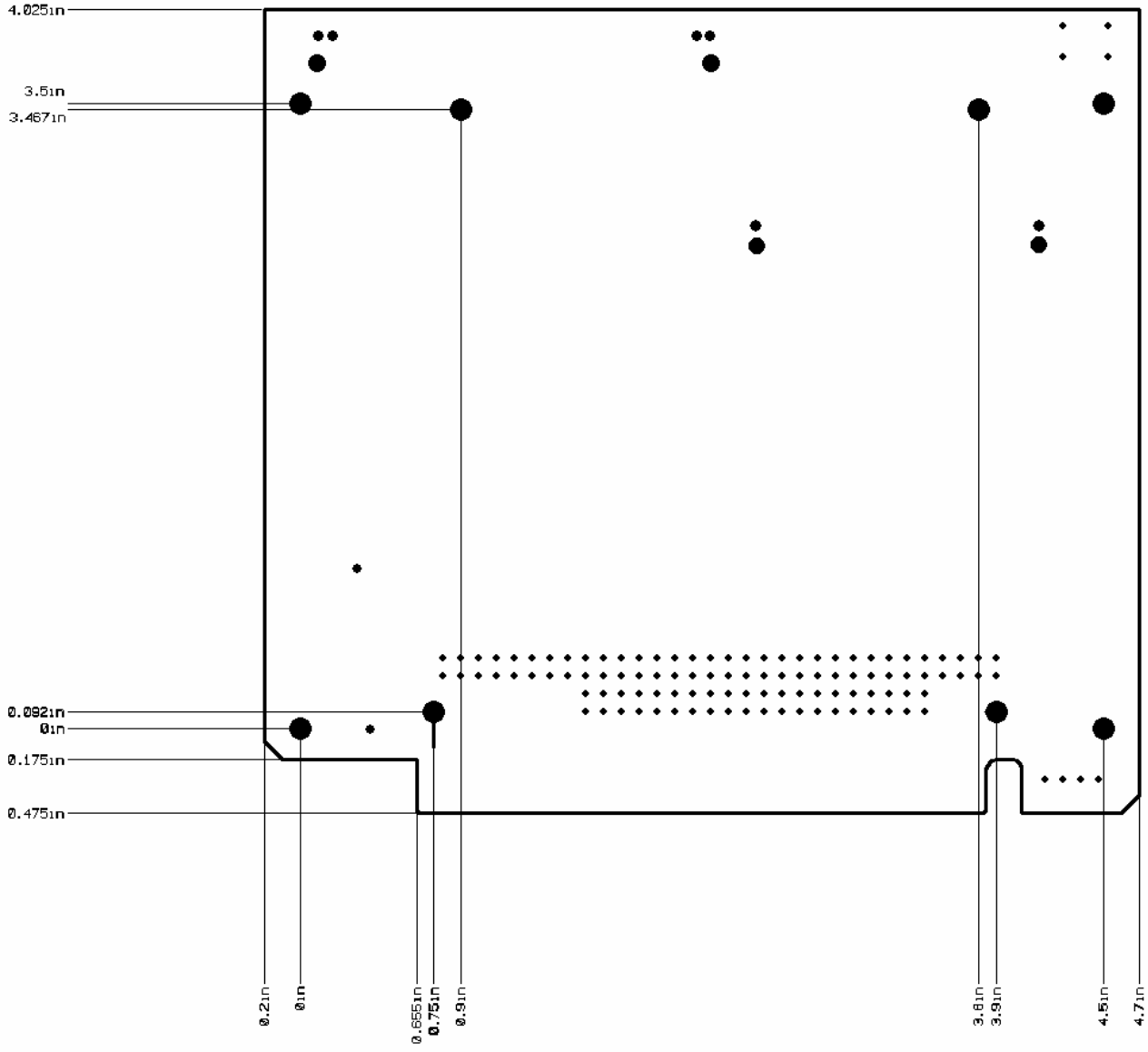
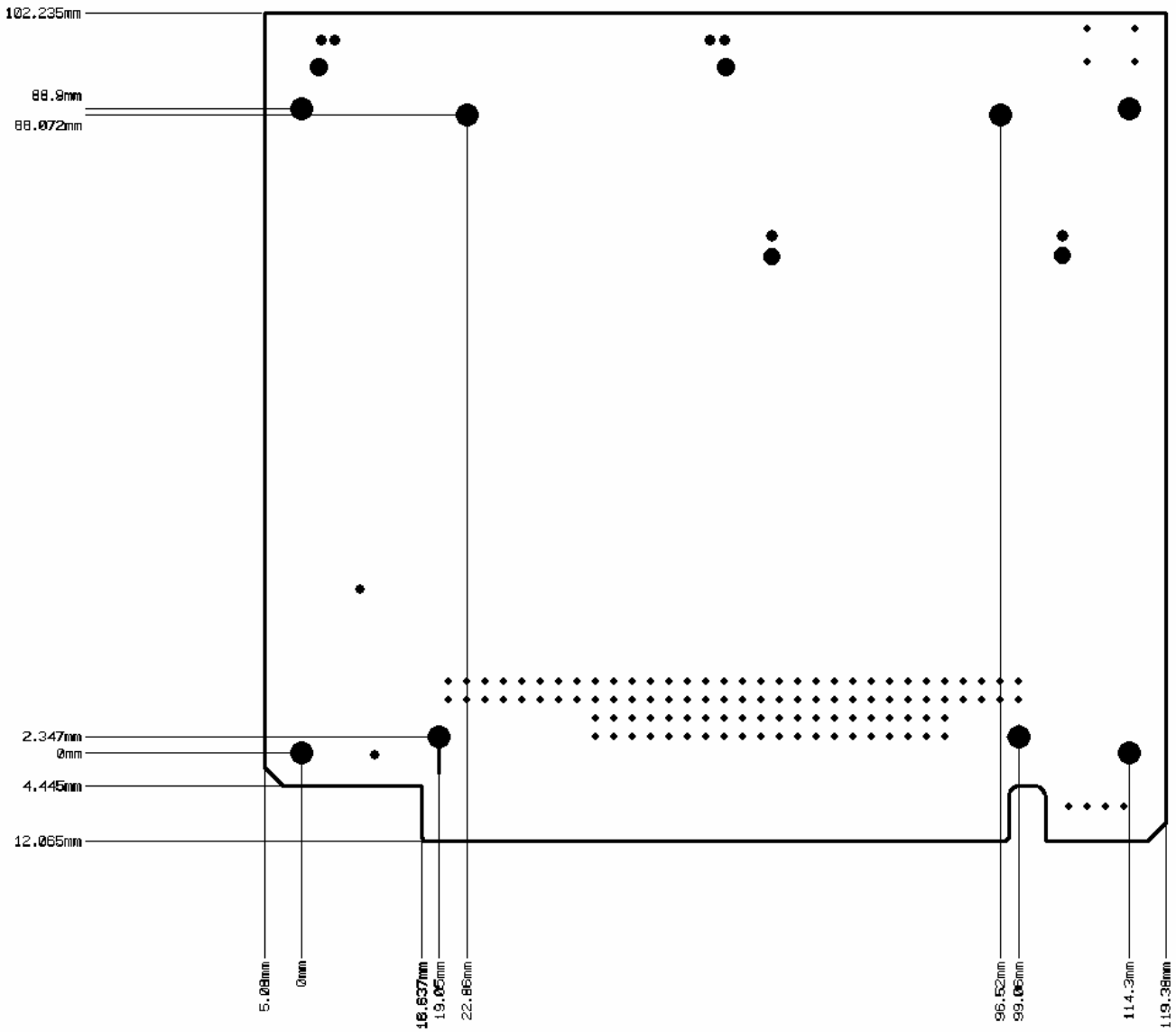


Figure 2–4 5070 dimensions (mm)



The 5070 can be installed in one of several ways:

- Plugging it directly into an 8–bit Micro PC card cage
- Using the optional PC mounting bracket and plugging it into any 8–bit passive ISA backplane
- Panel mounting it using the eight mounting holes
- Stacking it with other Micro PC cards

These methods all require the 5070 HDC-18-MPC-Multiport cable, #6240, and the VGA-12 video cable, #4865. Refer to the following section for information on the two HDC-18 cables.

5070 HDC-18 cables

The 5070 has two 80-pin connectors that provide an interface for several of the card functions. Octagon has two HDC-18 cables that break out these connectors into industry-standard interfaces.

Connector J1, on the front of the board, uses the HDC-18-HDD/FDD drive cable. This cable provides a floppy connector and a hard drive connector. Note that both of these connectors only support one device each. Also, these connectors do not provide power to the drives.

Connector J200, on the back side of the board, uses the HDC-18-MPC-Multiport Interface Cable. This cable provides an interface for LPT, COM1, COM2, USB1, USB2, Keyboard, Mouse, AT Battery, Speaker, Digital I/O, and Reset. Figures 2-5 and 2-6 show these two cables.

Table 2–1 HDC-18-HDD/FDD cable description

J1 connector	
Side A	**Side B**
J2A – IDE drive	J3 – Floppy drive
J2B – IDE drive	

J2A, J2B – IDE drives

Interfaces to two standard 40-pin IDE devices, such as hard drives or CD-ROM drives.

J3 – Floppy drive

Interfaces to a standard 34-pin floppy drive.

Figure 2–5 HDC-18-HDD/FDD drive cable

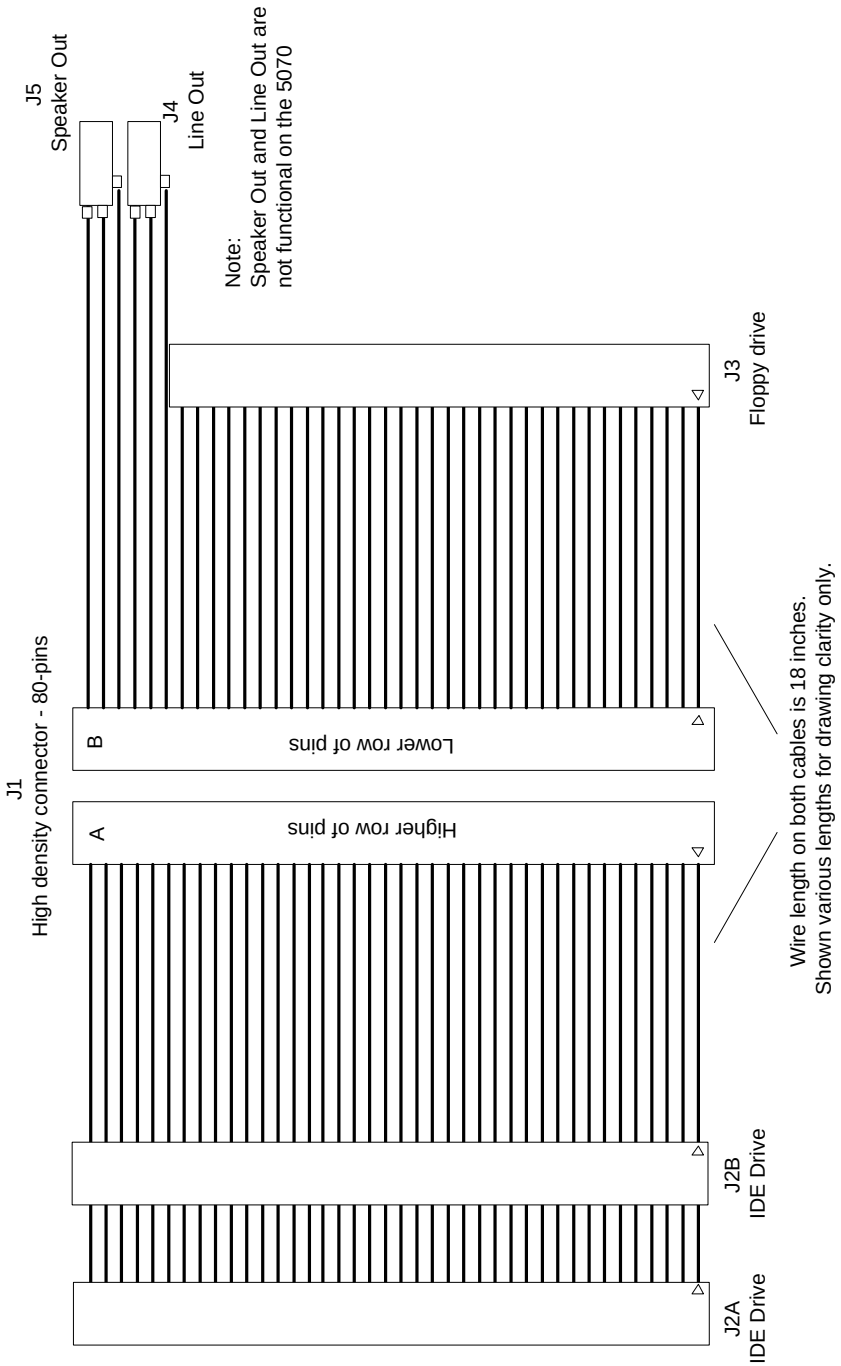


Table 2–2 HDC-18-MPC-Multiport cable description

J200 connector	
Side A	**Side B**
J8 – PS-2 Mouse	J7 – PS-2 Keyboard
Reset switch	J6 – USB 2
J10 – Speaker	J5 – USB 1
J9 – AT battery	J11 – Digital I/O
J2A – LPT1	J4 – COM2
J2B – LPT1	J3 – COM1

J2A, J2B – LPT

J2A is a standard DB-25 connector that can connect directly to a printer. J2B is used to connect to the Octagon 2010 LCD Display/Keypad Interface.

J3, J4 – COM1, COM2

J3 and J4 are standard DB-9 connectors that connect to RS-232, RS-422, and RS-485 serial devices.

J5, J6 – USB1, USB2

J5 and J6 interface directly to standard USB cables.

J7, J8 – Keyboard and mouse

J7 and J8 are standard PS-2 keyboard and mouse connectors.

J9 – AT battery

J9 mates with a standard AT style, 3.6V battery.

J10 – Speaker

8 ohm speaker.

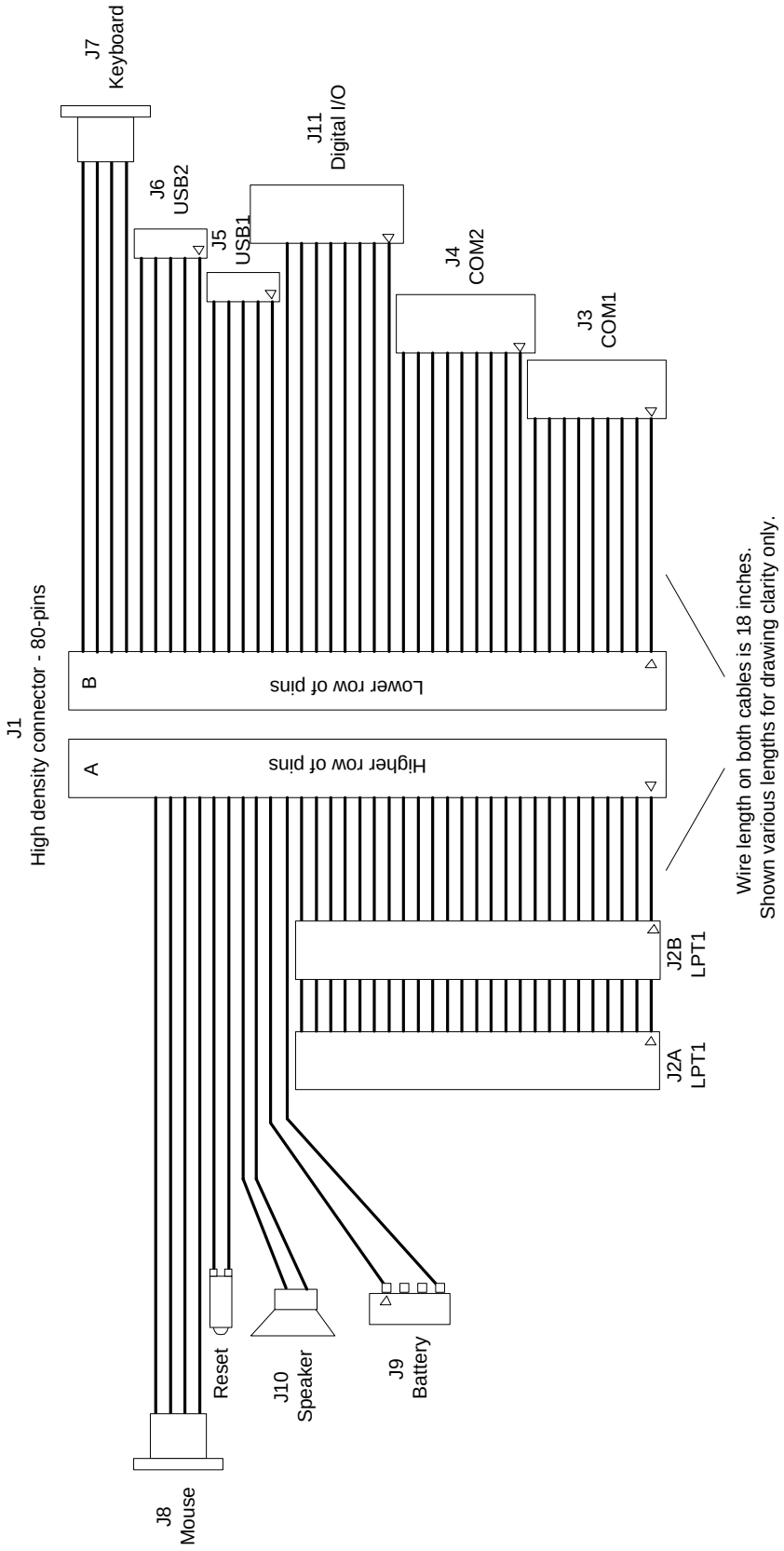
J11 – Digital I/O

J11 is the digital I/O interface connector.

Reset

A momentary contact switch is used to reset the 5070.

Figure 2–6 5070 HDC-18-MPC-Multiport cable



Using a Micro PC card cage

To install the 5070 in a Micro PC card cage, you will need the following equipment (or equivalent):

- 5070 CPU card
- Micro PC card cage (5xxx Card Cage)
- Power module (510x or 71xx Power Module)
- Optional – a device with an operating system. The device could be floppy, hard disk, or CD-ROM. The operating system can be Windows NT, Windows 98, Windows CE, Linux, QNX, or DOS. This is optional because ROM–DOS is installed in SSD1.
- 5070 HDC-18-MPC-Multiport cable, #6240
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor
- 5070 Utilities zip file (see page 139)

1. Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the 5070.
2. Attach the Octagon power module to the card cage following the instructions supplied with the power module.
3. Make sure power to the card cage is OFF.

Refer to Figure 2-7 for the following:

4. Before installing the 5070 into the card cage, connect the VGA-12 cable into J5.
5. Before installing the 5070 into the card cage, connect the 5070 HDC-18-MPC-Multiport cable into J200.
6. Slide the 5070 into the card cage. Refer to Figures 2–8 and 2–9 for the correct orientation of the 5070 and an illustration of a CPU card in a Micro PC card cage.
7. Connect a VGA monitor to the VGA-12 cable, and a PS-2 style keyboard to J7 of the HDC-18-MPC-Multiport cable.

Figure 2–7 5070, VGA monitor, and PS-2 compatible keyboard

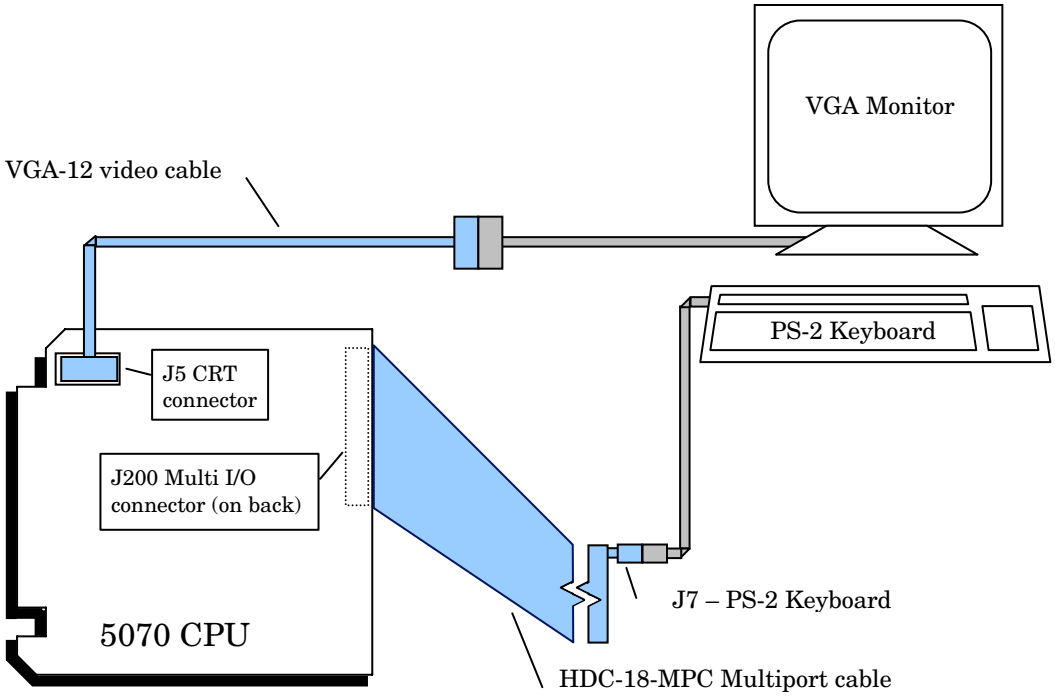


Figure 2–8 Edge connector orientation

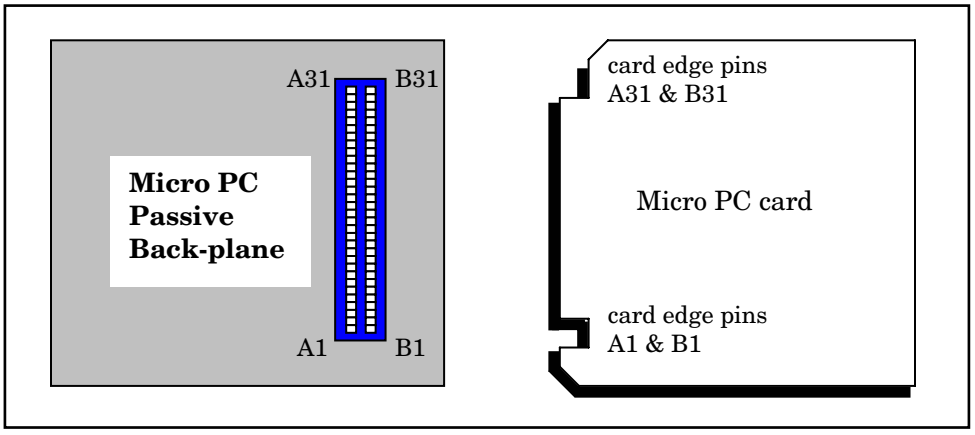
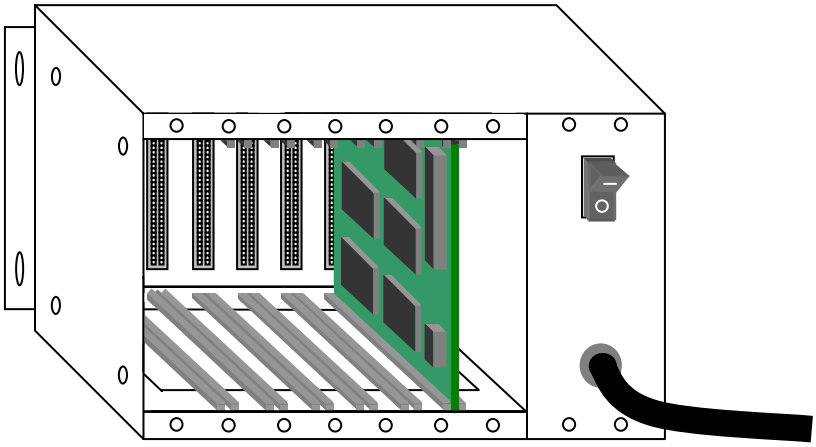


Figure 2–9 Populated Micro PC card cage



Panel mounting or stacking the 5070

To panel mount or stack the 5070, you will need the following equipment (or equivalent):

- 5070 CPU card
- +5V power supply and cable. Refer to the *Power Supply Requirements* section, page 36.
- Optional – a device with an operating system. The device could be floppy, hard disk, or CD-ROM. The operating system can be Windows NT, Windows 98, Windows CE, Linux, QNX, or DOS. This is optional because ROM–DOS is installed in SSD1.
- 5070 HDC-18-MPC-Multiport cable, #6240
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor
- 5070 Utilities zip file (see page 139)
- Qty 8 – #4-40 screws, #4-40 threaded hex standoffs, #4 internal star lock washers
- 5252MB stacking kit, #3590 (required for stacking only)

1. Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the 5070.

Refer to Figures 2-10 and 2-11 for the following:

2. Use the #4–40 standoffs, screws, and washers and secure them in the eight holes on the 5070. Refer to Figures 2-3 and 2-4 on pages 23 and 24 for the center–to–center mounting hole dimensions.

WARNING!

All eight standoffs, screws and washers must be used to secure the 5070. Using all of the standoffs ensure full support of the board. Also, verify that the washers and standoffs do not touch any of the component pads adjacent to the mounting holes. Damage may occur at power–up.

3. Connect the power supply +5V and ground wires to the 5070 power connector, J7, of the 5070. Refer to Figure 2–12 and Table 2–2.

WARNING!

Miswiring the voltage at J7 of the 5070 or at the power connector of the 5252MB stacking kit (reversing +5V and ground, or applying a voltage greater than +5V), will destroy the card and void the warranty!

Refer to Figure 2-13 for the following:

4. Connect the VGA-12 cable into J5.
5. Connect the HDC-18-MPC-Multiport cable into J200.
6. Make sure power to the power supply is OFF. Connect the power supply cable to J7.
7. Connect a VGA monitor to the VGA-12 cable, and a PS-2 style keyboard to J7 of the HDC-18-MPC-Multiport cable.

Figure 2–10 Panel mounting the 5070

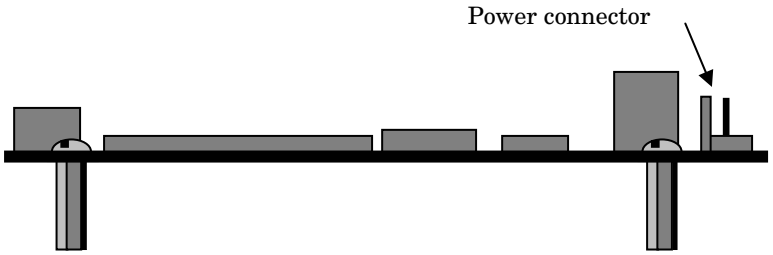


Figure 2–11 Stacking the 5070

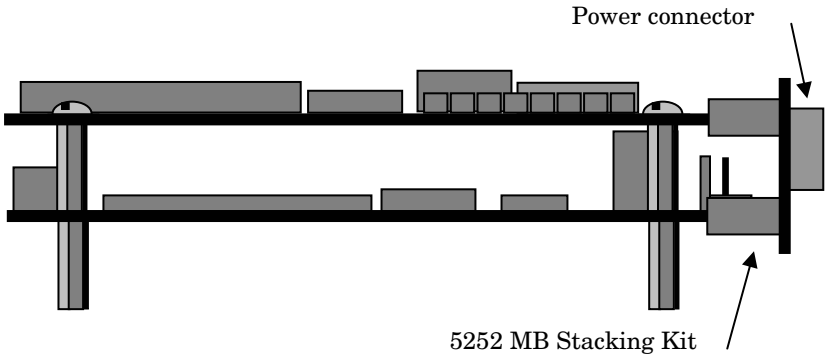


Figure 2–12 Power connector: J7 diagram

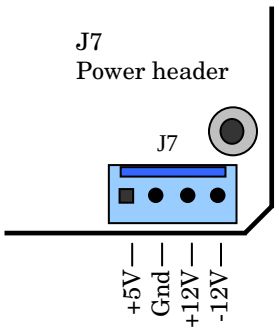
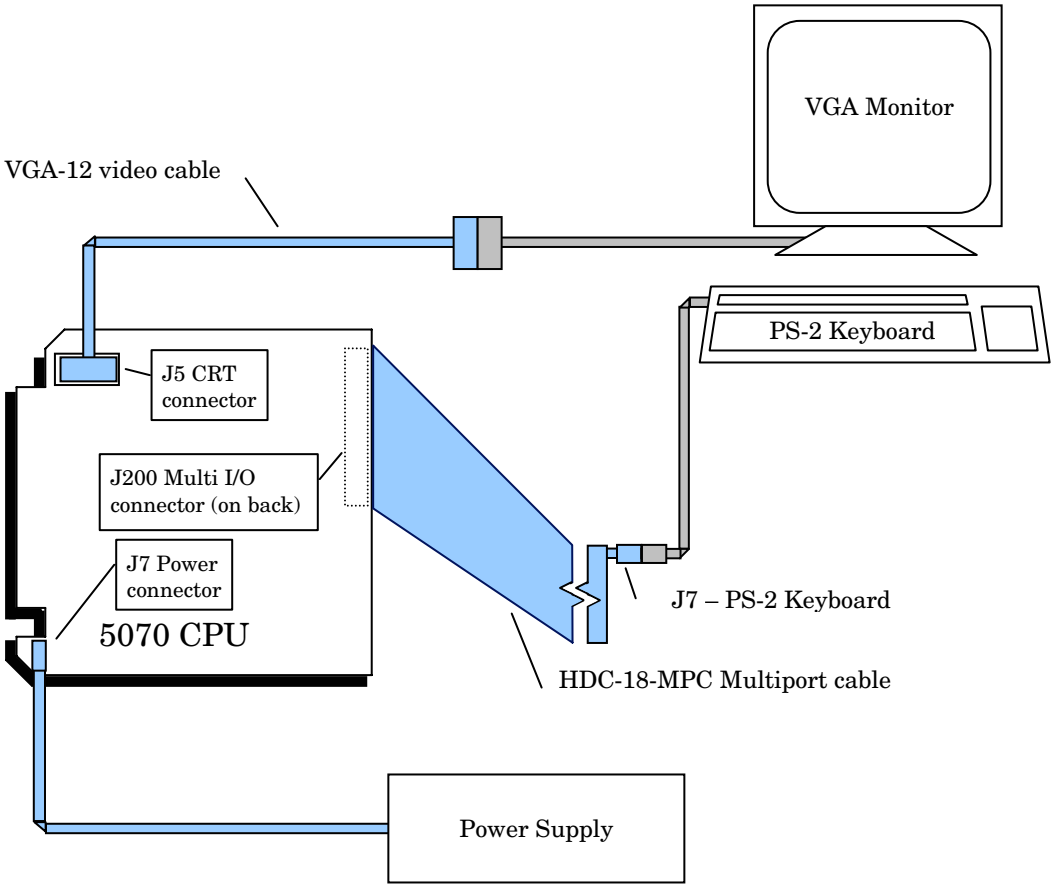


Table 2–3 Power connector: J7 pin-out

J7 – Power connector	
Pin#	**Function**
1	+5v
2	GND
3	+12V
4	–12V

Figure 2–13 5070, VGA monitor, PS–2 compatible keyboard, and power supply



Using the 5070 in a passive ISA backplane

To use the 5070 into a passive ISA backplane, you will need the following equipment (or equivalent):

- 5070 CPU card
- Unterminated backplane
- Mounting bracket (optional)
- 5V power supply and cable. Refer to the *Power Supply Requirements* section, page 36.
- Optional – a device with an operating system. The device could be floppy, hard disk, or CD-ROM. The operating system can be Windows NT, Windows 98, Windows CE, Linux, QNX, or DOS. This is optional because ROM–DOS is installed in SSD1.
- 5070 HDC-18-MPC-Multiport cable, #6240
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor
- 5070 Utilities zip file (see page 139)

1. Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the 5070.
2. Connect the power supply +5V and ground wires to the power connector of the unterminated backplane. Refer to the *Power Supply Requirements* section, page 36.

WARNING!

Miswiring the voltage to the backplane (reversing +5V and ground, or applying a voltage greater than +5V), will destroy the card and void the warranty!

3. Make sure power to the backplane is OFF.
4. Insert the 5070 into a connector on the backplane. Refer to Figure 2–14. Take care to correctly position the cards' edge with the connector of the backplane. Figure 2–8 shows the relative position of the 5070 card as it is installed into a backplane.

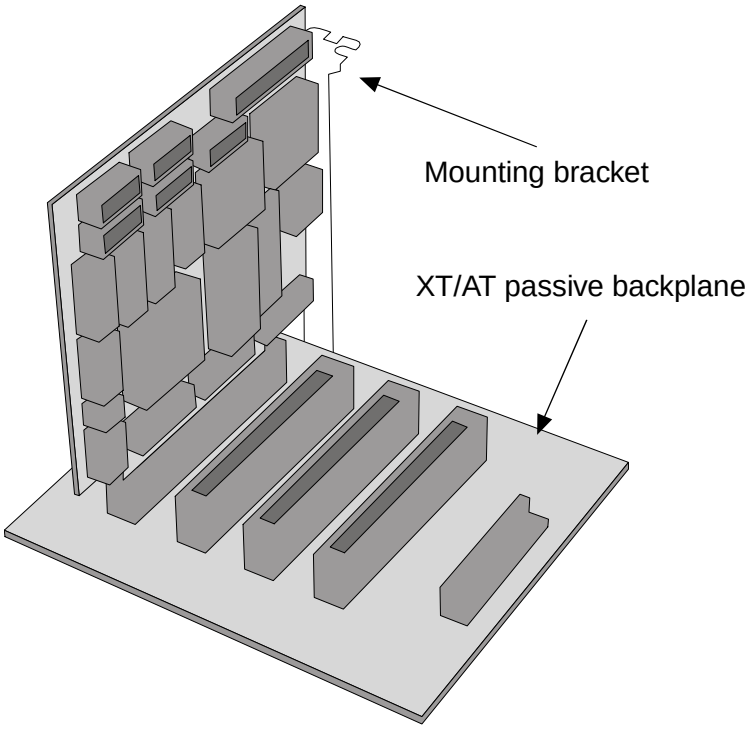
WARNING!

Incorrectly plugging the card into the backplane will destroy the card and void the warranty!

Refer to Figure 2-7 for the following:

5. Connect the VGA-12 cable into J5.
6. Connect the HDC-18-MPC-Multiport cable into J200.
7. Connect a VGA monitor to the VGA-12 cable, and a PS-2 style keyboard to J7 of the HDC-18-MPC-Multiport cable.

Figure 2–14 Using a passive ISA backplane



5070 power supply requirements

The 5070 is designed to operate from a single +5 VDC supply, connected at J7. The typical current requirements for the 5070 is listed in the *Technical data* appendix. If you are using the PC/104 interface, you may also require ±12 VDC.

The user should consider factors such as the power cable conductor gauge, number and length of conductors, mating connectors, and the power requirements of external devices such as hard drives, floppy drives, displays, mouse, and keyboard.

It is important that a quality power supply be used that has sufficient current capacity, line and load regulation, hold up time, current limiting, and minimum ripple.

The power supply for the 5070 must meet the startup risetime requirements specified in the ATX Power Design Guide, version 1.1, section 3.3.5. This assures that all the circuitry on the 5070 sequences properly and avoids system lockup.

Also, select a power supply that discharges quickly. If large power supply output capacitors are used, powering the system down and then up may lock up the 5070. If the power supply does not drain below 0.7V, the CMOS components on the 5070 will act like diodes and forward bias, potentially damaging the 5070 circuitry.

The proper selection of a quality power supply ensures reliability and proper functioning of the 5070.

WARNING!

Make sure the power supply is OFF when connecting the power cable to the 5070 board. Damage to the 5070 may occur if the power is ON when connecting the power cable.

Running a demo program

1. Power on the 5070 CPU card.
2. A logon message similar to the one below will appear on your monitor:

```
PhoenixBIOS(TM) A586 Version 1.03
Copyright (C) 1985-1992 Phoenix Technologies Ltd.
All Rights Reserved
Octagon Systems Corp. 128MHz 5070 CPU
Release x.xx-mm/dd/yy

640K Base Memory, 03072K Extended

5070 INT 17h BIOS extension v1.15 Copyright (c) 1995-2000, Octagon
Systems

Datalight FlashFX
V4.04.292  386 DOS
Copyright (c) 93-99
Patent US#5860082
  Octagon Systems Vx.xx - 5070
  SETSSD SSD1 /BEFORE
  1MB AMD Flash detected in SSD1
```

```
Starting ROM-DOS...

HIMEM v7.10 (Revision 3.00.42)
Copyright (c) 1989-2000 Datalight, Inc.
Using PTL A20 Control (ON)
32 XMS handles available.
Minimum HMA usage is 0K.

VDISK v6.22 (Revision 3.00.42)
Copyright (c) 1989-2000 Datalight, Inc.

Installed 960KB XMS RAM disk as drive E:

5070 C:\>_
```

If you do not get the proper logon message:

- Make sure all jumpers are set to factory defaults
- Make sure that all cables are installed properly
- Make sure that all power connections are properly made.
- If the system still does not respond, refer to the *Troubleshooting* chapter.

3. Use the directory command to make sure your equipment and software are working properly. Enter:

```
5070 C:\> DIR
```

A directory listing of ROM-DOS files stored in SSD1 should appear:

```
Volume in drive C is SSD1
 Volume Serial Number is 281F-9D7D
 Directory of C:\

COMMAND  COM        34,565 01-21-2000   6:22a
CONFIG   SYS            78 04-26-2000   1:51p
AUTOEXEC BAT            43 09-13-1999   2:14p
DOS          <DIR>          04-28-2000  12:09a
UTILS        <DIR>          04-28-2000  12:09a
        6 file(s)         34,686 bytes
                          469,472 bytes free
```

4. You are now ready to install files on the 5070 CPU card.

What’s next

1. To run BIOS setup and configure the system, refer to the *Setup programs* chapter.
2. To connect a floppy, hard drive, or a CompactFlash device, refer to the *External drives* chapter.
3. To install a different operating system, refer to the *Installing a different OS* chapter.
4. To use a serial console, or transfer files from a host PC, refer to the *Console Devices* chapter.

Chapter 3: Setup programs

This chapter discusses running the Setup configuration program on the 5070 CPU card. Setup configures devices set up by the BIOS such as serial ports, floppy drives, etc.

Setup

Setup can be entered by pressing the “F2” key during the BIOS POST sequence (this occurs between the memory test and boot).

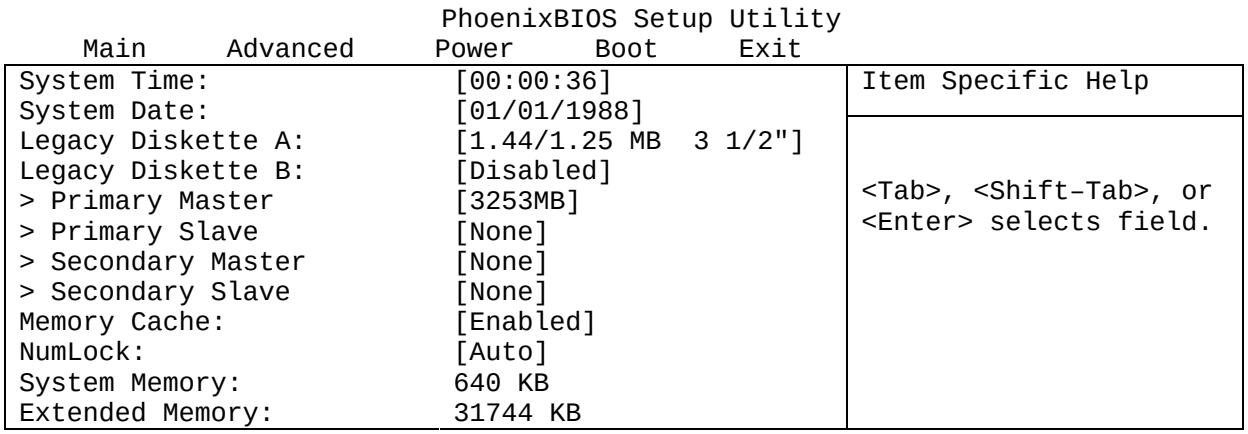
Also, by removing the “S” jumper W12[1–2], you will force the setup to revert to the factory programmed defaults shown in the following menus. This allows the user to reconfigure the setup.

The system will display the 5070 CPU card PhoenixBIOS Setup Utility Main menu. Select the submenu by using the up/down arrows, then press <ENTER> (when using a monitor connected to the 5070).

Note Options having an asterisk are default settings.

Main menu

The Main menu allows you to set the basic system configuration.



System Time: Sets the time for the system clock

System Date: Sets the date for the system clock

Legacy Diskette A: Enables or disables a legacy floppy disk drive. Choices are Disabled, 360 KB 5 ¼", 1.2 MB 5 ¼", 720 KB 3 ½", 1.44/1.25 MB 3 ½", 2.88 MB 3 ½"

Legacy Diskette B: Enables or disables a second legacy floppy disk drive. Note, however, that although the 5070 supports two floppy drives, the HDC-18-HDD/FDD drive cable only has one connector for a floppy disk drive.

> Primary Master: Accesses submenu for a Primary Master disk drive. Options are None, IDE Removable, CD–ROM, ATAPI Removable, Other ATAPI, User, and Auto.

> Primary Slave: Same as Primary Master

> Secondary Master: Same as Primary Master. Note, however, that the 5070 only supports two IDE devices.

> Secondary Slave: Same as Primary Master. Note, however, that the 5070 only supports two IDE devices.

Memory Cache: Enables or Disables the memory cache.

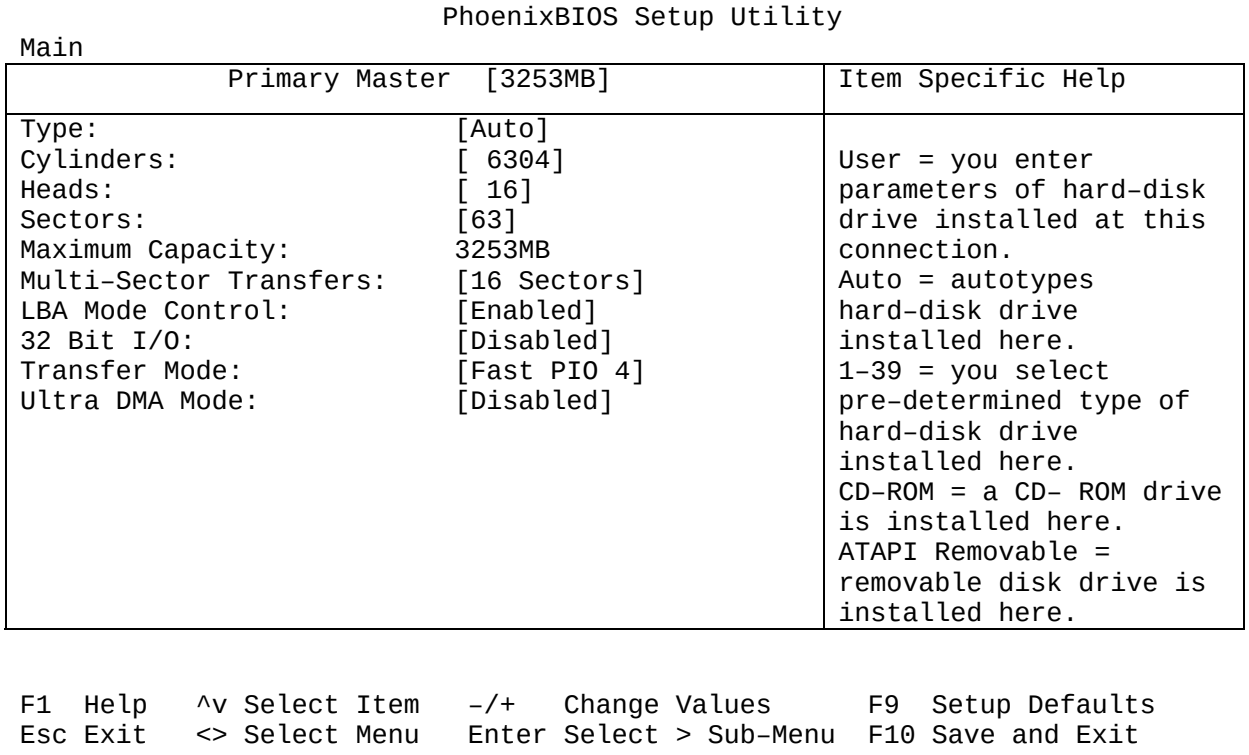
NumLock: Auto, On, or Off

System Memory: Displays the amount of system memory which is on the card

Extended Memory: Displays the amount of extended memory on the card

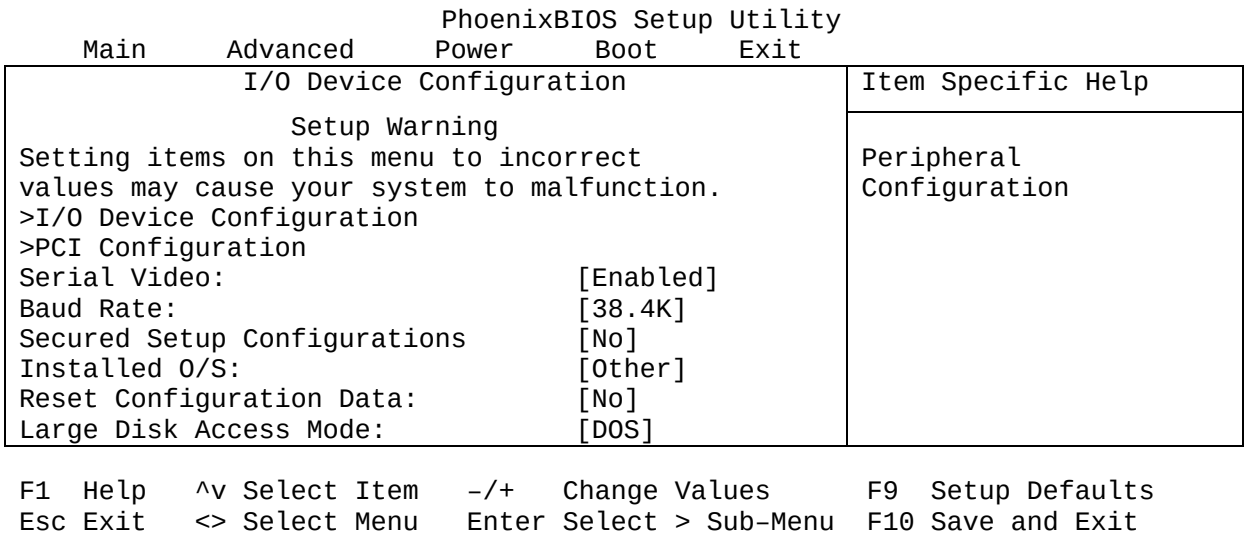
Hard drive submenus

The Hard drive submenus allow you to set the primary/secondary/master/slave parameters. Except for older disk drives, the Auto selection will detect and display the correct parameters.



Advanced menu

The Advanced menu allows you to set advanced system configuration. Note that if items are incorrectly set in this menu, the system might malfunction.



Serial Video: Enabled, Disabled. Enables redirection of video and keyboard to COM1.

Baud Rate: 9600, 19.2K, 38.4K, 57.6K, 115K. Selects baud rate for serial console.

Secured Setup Configurations: Yes or No. Yes prevents the operating system from overriding selections you have made in Setup.

Installed O/S: Other, Win95. Selects the operating system you use most often.

Reset Configuration Data: Yes or No. Yes erases all configuration data in a section of memory for ESCD (Extended System Configuration Data) which stores the configuration settings for non–PnP plug in devices. Select Yes when required to restore the manufacturer's defaults.

Large Disk Access Mode: DOS, Other. Select DOS if you have DOS. Select Other for another operating system such as Unix.

I/O Device Configuration submenu

The I/O Device Configuration submenu allows you to set the I/O configurations.

```
                    PhoenixBIOS Setup Utility
  Advanced
```

I/O Device Configuration		Item Specific Help
USB Host Controller:	[Enabled]	
PS-2 Mouse:	[Auto Detect]	
Serial port A:	[Enabled]	
Base I/O address:	[3F8]	
Interrupt:	[IRQ 4]	
Serial port B:	[Enabled]	
Base I/O address:	[2F8]	
Interrupt:	[IRQ 3]	
Parallel port:	[Enabled]	
Mode:	[Bi–directional]	
Base I/O address:	[378]	
Interrupt:	[IRQ 7]	
Floppy disk controller:	[Enabled]	
Local Bus IDE Adapter:	[Enabled]	

```
F1  Help   ^v Select Item   -/+   Change Values      F9  Setup Defaults
Esc Exit   <> Select Menu   Enter Select > Sub–Menu  F10 Save and Exit
```

USB Host Controller: Disabled, Enabled. Disables or Enables the USB hardware (Disabled resources will be freed up for other uses.)

PS-2 Mouse: Disabled, Enabled, Auto Detect. Frees up IRQ12 if disabled.

Serial port A: Disabled, Enabled, Auto. Enabled allows user to set configuration, while Auto uses the BIOS or OS configuration.

Base I/O address: Disabled, Enabled, Auto. Enabled allows user to set 3F8, 2F8, 3E8, 2E8

Interrupt: IRQ3, IRQ4

Serial port B: Same as Serial Port A.

Base I/O address: 3F8, 2F8, 3E8, 2E8

Interrupt: IRQ3, IRQ4

Parallel port: Disabled, Enabled, Auto. Enabled allows user to set configuration, while Auto uses the BIOS or OS configuration.

Mode: Output only, Bi–directional, EPP, ECP

Base I/O address: 378, 278, 3BC

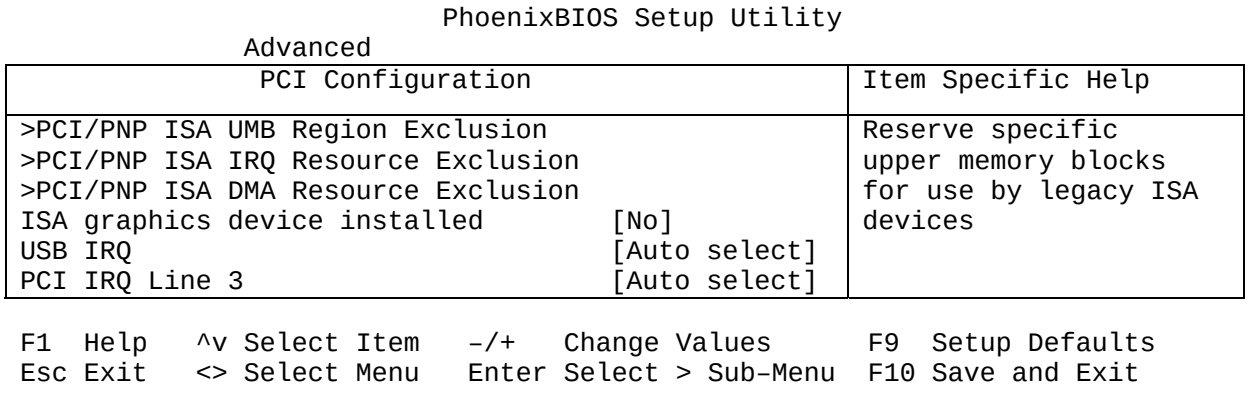
Interrupt: IRQ5, IRQ7

Floppy disk controller: Disabled, Enabled, Auto. Enabled allows user to set configuration, while Auto uses the BIOS or OS configuration.

Local Bus IDE Adapter: Disabled, Enabled. Enables the integrated local bus IDE adapter.

PCI Configuration submenu

The I/O Device Configuration submenu allows you to set the PCI configurations.



PCI/PNP ISA UMB Region Exclusion See submenu
PCI/PNP ISA IRQ Resource Exclusion See submenu
PCI/PNP ISA DMA Resource Exclusion See submenu

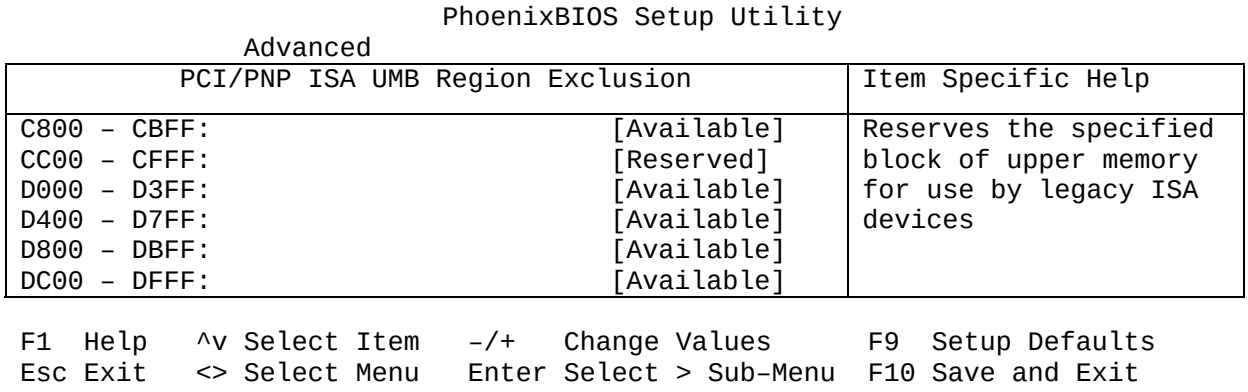
ISA graphics device installed: Yes, No. Enables ISA (non-VGA) graphics device to access palette

USB IRQ: Disabled, Auto Select, 3, 4, 5, 7, 9, 10, 11, 12, 14, 15. Specifies IRQ for use by USB. PCI cannot use an interrupt that is being used by an ISA or EISA device. Select Auto only if no ISA or EISA devices are on the system.

PCI IRQ Line 3: Disabled, Auto Select, 3, 4, 5, 7, 9, 10, 11, 12, 14, 15. Specifies IRQ for use by PCI. PCI cannot use an interrupt that is being used by an ISA or EISA device. Select Auto only if no ISA or EISA devices are on the system.

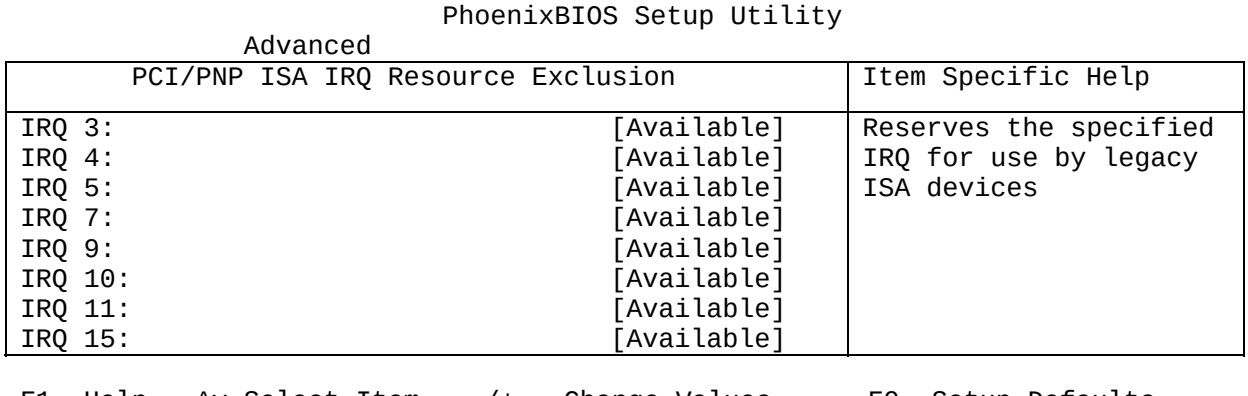
PCI/PNP ISA UMB Region Exclusion submenu

The PCI/PNP ISA UMB Region Exclusion submenu reserves the specified block of upper memory for use by legacy ISA devices. Options are Available or Reserved.



PCI/PNP ISA IRQ Resource Exclusion submenu

The PCI/PNP ISA IRQ Resource Exclusion submenu reserves the specified IRQ for use by legacy ISA devices. Options are Available or Reserved.



PCI/PNP ISA DMA Resource Exclusion submenu

The PCI/PNP ISA DMA Resource Exclusion submenu reserves the specified DMA channels for use by legacy ISA devices. Options are Available or Reserved.

PhoenixBIOS Setup Utility

Advanced

PCI/PNP ISA DMA Resource Exclusion		Item Specific Help
DMA 0:	[Available]	Reserves the specified DMA channel for use by non–Plug–and–Play ISA devices.
DMA 1:	[Available]	
DMA 2:	[Available]	
DMA 3:	[Available]	
DMA 5:	[Available]	
DMA 6:	[Available]	
DMA 7:	[Available]	

F1 Help ^v Select Item –/+ Change Values F9 Setup Defaults
Esc Exit <> Select Menu Enter Select > Sub–Menu F10 Save and Exit

Power menu

The Power menu allows you to set the power management configuration.

PhoenixBIOS Setup Utility

Main Advanced Power Boot Exit

		Item Specific Help
Power Savings: Standby Timeout: Auto Suspend Timeout: Video Timeout: IRQ 3: IRQ 4: IRQ 11: IRQ 12: Resume on Modem Ring:	[Disabled] Off Off Off [Enabled] [Enabled] [Enabled] [Enabled] [Off]	Maximum Power Savings conserves the greatest amount of system power. Maximum Performance conserves power but allows greatest system performance. To alter these settings, choose Customized. To turn off power management, choose Disabled.

F1 Help ^v Select Item -/+ Change Values F9 Setup Defaults
Esc Exit <> Select Menu Enter Select > Sub-Menu F10 Save and Exit

Power Savings: Disabled, Customize, Maximum Power Savings, Maximum Performance. Disabled disables all power management, Customize allows you to set parameters in the three menus below, the two Maximum settings use predefined values.

Standby Timeout: Off, 4, 8, 12, 16, 20, 24, 28 minutes. Inactivity period before system goes into Standby mode.

Auto Suspend Timeout: Off, 10, 20, 30, 40, 50, 60, 70 minutes. Inactivity period before system goes from Standby to Suspend mode.

Video Timeout: Off, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44 minutes. Inactivity period to power down monitor. Disabled turns CRT off in Standby mode.

IRQ 3/4/11/12: Disabled, Enabled. Enabling interrupt causes it to restore full On during Standby or Suspend.

Resume on Modem Ring: Modem ring serves as wakeup event

Boot menu

The Boot menu allows you to set the Boot configuration.

```
                    PhoenixBIOS Setup Utility
  Main     Advanced     Power     Boot     Exit
```

	Item Specific Help
Summary screen: [Disabled] Skip memory test [Yes] Floppy check: [Disabled] >Boot Order	Display system configuration on boot

```
F1  Help   ^v Select Item   -/+   Change Values      F9  Setup Defaults
Esc Exit   <> Select Menu   Enter Select > Sub-Menu  F10 Save and Exit
```

Summary screen:	Enables or disables summary screen during bootup
Skip memory test	Yes or No to skip memory test
Floppy check:	Enables or Disables search for floppy drives during bootup
Boot Order	Brings up Boot Order submenu, to set the order of drives to boot from.

Boot Order submenu

The Boot Order submenu allows you set the order of drives for booting.

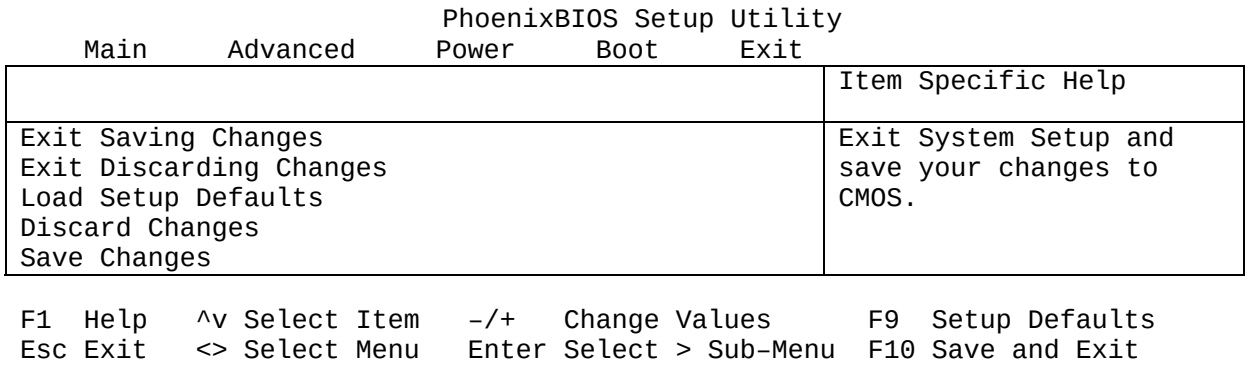
```
                    PhoenixBIOS Setup Utility
               Advanced
```

Boot Order	Item Specific Help
+Removable Devices +Hard Drive CD-ROM Drive	Order of Boot Devices <+> and <-> moves the device up or down.

```
F1  Help   ^v Select Item   -/+   Change Values      F9  Setup Defaults
Esc Exit   <> Select Menu   Enter Select > Sub-Menu  F10 Save and Exit
```

Exit menu

The Exit menu allows you to save or discard changes made during Setup. Esc does not exit this menu, you must select one of the menu items and press Enter. You can also press F9 or F10 at any time to exit Setup. When using the serial console F9 and F10 are not available; you must press down/up arrow to get to the proper option then press enter.



Chapter 4: *Save and run programs*

Save and run your programs on the 5070

Once you have written, tested and debugged your application, you can then save it to the SSD1 flash device, or to another device such as CompactFlash, or hard drive. As shipped from Octagon, SSD1 contains a bootable ROM-DOS. When you reboot the 5070, your program can automatically load and execute.

This chapter describes the following:

- Saving an application program to SSD1
- Autoexecuting the program from the 5070
- Overriding autoexecution of your program.

The information in this chapter assumes you are using ROM–DOS in your application.

Note Some Microsoft programs make undocumented DOS calls. With ROM–DOS, an error returns when an undocumented DOS call is made, causing your program to operate erratically. We recommend using Microsoft's MSDOS when using programs with undocumented DOS calls.

Saving programs and support files

By default, the drive in SSD1 comes preformatted from the factory, loaded with ROM-DOS, startup files, and an example demo program. To replace the demo program on SSD1 with your own, see the section *Adding your application*, in this chapter. To reformat SSD1, or to add your own operating system, please refer to the *SSD1, CompactFlash, SDRAM, battery backup, and Z–tag interface* chapter.

WARNING!

Reformatting SSD1 requires the use of a floppy or a hard drive to restore system files.

Adding your application

Three methods of copying your application to SSD1 are available. Do one of the following:

- From a local drive on the 5070 such as A: or C:, use the COPY command to copy your application to SSD1
- From a host PC using a terminal emulator, serially download your application program by using the TRANSFER command. Refer to the *SSD1, CompactFlash, SDRAM, battery backup, and Z–tag interface* chapter
- From a host PC, establish a remote drive and copy your application program from it, using the REMDISK and REMSERV commands. Refer to the *SSD1, CompactFlash, SDRAM, battery backup, and Z–tag interface* chapter.

1. Add or remove any device drivers for your application. CONFIG.SYS on SSD1 may require modification to include the device drivers.
2. To auto-execute your application, add your application name to the AUTOEXEC.BAT file. To replace the Octagon example program (DEMO.EXE) with your application, substitute DEMO in the AUTOEXEC.BAT file with your application program filename.

Autoexecuting your application from SSD1

SSD1 is the default boot device, drive C:. After adding your program information to SSD1, including any required CONFIG.SYS files and modifying AUTOEXEC.BAT to run your application file, reset the system. Your application should begin execution.

Overriding the auto-execution of your application

You may stop the auto-execution of your application by doing one of the following:

Option 1

1. Press F5 or F8, of press Ctrl–C on your local keyboard when the system is first starting. This halts all batch files.
2. Change AUTOEXEC.BAT and/or CONFIG.SYS to **not** call out your program.

Option 2

1. Install a floppy.
2. Change the BIOS Setup to enable the floppy drive and to boot from it. Refer to the *Setup programs* chapter.

3. Reset the system and boot from the floppy using a bootable disk.
4. Boot from floppy.
5. Change AUTOEXEC.BAT and/or CONFIG.SYS to **not** call out your program.

Chapter 5: Installing a different OS

The 5070 comes with ROMDOS 7.1 installed on SSD1. However, Octagon Systems has software development kits available for Windows CE.NET, Linux, and QNX. These kits directly support the unique features of Octagon products, such as digital I/O, watchdog timer, etc., eliminating the need to write special drivers. Other software kits may also be available. Contact Octagon Systems for information concerning the software development kits.

To install an operating system you will need:

- 5070 HDC-18-MPC-Multiport cable, #6240
- 5070 HDC-18-HDD/FDD drive cable, #6239
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor
- Floppy drive or CD-ROM drive, depending on the operating system media to be used
- Operating system media
- Hard drive or CompactFlash to install the operating system onto.

OS on floppy onto a hard drive or CompactFlash

Refer to Figure 5-1 for the following:

1. Attach the HDC-18-MPC-Multiport cable to J200, HDC-18-HDD/FDD drive cable to J1, and the VGA-12 video cable to J5.
2. Connect the PS-2 keyboard to J7 of the HDC-18-MPC-Multiport cable, a VGA monitor to the VGA-12 video cable, and a floppy drive to J3 of the HDC-18-HDD/FDD drive cable.
3. If using a hard drive, configure is as a master device and install it on J2A or J2B of the HDC-18-HDD/FDD drive cable.
4. If using a CompactFlash, configure is as a master device and install it into the CompactFlash socket.
5. Apply power to the 5070 system.
6. Enter Setup by pressing the F2 key during BIOS POST sequence

(this occurs between the memory test and bootup).

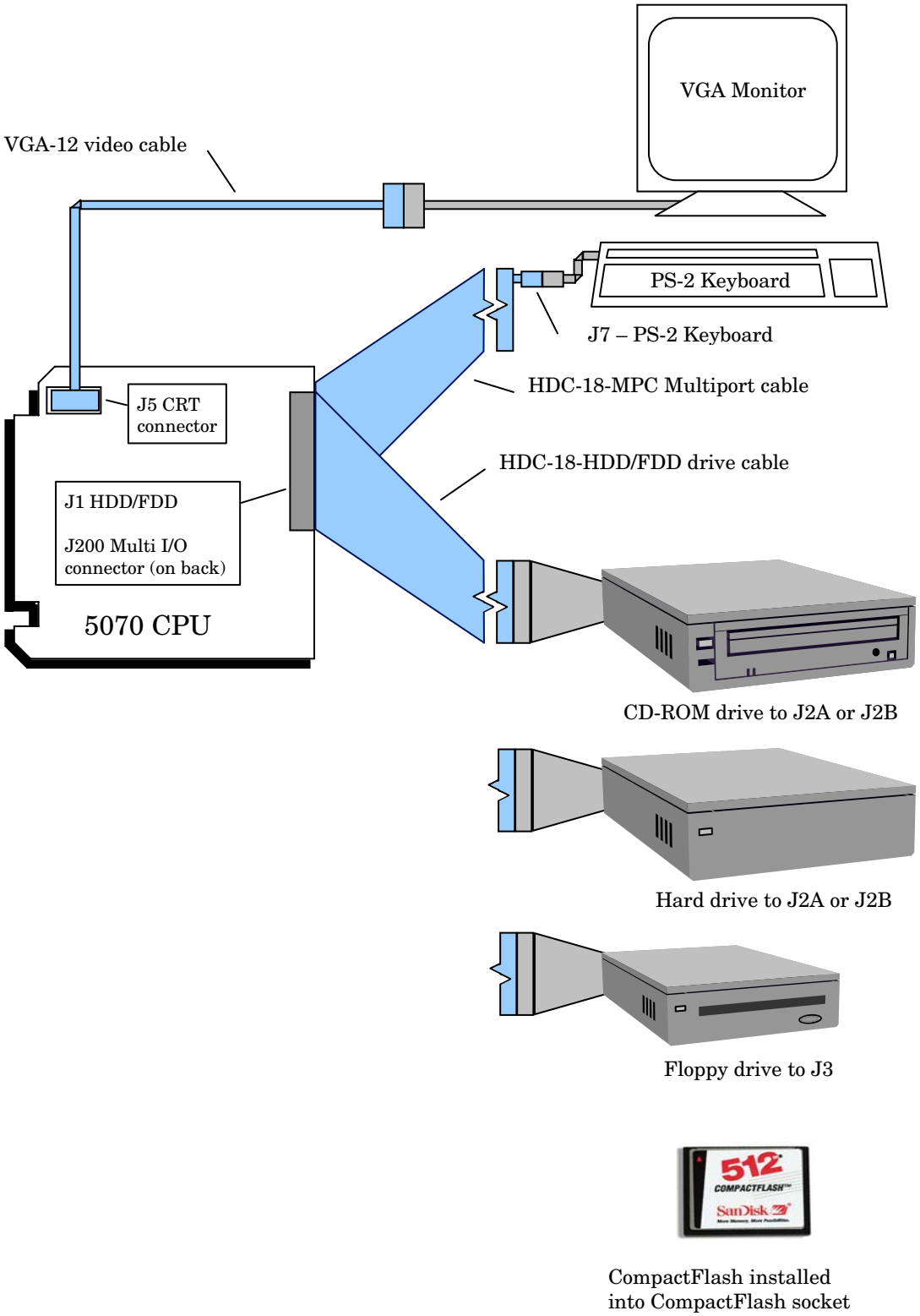
7. Configure the floppy drive as a device, and change the boot sequence to floppy drive first.
8. Insert the operating system media into the floppy drive.
9. Reboot the system.
10. The system should boot to the floppy drive.
11. Refer to the OS documentation to load the operating system.

OS on CD-ROM onto a hard drive or CompactFlash

Refer to Figure 5-1 for the following:

1. Attach the HDC-18-MPC-Multiport cable to J200, HDC-18-HDD/FDD drive cable to J1, and the VGA-12 video cable to J5.
2. Connect the PS-2 keyboard to J7 of the HDC-18-MPC-Multiport cable, a VGA monitor to the VGA-12 video cable, and a CD-ROM drive to J2A of the HDC-18-HDD/FDD drive cable. Configure the CD-ROM drive as a master.
3. If using a hard drive, configure is as a slave device and install it on J2B of the HDC-18-HDD/FDD drive cable.
4. If using a CompactFlash, configure is as a slave device (jumper W6[1–2]) and install it into the CompactFlash socket.
5. Apply power to the 5070 system.
6. Enter Setup by pressing the F2 key during BIOS POST sequence (this occurs between the memory test and bootup).
7. Configure the CD–ROM as a master device, and change the boot sequence to drive C: first.
8. Insert the operating system media into the CD-ROM drive.
9. Reboot the system.
10. The system should boot to the CD-ROM.
11. Follow the on-screen dialog to load the operating system.
12. Refer to the OS documentation for further information.

Figure 5–1 Installing a different operating system



Overview: Section 2 – Hardware

Section 2 discusses usage, functions, and system configurations of the 5070 major hardware features. The following chapters are included:

Chapter 6:	Serial ports
Chapter 7:	LPT1 parallel port
Chapter 8:	Console devices
Chapter 9:	SSD1, CompactFlash, SDRAM, battery backup, and Z–tag interface
Chapter 10:	External drives
Chapter 11:	Digital I/O
Chapter 12:	CRTs and flat panels
Chapter 13:	Ethernet
Chapter 14:	PC/104 expansion
Chapter 15:	USB

Chapter 6: Serial ports

Description

The 5070 has two serial ports, COM1 and COM2, which are accessed through the multipurpose connector, J200. These serial ports interface to a printer, terminal, or other serial device. All ports support 5-, 6-, 7-, or 8-bit word lengths, 1, 1.5, or 2 stop bits, and baud rates up to 115.2K.

COM1 and COM2 are 8 wire interfaces and can be configured as RS–232, RS–422, or RS–485 interfaces.

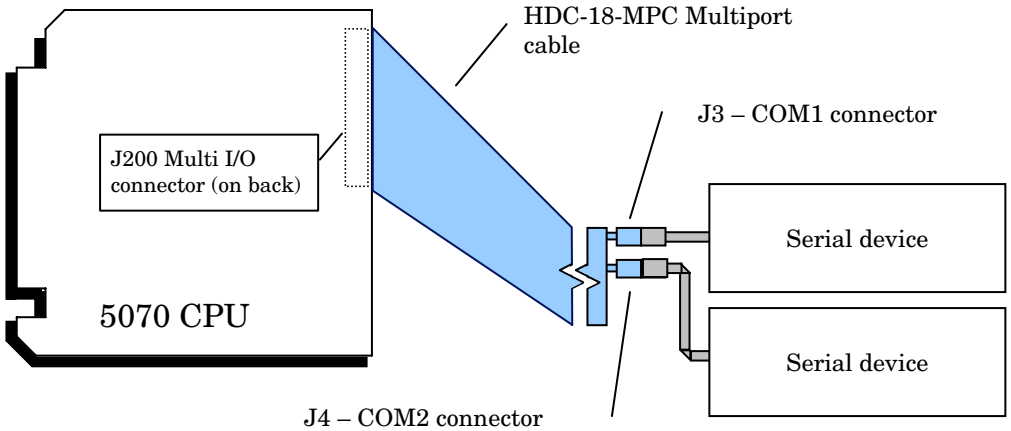
Both serial ports have the following specifications:

- 16550 compatible
- 16–byte FIFO buffers
- IEC 1000, level 3, ESD protection
  — Contact discharge ±4 kV
  — Air–gap discharge ±8 kV
- Backdrive protection
- Up to 115.2k Baud operation

Mating receptacle

Use a 5070 HDC-18-MPC-Multiport cable to connect the COM ports to external serial equipment. The J3 and J4 connectors of the 5070 HDC-18-MPC-Multiport cable are DB–9 female connectors that plug directly into a 9–pin PC serial cable. See pages 27 and 28 for information on the 5070 HDC-18-MPC-Multiport cable.

Figure 6–1 COM ports



Serial port configurations

The COM ports are defined in Table 6–1. Table 6–2 shows the jumper settings, and Table 6–3 shows the COM pin-outs for J200.

Table 6–1 Serial port configurations

COM Port	Address	IRQ	Interface	Connector (5070 HDC-18-MPC-Multiport cable)
COM1	3F8h*, 2F8h, 3E8h, 2E8h	IRQ4*, IRQ3	RS–232 – 8 wire RS–422 – 4 wire RS–485 – 2 wire	J3 – COM1
COM2	2F8h*, 3F8h, 3E8h, 2E8h	IRQ3*, IRQ4	RS–232 – 8 wire RS–422 – 4 wire RS–485 – 2 wire	J4 – COM2

* = default

Table 6–2 COM1 and COM2 connector pin-outs

COM1 - J3 (pin numbers shown for J200)				COM2 - J4 (pin numbers shown for J200)			
Pin#	**RS–232 signal**	**RS–422 signal**	**RS–485 signal**	**Pin#**	**RS–232 signal**	**RS–422 signal**	**RS–485 signal**
2	DCD	Tx+	DATA+	20	DCD	Tx+	DATA+
4	DSR	Tx–	DATA-	22	DSR	Tx–	DATA-
6	RxD			24	RxD		
8	RTS			26	RTS		
10	TxD			28	TxD		
12	CTS			30	CTS		
14	DTR	Rx+		32	DTR	Rx+	
16	RI	Rx–		34	RI	Rx–	
18	GND	GND	GND	36	GND	GND	GND

Table 6–3 5070 COM port jumper settings: W3, W8, W10, W11

W3, W8, W10, W11 – COM Ports		
COM Port	**Communication Mode**	**Jumper Settings**
COM1	RS–232C*	W3[4–6][10–12]* W10[1–2]* W11[1–2][4–6][5–7]*
	RS–422 no termination	W3[4–6][10–12] W10[1–3] W11[1–2][4–6][5–7]
	RS–422 with termination	W3[2–4][8–10] W10[1–3] W11[1–2][4–6][5–7]
	RS–485 no termination	W3[4–6][10–12] W10[2–4] W11[1–2][4–6][5–7]
	RS–485 with termination	W3[4–6][10–12] W10[2–4] W11[1–3][7–9][8–10]
COM2	RS–232C*	W3[3–5][9–11]* W8[1–2][4–6][5–7]* W10[7–8]*
	RS–422 no termination	W3[3–5][9–11] W8[1–2][4–6][5–7] W10[7–9]
	RS–422 with termination	W3[1-3][7-9] W8[1–2][4–6][5–7] W10[7–9]
	RS–485 no termination	W3[3–5][9–11] W8[1–2][4–6][5–7] W10[8-10]
	RS–485 with termination	W3[3–5][9–11] W8[1–3][7-9][8–10] W10[8-10]

* = default jumper installed

Function and use of serial ports

COM1 as serial console device

You can use COM1 as a console device to communicate with another PC. For COM1 to be a serial console, the “V” video jumper W12[5-6] must be removed. See the *Console devices* chapter for more information.

Mating receptacle

Use a 5070 HDC-18-MPC-Multiport cable to connect the COM ports to external serial equipment. The J3 and J4 connectors are DB–9 female connectors that plug directly into a 9–pin PC serial cable.

Note When interfacing the 5070 to your desktop PC, you must use a null modem adapter.

Note See pages 27 and 28, and *Appendix A: Mating connectors* for mating information on the multipurpose connector.

COM ports as RS–232 I/O

COM1 and COM2 are 8–wire RS–232 interfaces. You can connect two serial I/O devices.

In the default configuration, the video jumper W12[5-6] is installed. This jumper automatically disables the Serial Video option in the Advance menu in Setup, and both COM ports are available for serial I/O devices. In some instances, such as running a program on the 5070 that will ultimately be used on another card without on-board video, you might want to remove the video jumper and still use COM1 as a COM port instead of a serial console. In this instance, you must go into Setup and set Serial Video in the Advanced menu to Disabled.

RS–422

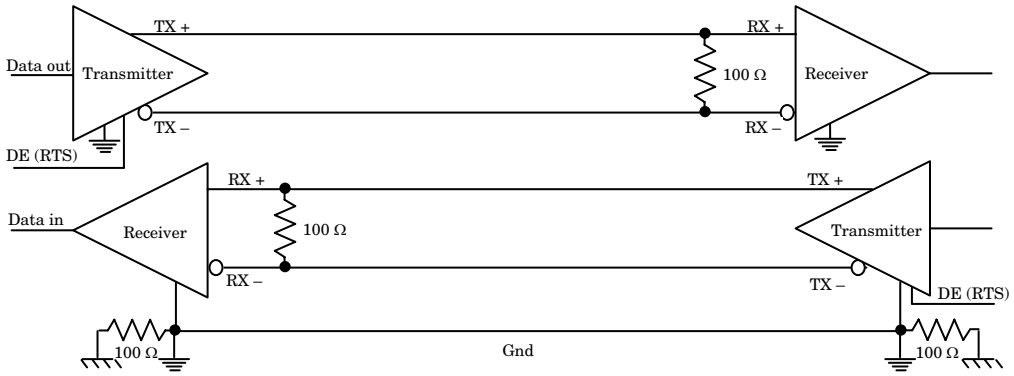
COM1 and COM2 can be used as RS–422 ports. RS–422 is typically a point–to–point configuration using differential signaling to communicate between the devices on a network. Differential signal reduces the effect of environmental noise, allowing communication over distances up to 1200 meters. The 5070 uses RTS internally to enable the transmit function.

RS–422 is also specified for multi–drop (party–line) applications where only one driver is connected to, and transmits on, a “bus” of up to 10 receivers.

The device at the end of an RS–422 network must be terminated. The 5070 optionally terminates with a 100 ohm resistor. Refer to Table 6–4. Figure 6–5 shows a typical RS–422 four wire interface circuit. Figure 6–6 shows a typical connection.

Note The 5070 RS-422/RS-485 circuitry is configured as space condition and must be configured externally if mark condition is required.

Figure 6–2 Typical RS–422 four–wire interface circuit



RS–485

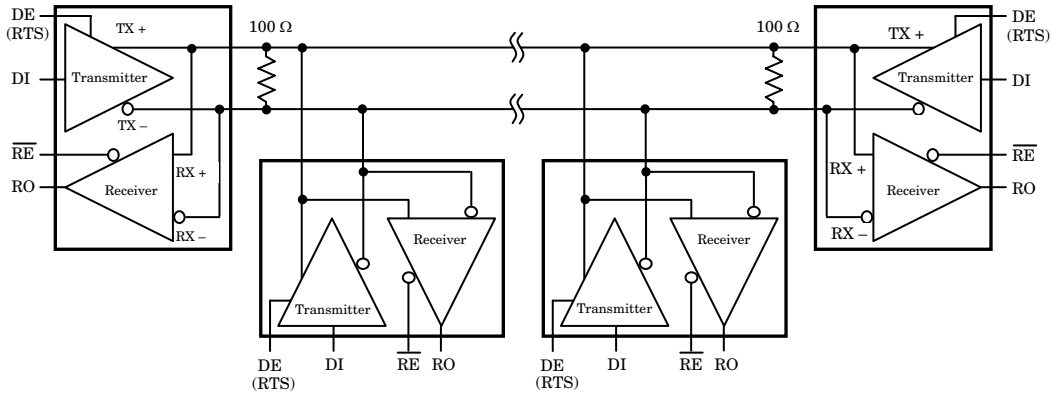
An application may implement a node as either the “host” node or as a “remote” node in an RS–485 network. There can be as many as 32 nodes without any bus repeaters in the network. A host is referred to as the node that initiates communication; while a remote is referred to as a node that is addressed by the host.

In any given communication sequence in an RS–485 network, there can only be one host. The host is responsible for initiating communication, maintaining network registration, and providing housekeeping tasks with other nodes. Remotes, however, cannot initiate a communication. They can only respond to messages that are addressed to them from the host. The 5070 uses RTS internally to enable the transmit function.

The devices at each end of an RS–485 network must be terminated. Any node located between the end points should not be terminated. The 5070 optionally terminates with a 100 ohm resistor. Refer to Table 6–4. Figure 6–7 shows a typical RS–485 network.

Note The 5070 RS-422/RS-485 circuitry is configured as space condition and must be configured externally if mark condition is required.

Figure 6–3 Typical RS–485 half duplex interface circuit



Chapter 7: LPT1 parallel port, LCD and keypad

LPT1 parallel port

LPT1 is a multifunction parallel port, which is accessed through the multipurpose connector, J200. It supports the unidirectional standard mode, bi-directional mode, enhanced parallel port (EPP) mode, and extended capabilities port (ECP) mode. The default I/O address for LPT1 is 378h, with the default interrupt is IRQ7. You can choose the addresses 278h or 3BCh, or interrupt IRQ5, in the 5070 Setup utility.

The LPT1 port supports a number of devices including a PC compatible printer, an LCD display, or a keypad.

Mating receptacle

Use a 5070 HDC-18-MPC-Multiport cable to connect the LPT port to external parallel equipment. The P3 connector is a DB–25 female connector which plugs directly into a 25–pin parallel cable.

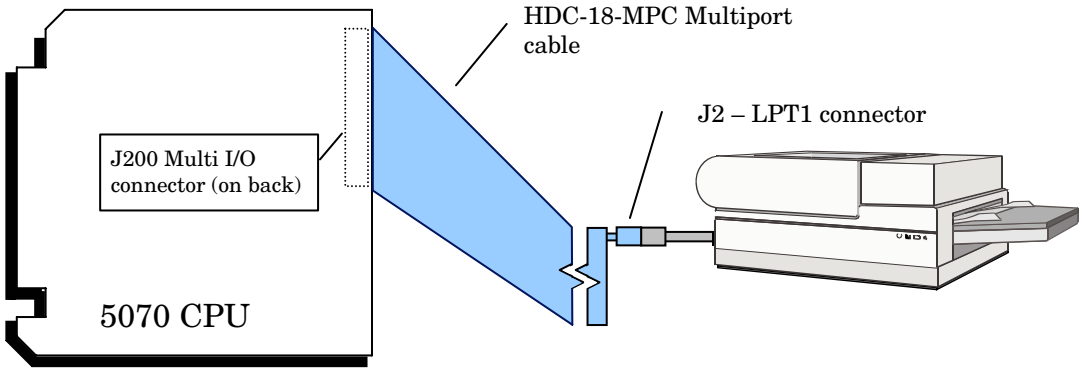
Note See pages 27 and 28, and *Appendix A: Mating connectors* for mating information on the multipurpose connector.

Printer

Installing a printer

1. Make sure that the LPT1 port is in standard or bi-directional mode. This is done in Setup.
2. Connect the 5070 HDC-18-MPC-Multiport cable to the 5070 card.
3. Connect the DB–25 of the 5070 HDC-18-MPC-Multiport cable to the printer cable.

Figure 7–1 LPT1 as a printer port



LCD display

The LPT1 port supports either a 4 x 20 or a 4 x 40 liquid crystal display (LCD). To interface the displays to the 5070, a 5070 HDC-18-MPC-Multiport interface cable and an Octagon 2010 interface board are required.

The program DISPLAY.EXE in the Utilities zip file (see page 139) provides an easy method to use the display. Refer to the file DISPLAY.TXT for information on initializing and using the display. Also, refer to the *2010 product sheet* for more information on the interface board.

Installing an LCD display

1. Connect J2B (26–pin connector) of the 5070 HDC-18-MPC-Multiport cable to J3 on the 2010. Refer to Figure 7–3.
2. Connect the display cable to either the 14–pin or 16–pin header on the 2010. The size of the display will determine which header to use.
3. Refer to the file DISPLAY.TXT for more information on initializing and using the display.

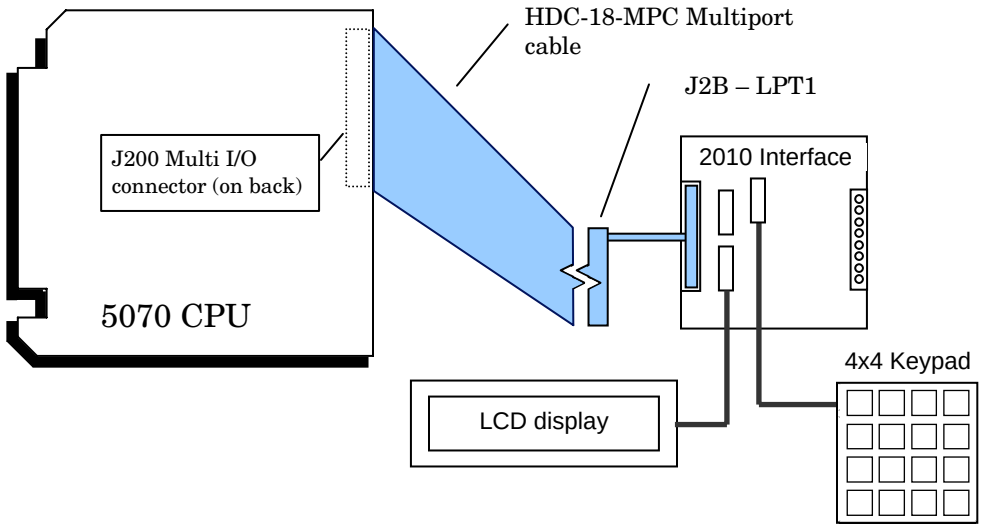
Keypad

LPT1 also supports 4 x 4 matrix keypads. To interface the keypad to the 5070, use the Octagon 2010 interface board. The program DISPLAY.EXE in the Utilities zip file (see page 139) provides an easy method to use the keypad. Refer to the file DISPLAY.TXT on the CD-ROM for information on initializing and using the keypad. Also, refer to the *2010 product sheet* for information on the interface board.

Installing a keypad

1. Connect J2B (26–pin connector) on the 5070 HDC-18-MPC-Multiport cable to J3 on the 2010. Refer to Figure 7–3.
2. Connect the keypad cable to the 10–pin header on the 2010.
3. Refer to the DISPLAY.TXT file for more information on reading the keypad.

Figure 7–2 LPT1 as a display or keypad port



Chapter 8: Console devices

Description

The 5070 has three options for console devices. You can use a monitor and a keyboard as your console. You can use COM1 as the console, or you can run the system without a console device.

Selecting console devices

The following represent the options on the 5070 for console devices:

- A standard VGA monitor and a keyboard.
- Serial console from COM1. A serial cable/null modem adapter plugged into a host PC running HyperTerminal (or equivalent) provides both input and output. The local keyboard also allows input but is not required.
- No console device means no video output, either from a monitor or the serial console. A local keyboard allows input but is not required.

Monitor and keyboard console

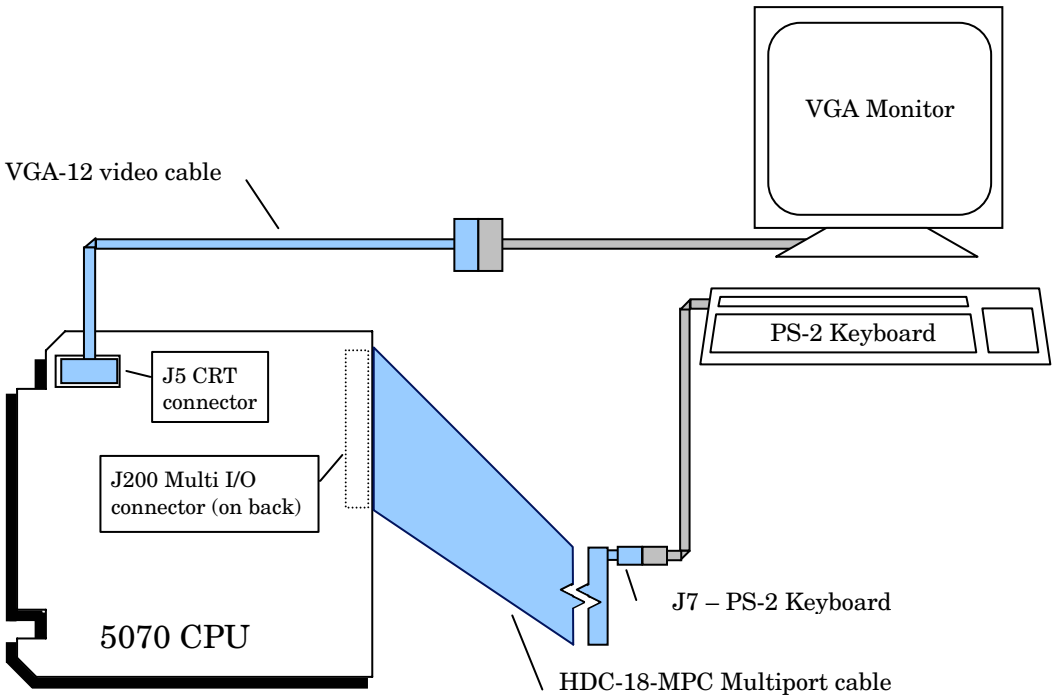
To use a monitor and keyboard as the console, you will need the following equipment (or equivalent):

- 5070 CPU card
- 5070 HDC-18-MPC-Multiport cable, #6240
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor

1. Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the 5070.
2. Make sure that jumper the “V” video jumper, W12[5-6], is installed.
3. Connect the VGA-12 video cable into J5.
4. Connect the 5070 HDC-18-MPC-Multiport cable into J200.

5. Connect a VGA monitor to the VGA-12 cable, and a PS-2 style keyboard to J7 of the HDC-18-MPC-Multiport cable.

Figure 8–1 Monitor and keyboard as console



Serial console

COM1 is used as the console device if the serial console is enabled.

To use COM1 as the console, you will need the following equipment (or equivalent):

- 5070 CPU card
- 5070 HDC-18-MPC-Multiport cable, #6240
- Null modem adapter, #2470 (9-pin to 9-pin)
- Host computer running HyperTerminal (or equivalent)
- Serial cable to connect 5070 COM1 to host computer serial port
- PS-2 style keyboard (optional)

1. Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the 5070.

2. Remove the video jumper, W12[5-6].
3. Connect the 5070 HDC-18-MPC-Multiport cable into J200.
4. Connect the null modem adapter to J3 of the HDC-18-MPC-Multiport cable.
5. Connect the serial cable between the null modem adapter and the serial port of the host computer.

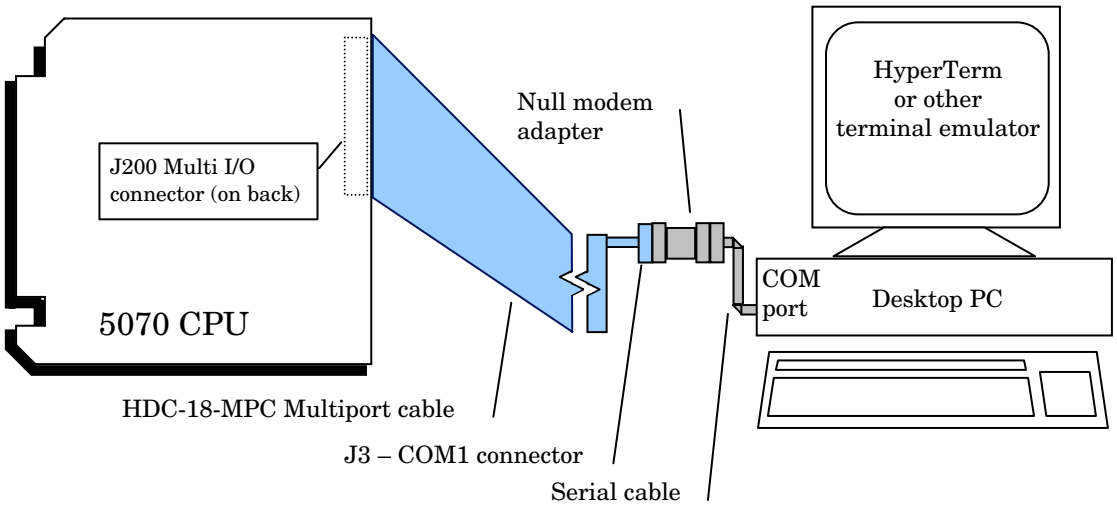
Follow these steps to use the serial console:

6. For communication using HyperTerminal (or equivalent), the following settings must be used:

Connect using:	**Direct to COM1, COM2, COM3, or COM4** (select the port the serial cable is connected to)
Baud rate:	**38400**
Communications parameters:	**no parity, 8 data bits, 1 stop bit**
Flow control:	**none**
Terminal support:	**ANSI**
ANSI terminal option– Wrap lines that exceed terminal width:	**Yes**

7. Start HyperTerminal. You are now ready to establish communications between the host PC and the 5070.
8. Power on the 5070.
9. If you do not get the proper logon message check the HyperTerminal serial parameters of the host PC to make sure they match the settings in step 6.

Figure 8–2 The 5070 and a serial console



Chapter 9: SSD1, CompactFlash, SDRAM, battery backup, and Z–tag interface

Description

The 5070 is shipped with a 2 MB SMT Flash for SSD1 for storing programs and data, a CompactFlash socket for storing programs and data, and 32 MB of surface mounted SDRAM for program execution. A battery backup connector is provided via the HDC-18-MPC-Multiport cable to back-up the real time clock, and a Z-tag interface is provided to allow updating of the BIOS.

SSD1

SSD1 is a 2 MB SMT flash soldered directly onto the PCB board. It contains the BIOS drive. SSD1 can be used as a hard drive to store programs and data.

CompactFlash

A CompactFlash device is installed into J201 and appears to the system as an IDE device for storing programs and data. It is automatically detected and configured as a hard drive during bootup. To configure the 5070 to boot from a CompactFlash, refer to the following section “Creating a Bootable CompactFlash.”

The CompactFlash can be configured as a master or slave using jumper W6. It can also be configured for 3V or 5V operation using jumper W5. Table 9–1 shows the jumper settings.

Refer to Figures 2-1 and 2-2 on pages 21 and 22 for the location of various connectors and jumpers before installing the CompactFlash.

Note Octagon Systems only recommends Industrial Grade CompactFlash (NAND technology) that implements ECC error code correction, and wear level technology.

Table 9–1 CompactFlash configuration jumpers: W5, W6

W5, W6 – CompactFlash	
Configuration	**Jumper**
Master	W6[1–3]
Slave	W6[1–2]*
5V	W5[1–2]*
3V	W5[3–4]

* = default

Creating a bootable CompactFlash

A CompactFlash as shipped from the factory may or may not be formatted; even if formatted, it may or may not be bootable. The following sequence shows how to create a bootable CompactFlash, and how to configure the 5070 to boot from the CompactFlash. The CompactFlash can be formatted, partitioned, and sys'ed from SSD1, or from an external drive such as a hard drive, floppy, or CD.

1. Create a bootable external device such as a hard drive, floppy, or CD.

Note Octagon offers OS Embedder kits that include a CD boot disk for a variety of operating systems. Contact your Octagon representative for additional information.

2. Change the boot sequence in Setup so the 5070 boots from the external drive first. If booting from SSD1, issue the command *SETSSD1 /before*.
3. Use FDISK to create partitions on the CompactFlash. Refer to your operating system manual for the appropriate parameters for using FDISK. You might also have to refresh the MBR (Master Boot Record). For ROMDOS, the command for refresh is `fdisk 80/r`.
4. Reboot, using the external device.
5. Format the CompactFlash.
6. Copy your operating system from the external device to the CompactFlash.
7. Change the boot sequence in Setup so that the CompactFlash (hard drive) is first.
8. Issue the command *SETSSD1 /after*.
9. Power off the 5070 and remove the external device.
10. Ensure the CompactFlash is configured as a Master with jumper W6[1–3].
11. Power on the 5070 card. The card should now boot from the CompactFlash.

SDRAM

The 5070 comes with 32 MB of surface mount SDRAM.

Battery backup for real time calendar clock

An AT battery can be installed to back up the CMOS real time clock.

Installing an AT battery

1. Power off the 5070.
2. Install the 3.6V AT clock battery at the J9 connector of the 5070 HDC-18-MPC-Multiport cable.

Note See *Appendix A: Mating connectors* for mating information on the battery connector.

Z–tag interface

The Z–tag interface allows you to reload the BIOS should it become corrupted. This procedure requires a Z–tag dongle, available from ZF Microdevices. Refer to \z-tag_mgr\readme.htm in the Utilities zip file and the ZFx86 Data book at www.zfmicro.com.

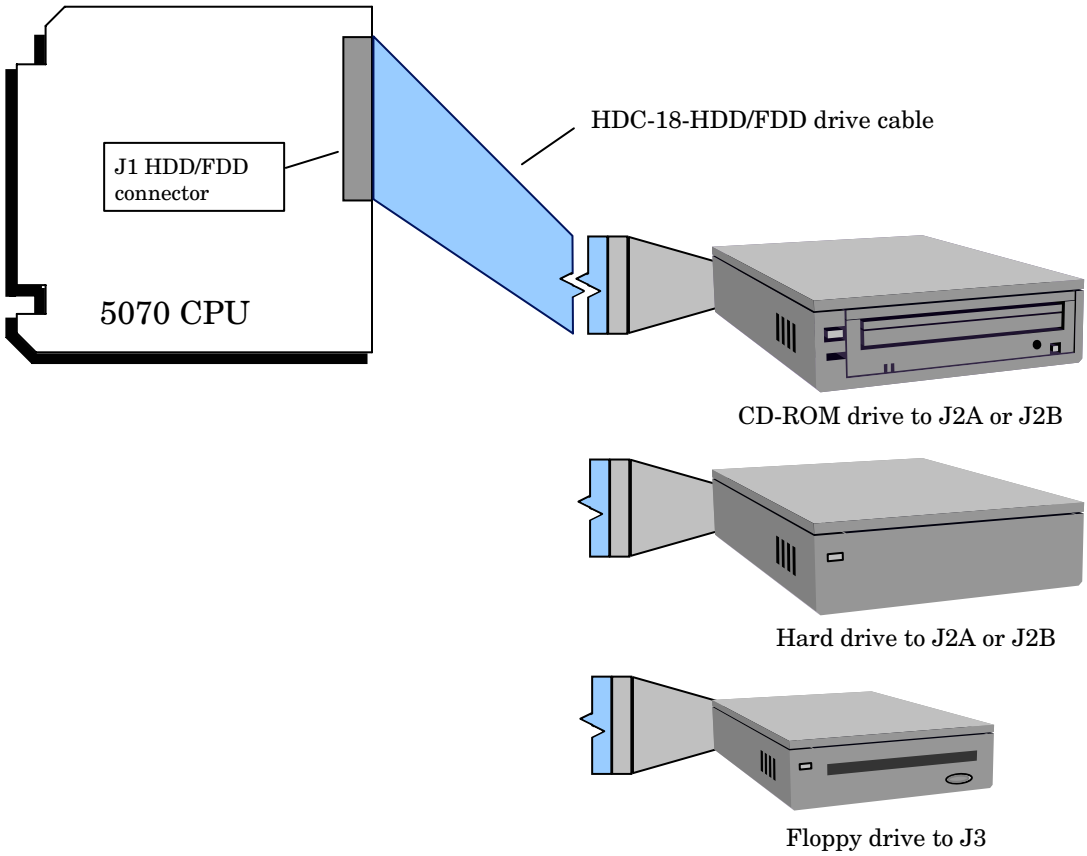
Chapter 10: External drives

Description

The 5070 is compatible with any standard floppy drive, and any standard IDE hard drive that has a 16–bit IDE interface. This includes CD-ROMs, CompactFlashes, and other IDE-compatible drives. The BIOS extension ROM for the hard drive is supplied on the card so that no additional software is needed.

Note The BIOS supports two IDE devices and two floppy drives, however, the HDC-18-HDD/FDD drive cable only has one connector for the floppy drive.

Figure 10–1 5070 with floppy/IDE device

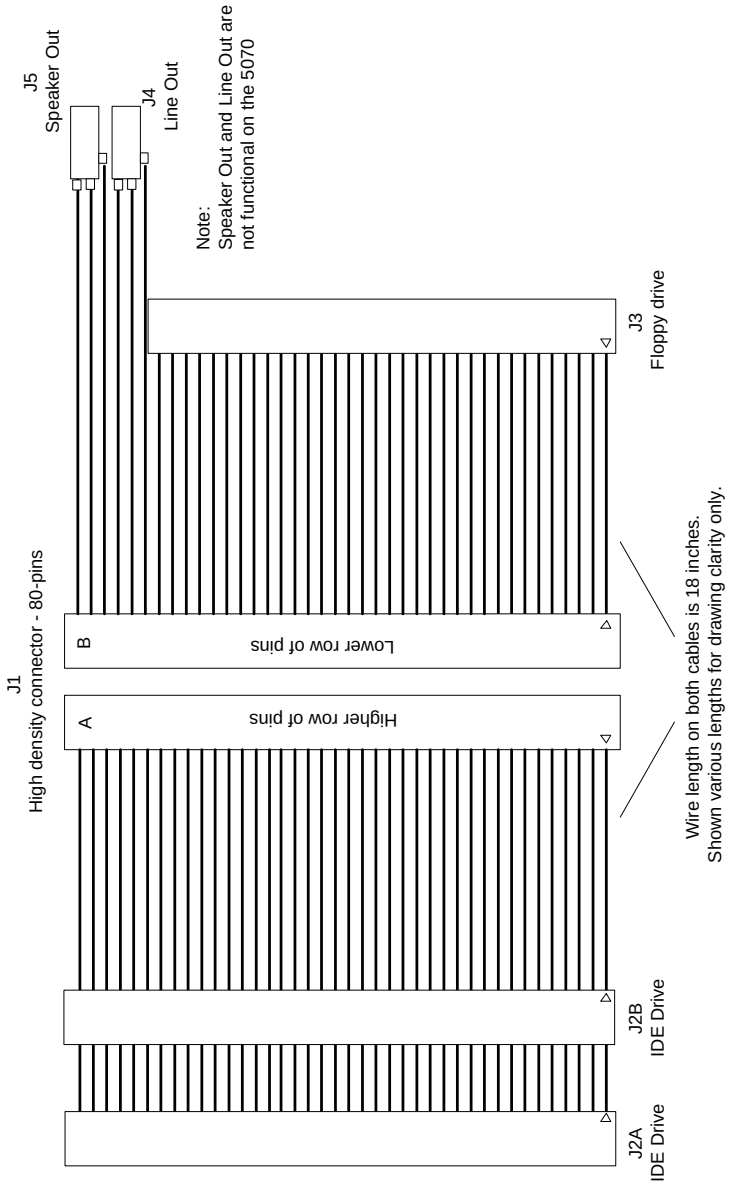


HDC-18-HDD/FDD drive cable

The floppy drive and hard drive pins are located on an 80-pin connector located at J1 on the front side of the 5070 card. The HDC-18-HDD/FDD drive cable breaks out these pins into standard 40–pin, 2.54–mm IDE connectors and a standard 34–pin, 2.54–mm floppy drive connector. Note that the IDE connector does not supply +5V to the hard drive.

Note The HDC-18-HDD/FDD drive cable is used with other Octagon products, and contains two additional connectors for speaker jacks. These connectors are not used on the 5070. Figure 10-1 shows the HDC-18-HDD/FDD drive cable.

Figure 10–2 HDC-18-HDD/FDD drive cable



Floppy disk controller

The 5070 can interface directly to one 3.5 in. or 5.25 in. floppy drive via the HDC-18-HDD/FDD drive cable and the connector at J1. The BIOS will support two floppy disk drives.

Note See *Appendix A: Mating connectors* for mating information on the floppy disk connector.

Power requirements

You supply power to the floppy drive through an external source. Refer to your floppy drive manual for specific instructions.

Installing a floppy disk drive

1. Disconnect power to the 5070.
2. Connect the HDC-18-HDD/FDD drive cable to J1 on the 5070 (see page 21 for location of J1, and see page 75 for an illustration of the HDC-18-HDD/FDD drive cable).
3. Insert the 34–pin connector on the HDC-18-HDD/FDD drive cable into the rear of the floppy drive. Make sure pin 1 on the cable is connected to pin 1 on the drive.
4. Connect power to the floppy drive.
5. Power on the 5070. Enter Setup to set up the BIOS. You can execute this program by pressing "F2" during system bootup. The system steps you through the configuration. Also, refer to the *Setup programs* chapter for more information on the BIOS Setup program.

Hard disk controller

The 5070 supports two 16–bit IDE devices. Since the CompactFlash is seen by the system as an IDE hard drive, only one additional IDE drive can be installed if the CompactFlash is installed.

Standard IDE devices such as hard drives and CD-ROM drives are interfaced via 40–pin connectors on the HDC-18-HDD/FDD drive cable installed at J1.

IDE combinations:

- 2 hard drives
- 1 hard drive and 1 CD-ROM drive

- CompactFlash and either 1 hard drive or 1 CD-ROM drive

Note Power is NOT supplied to the IDE device through the HDC-18-HDD/FDD drive cable.

Installing a hard drive

1. Disconnect power to the 5070.
2. Connect the HDC-18-HDD/FDD drive cable to J1 on the 5070 (see page 21 for location of J1, and see page 75 for an illustration of the HDC-18-HDD/FDD drive cable).
3. Insert the 40–pin connector on the HDC-18-HDD/FDD drive cable into the interface connector of the hard drive. Make sure pin 1 on the cable is connected to pin 1 on the drive.
4. Connect power to the hard drive.
5. Execute the BIOS Setup program to configure your system for a hard drive. You can execute this program by pressing “F2” during system bootup. The system steps you through the configuration. Also, refer to the *Setup programs* chapter for more information on the BIOS Setup program.
6. If you want to boot the system from the hard drive, you need to format the drive accordingly, and change the boot order in Setup.

Chapter 11: Digital I/O

Description

The 4–bit digital I/O port provides two electrically isolated input lines and two electrically isolated output lines. These lines will interface with logic devices, switch inputs, LEDs and industry standard opto module racks. The HDC-18-MPC-Multiport cable, connector J11, brings out the digital I/O signals.

Figures 11-1 and 11-2 show diagrams using the digital I/O on the 5070 card. Table 11-1 shows the pin-out for the connector.

Figure 11–1 Digital I/O diagram

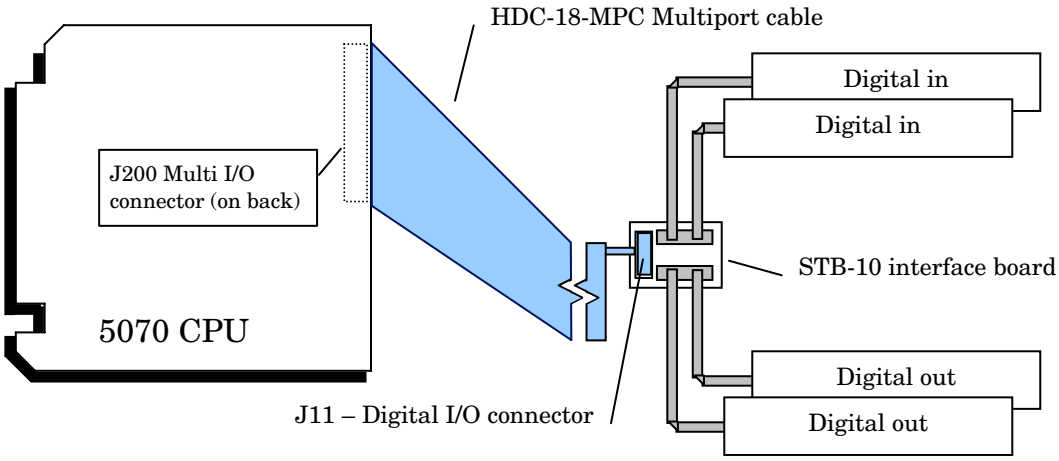


Table 11–1 Digital I/O connector: J11

J11 – Digital I/O connector		
Function	**Pins**	**Name**
Input 0 (GPIO0)	1	DIN0 –
	2	DIN0 +
Input 1 (GPIO1)	3	DIN1 –
	4	DIN1 +
Output 0 (GPIO2)	5	DOUT0 –
	6	DOUT0 +
Output 1 (GPIO3)	7	DOUT1 –
	8	DOUT1 +
	9	Not used
	10	Not used

Figure 11–2 Digital I/O example

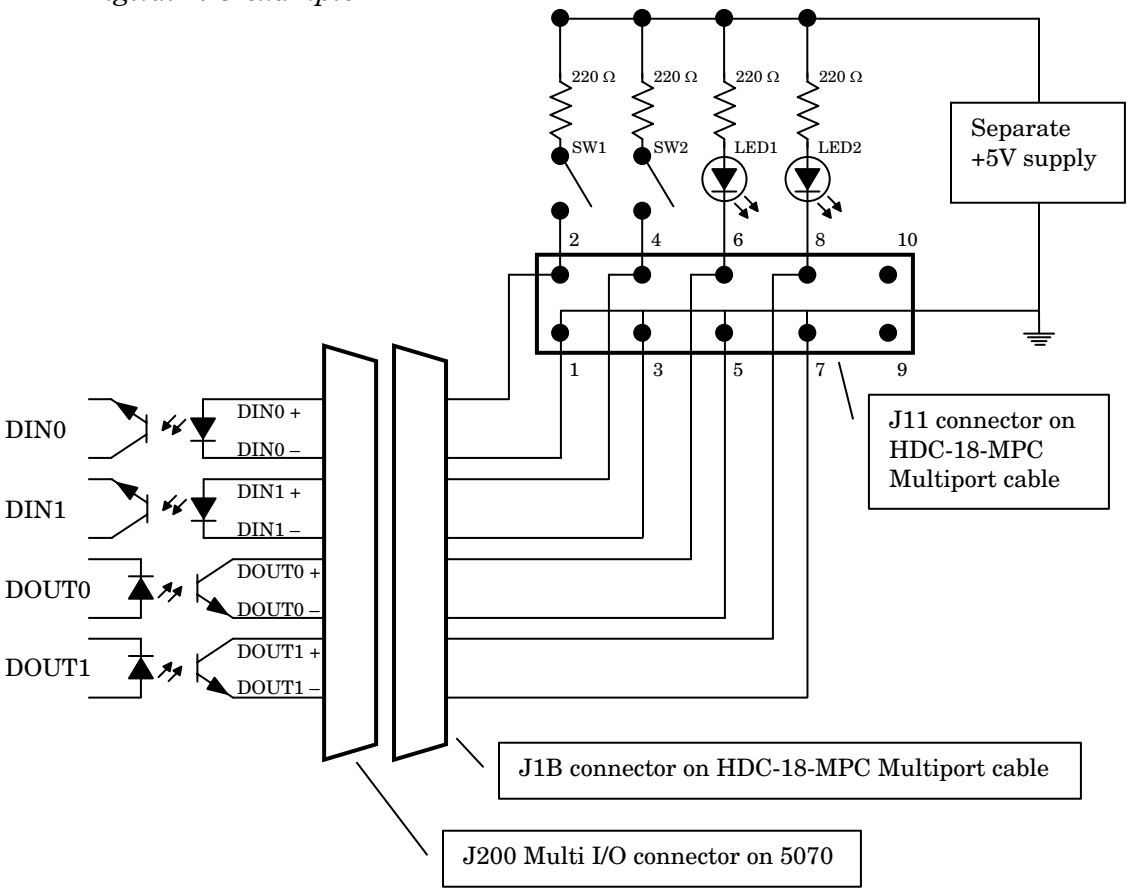


Table 11–2 Digital I/O – optically coupled isolator ratings

Optically coupled isolator ratings *		
Isolation	Input-to-output	2500Vdc
	Channel to channel	500Vdc
Input diode	Forward DC current	60mA
	Reverse DC voltage	3V
	Peak forward current	3A (1us pulse, 300pps)
	Power dissipation	100mW
	Derate linearly	1.33mW/°C above 25° C
Output transistor	Collector current	30mA
	Power dissipation	150mW
	Derate linearly	2.00mW/°C above 25° C

* Absolute maximum ratings

WARNING!

Do not exceed the absolute maximum ratings. Damage to the 5070 card may occur.

INT17 calls for digital I/O

This section provides definitions for the functions using the INT17 handler, I17HNDLR.EXE.

I17HNDLR.EXE is a TSR program and is called out by the 5070 BIOS. By default, when the “X” jumper is on, the INT17 extended BIOS is operational. If the “X” jumper is removed and DOS is the operating system, the I17HNDLR.EXE TSR can be used. Once executed, the TSR is active, but it must be executed each time the system is rebooted. Copy the I17HNDLR.EXE utility to your boot device and add it to your AUTOEXEC.BAT.

Note The INT17 functions can only be used with DOS operating systems. If you use a different operating system, the INT17 functionality can still be used by your application but must be integrated into your software.

GPIO register

The GPIO register is used to control the digital inputs and outputs, the user jumper, and the CR3 and CR5 system activity LEDs.

The current GPIO input state is at 8104h. The GPIO output is at 8100h. Bit0 is GPIO0, bit 1 is GPIO1, etc.

IMPORTANT

To assure that output bits are not inadvertently changed, always use the read-modify-write method to change this register.

Table 11–3 GPIO register

GPIO register								
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Name	CR5 green LEDB	CR3 amber LEDA	Reserved	User jumper	GPIO3	GPIO2	GPIO1	GPIO0
Function	Status LED	Status LED		User defined	Digital output	Digital output	Digital input	Digital input
Direction	Output	Output		Input	Output	Output	Input	Input

Write digital output

```
Function:            0ech
Subfunction:         01h
Purpose:             To write a value to the GPIO outputs at GPIO[2]
                     and GPIO[3].
Calling Registers:   AH     0ech
                     AL     01h
                     DI     mmxxmmxx ddxxddxx
                            The m bits are mask bits. If set to 1, the
                            corresponding d bit will be written. If
                            set to 0, the corresponding d bit will
                            remain at current value.
                                          DI mask bit   DI data bit
                                   GPIO[7]   15            7
                                   GPIO[6]   16            6
                                   GPIO[3]   11            3
                                   GPIO[2]   10            2
                     DX     ffffh
Return Registers:    Carry flag cleared if successful
                     Carry flag set if error
                     AL     Error code
Comments:            This function is used to write to GPIO output
                     pins.
Programming example 1:
                     /* Inline assembly code for Borland C++ 3.1*/
                     asm {
                            mov ax,0ec01h
                            /* Set GPIO[0] to one and set GPIO[1] to
                            zero */
                            mov di,0c0ch
                            mov dx,0ffffh
                            int 17h
                             }
```

Read digital I/O input

```
Function:            0cfh
Subfunction:         02h
Purpose:             To read from the GPIO[0], [1], [2], or [3] pins
Calling Registers:   AH     0ech
                     AL     02h
                     DX     ffffh
Return Registers:    Carry flag cleared if successful
                     AL     xxxxdddd dddd is the state of GPIO[0-3]
                     Carry flag set if error
                     AL     Error code
Comments:            This function is used to read from the GPIO[2]
                     and GPIO[3] pins.
Programming example 1:
                     /* Inline assembly code for Borland C++ 3.1*/
                     unsigned char aData, bData, cData;
                     asm {
                            mov ax,0ec02h
```

```
mov dx,0ffffh
int 17h
mov aData,al
 }
```

Chapter 12: CRTs and flat panels

Description

The video system on the 5070 uses the advanced 69000 video controller from Asilant (formerly Chips & Technologies). It supports VGA, SVGA, and XVGA monitors, and LCD and EL flat panel displays with resolutions to 1024 x 768 x 16 bpp (1280 x 1024 on some selected displays). The 69000 is also a graphics accelerator. Since the video circuitry operates on the PCI bus at the full PCI bus speed, programs like Windows execute very rapidly. The video section has 2 MB of video SDRAM for high–resolution displays and simultaneous CRT and flat panel operation. The 5070 supports 3V and 5V flat panel displays.

Standard VGA monitors with analog inputs are connected using a VGA–12 cable (p/n 4865) connected to J5. Flat panel displays are connected using a 50–pin 2mm connector at J4. Refer to the Utilities zip file (see page 139) for additional information on flat panel displays.

Video features

Below is a list of standard video features installed on the 5070:

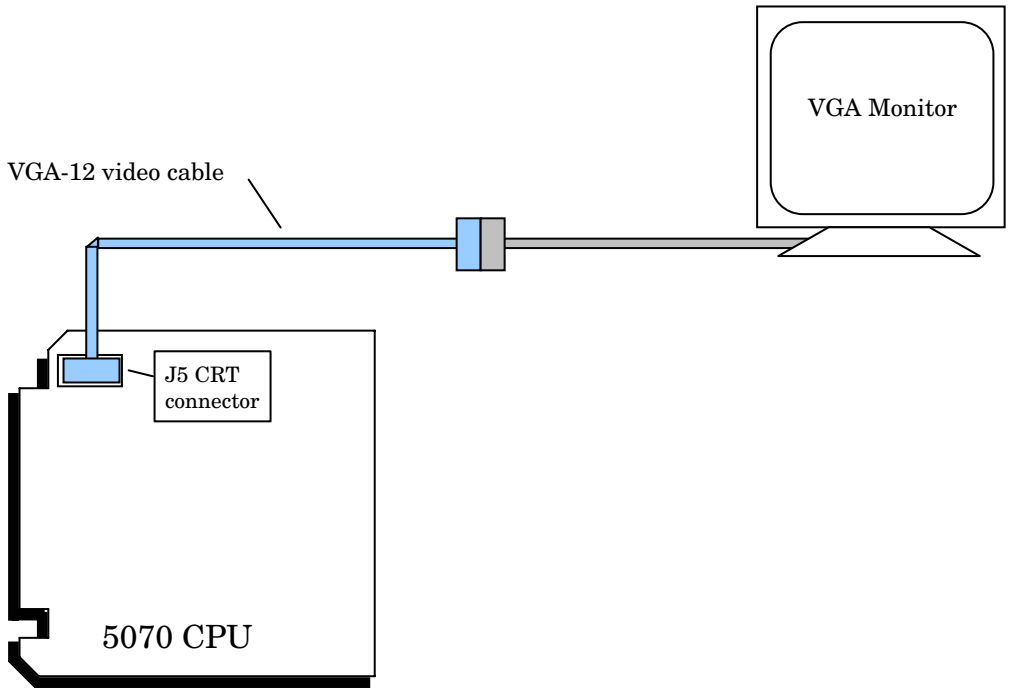
- High performance Asilant VGA 69000 video controller
- 2 MB DRAM for display buffering
- CRT support with resolutions to 1024 x 768 x 16 at 75 MHz
- Flat panel support with the following resolutions:
  - — 640 x 480 x 24 bpp
  - — 800 x 600 x 24 bpp
  - — 1024 x 768 x 16 bpp
  - — 1280 x 1024 x 8 bpp on some selected displays
- Support for plasma, EL and LCD displays
  - — 3V and 5V flat panel support
  - — Flat panel power sequencing
- PCI bus interface for fast execution

Connecting a monitor

To use a monitor or a flat panel, the video jumper (W12[5-6]) must be installed. This is the default configuration. The 16–pin connector at J5 supports an analog VGA/SVGA/XVGA CRT color or monochrome monitor. Refer to Figure 12–2.

The 5070 supports both an analog monitor and/or a flat panel display. The CT.COM and FP.COM programs allow you to toggle between the monitor and the flat panel. If the flat panel supports simultaneous mode, the SM.COM program will allow you to display images from both the monitor and the flat panel at the same time. These programs are in the Utilities zip file along with other diagnostic and configuration utilities. Refer to the README.DOC file.

Figure 12–1 The 5070 and a VGA monitor



To connect a monitor:

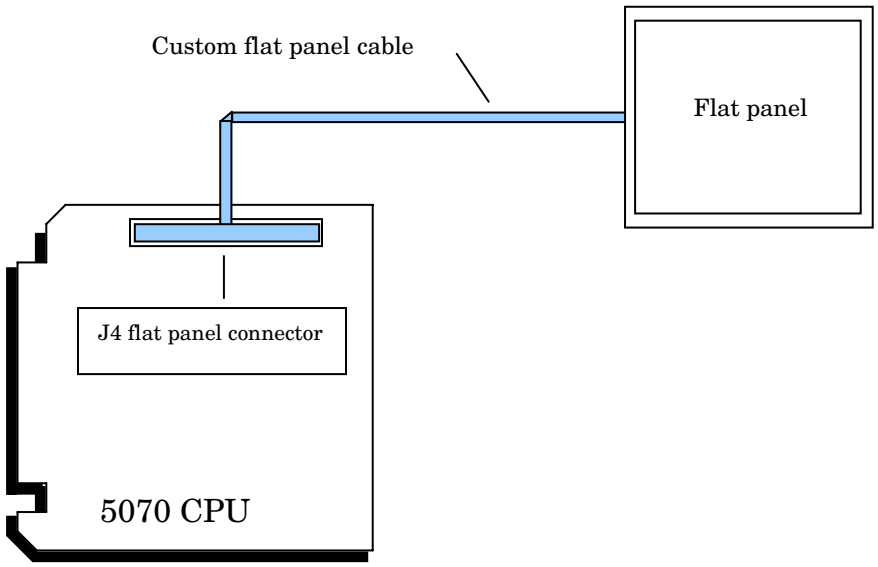
1. Using a VGA–12 video cable, plug the VGA–12 video cable into J5 on the 5070.
2. Plug the DB-15 end of the VGA-12 cable into the VGA cable of the monitor.

Connecting a flat panel display

Due to the varied selection of available flat panel, the 5070 is factory configured and programmed for a VGA/SVGA/XVGA CRT monitor. If you wish to use a flat panel, you must reprogram the video BIOS with the appropriate flat panel driver. To reprogram your video BIOS refer to *Programming the video BIOS* in this chapter. Note that both 3V and 5V flat panels are supported. Jumper W2 selects the flat panel voltage.

The Utilities zip file contains text files for each of the supported flat panels. These text files include wiring diagrams specific to individual flat panels. Refer to the specific text file associated with your flat panel to build an interface cable, and to determine the correct settings for the flat panel jumpers.

Figure 12–2 The 5070 and a flat panel



Flat panels requiring bias voltage

Some flat panels require a bias voltage. To determine if your flat panel requires bias voltage, refer to the text file in the Utilities zip file which is specific to your flat panel or refer to your flat panel information. If your flat panel requires a bias voltage, refer to the manufacturer's documentation for procedures on supplying the proper bias voltage.

WARNING!

Since improper voltage levels can severely damage the flat panel, make sure the bias voltage is correct before the flat panel is connected to the 5070.

Connecting the flat panel to the 5070

Text files are located in the Utilities zip file. These text files include wiring diagrams specific to individual flat panels. Refer to the specific text file associated with your flat panel to build your cable. The maximum recommended cable length is 18 inches and EMI shielding is also recommended to reduce emissions that affect video quality.

Jumpers W1, W2, and W7 are configuration jumpers for flat panels. Jumper W1 sets the polarity for the SHFCLK signal for flat panels that invert the shift clock polarity. Jumper W2 sets the interface voltage for 3.3V or 5V panels. Jumper W7 routes either the latch pulse or blank (M) signal to J4. J7 also routes the +12V or +5V back-light voltage to J4.

Table 12-1 shows the jumper settings for SHFCLK, flat panel polarity, and signal routing to J4.

Table 12–1 Flat panel jumpers: W1, W2, W7

W1, W2, W7 – Flat Panel		
Function	**Jumper**	**Description**
SHFCLK polarity	W1[1-3]	Inverted polarity
	W1[2-4]*	Normal polarity
Flat panel voltage	W2[1-2]*	5V panel
	W2[2-4]	3V panel
Latch pulse /	W7[2-4]*	Route latch pulse to J4, pin 8
Blank pulse routing	W7[1-3]	Route blank pulse to J4, pin 8
Back-light voltage	W7[7-9]	Route 12V to J4, pin 2
select	W7[8-10]*	Route 5V to J4, pin 2

* = default

1. Refer to the text file associated with your flat panel to determine the supply voltage for your panel, and whether a bias voltage is required.
2. Connect a cable from the flat panel to the flat panel connector located at J4 on your 5070. Refer to Figure 12–3.

WARNING!

Improper wiring or connection from the flat panel to the 5070 can damage the 5070 and the flat panel. Verify the flat panel cable connections before connecting the cable to the 5070 and applying power to the system.

Note See *Appendix A: Mating connectors* for mating information.

Programming the video BIOS

The 5070 BIOS is factory configured and programmed for a VGA/SVGA/XVGA CRT monitor. If you wish to use a flat panel, you must reprogram the video BIOS with the appropriate flat panel driver. To reprogram your video BIOS, refer to the README.HTM file in the Utilities zip file. If your particular display is not currently listed, contact Octagon Technical Support (303–426–4521) for assistance.

To reprogram the video BIOS you will need:

- 5070 HDC-18-MPC-Multiport cable, #6240
- 5070 HDC-18-HDD/FDD drive cable, #6239
- VGA-12 video cable, #4865
- PS-2 style keyboard
- VGA monitor
- Floppy drive
- 5070 Utilities zip file (see page 139)

To program a new video BIOS to support a flat panel:

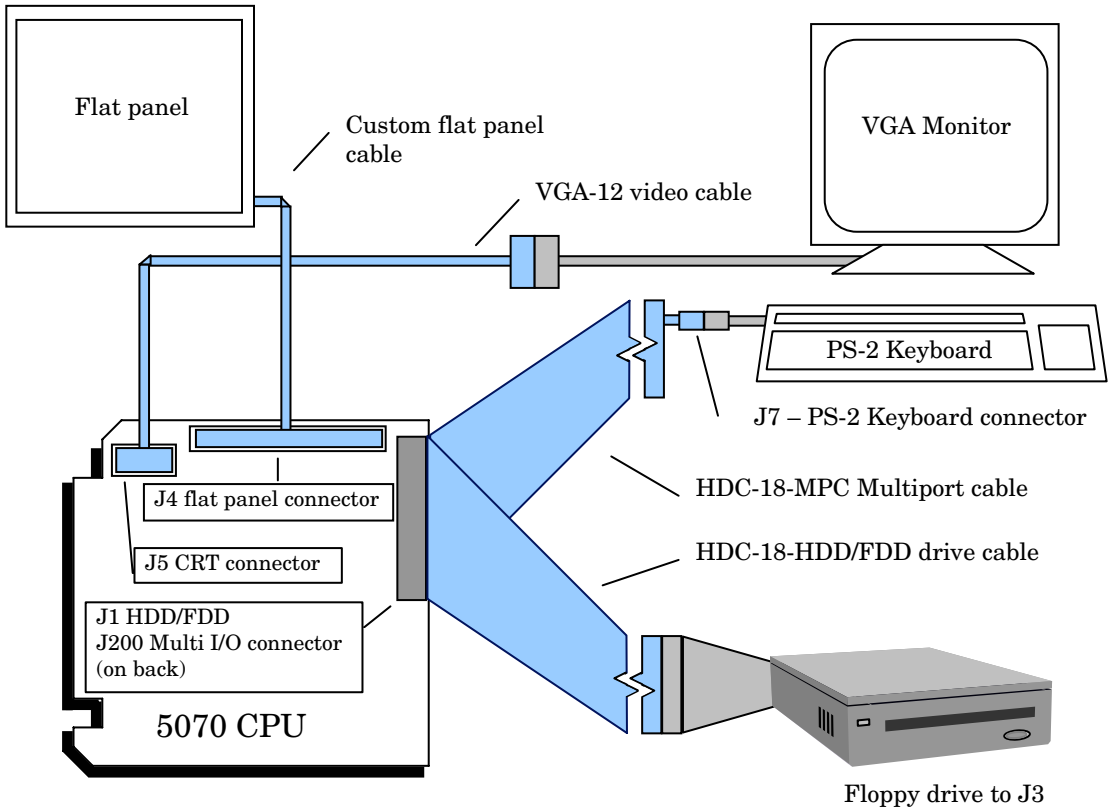
1. Attach a VGA monitor, a PS-2 compatible keyboard, and a floppy drive to the 5070. Refer to Figure 12–3.

Note If a monitor and keyboard are not available, connect the 5070 to your PC by using a remote serial console. Refer to the *Serial Console* section in the *Console devices* chapter.

Note When a new video BIOS is installed, the CRT may no longer display information since the new BIOS may only support a flat panel. If the flat panel is not functioning properly, is hooked up incorrectly, or the BIOS used does not support the installed flat panel, the serial console mode may have to be used to load a new BIOS. Refer to the *Serial Console* section in the *Console devices* chapter.

2. Refer to the 5070_reprogram.htm document in the Utilities zip file for reprogramming instructions.

Figure 12–3 Reprogramming the flat panel BIOS



Chapter 13: Ethernet

Description

The 5070 provides a 10/100Base–T Ethernet port and supports the IEEE 802.3 Ethernet standard. The Ethernet controller IC chip provides the following:

- 8K x 16 SRAM buffer
- Integrated 10/100 Base–T transceiver interface
- Two LEDs for link and traffic status integrated into connector

The 5070 Ethernet uses twisted–pair wiring cable, which is built in a star configuration. The interface terminates at the standard, 8–position, RJ–45 latching phone jack that can be vertically accessed.

CAUTION

Use a strain relief loop when connecting to the 5070 Ethernet connector to avoid damaging the connector.

For more information on programming the Ethernet port, see the README.DOC in the Ethernet directory in the Utilities zip file (see page 139). By default the Ethernet port connects to IRQ10, but can be reconfigured to IRQ11 via the SETIRQ.DOC program.

Table 13–1 Ethernet LEDs

Ethernet LEDs		
Function	**Color**	**Description**
Activity LED	Amber	Activated by access to I/O space
Link LED	Green	Activated by network link

* = default

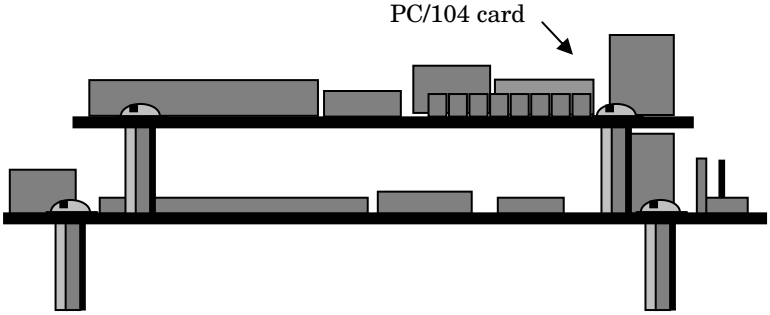
Chapter 14: PC/104 expansion

Description

This connector allows you to interface to one or two PC/104 modules including A/D converters, digital I/O, serial ports, etc. The 5070 supports 8– and 16–bit cards**.** These modules can be stacked on top of the 5070 to form a highly-integrated control system.

Note The actual maximum number of modules in a stack is limited primarily to the capacitive loading on the bus and the electrical noise environment. This is especially true when wide temperature operation is required. Good design practice dictates that the modules present only one load to each bus signal. Unfortunately, there are modules on the market that violate this practice by loading the bus more heavily. Typically, it is the IOW*, IOR*, MEMW*, and RSTDRV* lines. For example, if the IOW* line is routed to four ICs on the module without a buffer, then the loading is equivalent to four PC/104 modules. Stacks with three or more expansion modules should be carefully tested under all environmental conditions. If possible, query the manufacture of the expansion module regarding loading. All Octagon products present one load.

Figure 14–1 Typical PC/104 module stack



WARNING!

When installing any PC/104 module, avoid excessively flexing the 5070 card. Mate pins correctly and use the required mounting hardware.

Chapter 15: USB

Description

The 5070 contains two Universal Serial Bus (USB) ports. USB is a hardware interface for low-speed peripherals such as the keyboard, mouse, joystick, scanner, printer and telephony devices. USB has a maximum transfer rate of 12 Mbits/sec, and up to 127 devices can be attached. Peripherals can be plugged in and unplugged without turning the system off.

Any USB device can be plugged directly into either USB socket on the 5070 HDC-18-MPC-Multiport Interface Cable or into a multi-port hub that then plugs into a USB port. An operating system capable of utilizing USB or USB specific device drivers is required for USB operation.

Overview: Section 3 – System management

Section 3 provides information on managing the 5070 in the areas of internal control and troubleshooting. The following chapters are included:

Chapter 16:	Watchdog timer and hardware reset
Chapter 17:	Serial EEPROM
Chapter 18:	Temperature Sensor
Chapter 19:	CPU clock, ISA bus interrupt routing
Chapter 20:	System jumpers, user jumper, system LEDs
Chapter 21:	Troubleshooting

Chapter 16: Watchdog timer and hardware reset

Description

The watchdog timer is a fail–safe against program crashes or processor lockups. It has a programmable timeout period, ranging from 0.5 seconds to 2 seconds. INT17 software calls, a built–in function on the 5070, are used to enable and set the timeout, extend the timeout, strobe, and disable the watchdog timer from your application. If the timer expires, it performs a hardware reset.

Watchdog function definitions using enhanced INT 17h handler

This section provides definitions for the watchdog functions using the INT17 handler (I17HNDLR.EXE). I17HNDLR.EXE is a TSR program. It is called out by the 5070 BIOS. Once executed it is active, but it must be executed each time the system is rebooted. If you use a different BIOS the INT17 functions can still be used by your application. Copy the utility to your hard drive and add it to your AUTOEXEC.BAT.

Note The INT17 functions can only be used with DOS operating systems.

Enable watchdog

```
Function:            fdh
Subfunction:         01h
Purpose:             To enable the watchdog.
Calling registers:   AH     fdh
                     AL     01h
                     DX     ffffh
Return registers:    None
Comments:            This function enables the watchdog. Once the
                     watchdog is enabled, it has to be strobed at a
                     period of not less than 2 seconds or until the
                     watchdog is disabled.  Otherwise, a system reset
                     will occur.
Programming example:
                      /* Inline assembly code for Borland C++ 3.1*/
                         asm {
                         mov ax,0fd01h
                         mov dx,0ffffh
                         int 17h
                         }
```

Strobe watchdog

```
Function:            fdh
Subfunction:         02h
Purpose:             To strobe the watchdog.
Calling registers:   AH     fdh
                     AL     02h
                     DX     ffffh
Return registers:    None
Comments:            This function strobes the watchdog. Once the watch
                     dog is enabled, it has to be strobed at a period
                     of not less than 2 seconds or until the watchdog
                     is disabled (Function 0fdh, Sub-function 3).
                     Otherwise, a system reset will occur.

Programming example:
                      /* Inline assembly code for Borland C++ 3.1*/
                      asm {
                      mov ax,0fd02h
                      mov dx,0ffffh
                      int 17h
                          }
```

Refer to the examples TESTWDOG.CPP and TESTWDOG.EXE in the Utilities zip file (see page 139).

Disable watchdog

```
Function:            fdh
Subfunction:         03h
Purpose:             To disable the watchdog.
Calling registers:   AH     fdh
                     AL     03h
                     DX     ffffh
Return registers:    None
Comments:            This function disables the watchdog Once the
                     watch dog is enabled, it has to be strobed at a
                     period of not less than the time specified when
                     enabled (see Function 0fdh, Sub-function 2) or
                     until the watchdog is disabled.  Otherwise, a
                     system reset will occur.
Programming example:
                      /* Inline assembly code for Borland C++ 3.1*/
                      asm {
                      mov ax,0fd03h
                      mov dx,0ffffh
                      int 17h
                          }
```

Extending Watchdog time-out

Since the hardware watchdog on the card is limited to 2 seconds (which

may be shorter than some applications require) the following additional INT 17 functions have been defined. The WD Timer functions are a software extendable hardware watchdog.

The functions use a counter which decrements every 54ms (about 18.2 times per second). Periodically (and before the HW watchdog triggers and resets the card) the counter strobes the HW watchdog keeping the system alive. If the counter reaches 0 the HW watchdog is no longer strobed, allowing the system to reset.

There are two INT 17 functions implemented to use this watchdog timer. The set WD Timer (INT 17 function fd, subfunction 4) is used to set the counter. The count (passed in BX) sets the counter. For example if a 10 second timer is necessary, use 182 in BX. (18.2 counts per second * 10 seconds = 182). Passing 0 in BX stops the function (and the system will only be controlled by the HW watchdog, which is a 2 second timeout).

Note: Before using these functions the enable watchdog call (INT 17 function fdh, subfunction 1) must be used to enable the HW watchdog. Likewise the disable watchdog call (INT 17 function fdh, subfunction 3) should be used to disable the watchdog (after calling the set watchdog timer function with 0 in BX.

The second function increment WD Timer (INT 17 function fd, subfunction 5) is used to increment the current count. To add a 5 second delay pass 91 in BX, if the current count was 18 (about 1 second), the new count would be 109 (about 6 seconds).

CAUTION

The system timer interrupt must not be disabled or changed in frequency. Interrupts must not be disabled for longer than the HW timeout. This function is designed for use in DOS since it uses the INT 1C system timer interrupt to count down and strobe the HW watchdog. Once the SW watchdog counter expires, the HW watchdog will expire which may take up to 2 seconds.

Set watchdog timer

Function:	fdh
Subfunction:	04h
Purpose:	To set the SW watchdog counter. Used by the INT 17 SW watchdog timer which strobes the HW watchdog periodically. Allowing longer timeouts than are available from the HW.
Calling registers:	AH fdh
	AL 04h
	BX count in number of system timer ticks
	DX ffffh
Return registers:	Carry flag cleared if successful. BX = previous

	value of the counter.
Comments:	Use this function after enabling the HW watchdog via subfunction 1. The BX value is in timer tick increments, so 182 = 10 seconds, 91 = 5 seconds, etc. The previous count is returned so a program can determine the optimal setting after experimental use.

Increment watchdog timer

Function:	fdh
Subfunction:	05h
Purpose:	To increment the SW watchdog counter.
Calling registers:	AH fdh
	AL 05h
	BX Value to increment the count by
	DX ffffh
Return registers:	Carry flag cleared if successful. BX = previous value of the counter.
Comments:	Use this function to add to the current watchdog timer.

Hardware reset

The 5070 has a button which allows you to reset the system without turning off the power. This provides a more complete reset than the <CTRL><ALT><DEL> method. The J3 connector also has a reset function. By depressing the button (connecting the two lines), the circuit is pulled to ground and resets the system.

The RESET command accomplishes the same thing as the reset button. Refer to the component diagram in the *Quick start* chapter for the location of the reset button.

WARNING!

When using COM1 as the console, the <CTRL><ALT> <DEL> commands on the host system keyboard only reset the host system. Use the RESET command to issue a hardware reset on the 5070.

Chapter 17: Serial EEPROM

Description

Up to 512 bytes of user–definable data can be saved in the serial EEPROM. The serial EEPROM does not require battery backup to maintain the data when the system power is off. The serial EEPROM is easily accessible via software interrupts by most programming languages.

Enhanced INT 17h function definitions

The serial EEPROM definitions include the following functions: Read a single word from serial EEPROM, Write a single word to serial EEPROM, Read multiple words from serial EEPROM, Write multiple words to serial EEPROM, and Return serial EEPROM size.

Serial EEPROM

Read a single word from the serial EEPROM

```
Function:              fch
Subfunction:           00h
Purpose:               To read a single word from the on-board serial
                       EEPROM.
Calling registers:     AH     fch
                       AL     00h
                       BX     Word address (zero based)
                       DX     ffffh (User area relative address)
                              9876h (Absolute address)
Return registers:      Carry flag cleared if successful
                       AX     Word read
                       Carry flag set if error
                       AL     Error code
                  Error code   Meaning
                       ffh    Unknown error
                       01h    Function not implemented
                       02h    Defective serial EEPROM
                       03h    Illegal access
Comments:              This function reads a word from the user area of
                       the serial EEPROM.
Programming example:
                        /* Read word 2*/
                        unsigned int seeData;
```

```
/* Inline assembly code for Borland C++ 3.1*/
asm {
mov ax,0fc00h
mov bx,02h /* Read word 2*/
mov dx,0ffffh
int 17h
mov seeData,ax/* store data in c environment */
  }
```

Write a single word to the serial EEPROM

Function:	fch
Subfunction:	01h
Purpose:	To write a single word to the on-board serial EEPROM.
Calling registers:	AH fch
	AL 01h
	BX Word address (zero based)
	CX Data word to write
	DX ffffh (User area relative address)
	9876h (Absolute address)
Return registers:	Carry flag cleared if successful
	Carry flag set if error
	AL Error code

Error code	Meaning
ffh	Unknown error
01h	Function not implemented
02h	Defective serial EEPROM
03h	Illegal access

Comments: This function writes a word to the user area of the serial EEPROM.

Programming example:

```
/* Write 0x1234 to word 3*/
unsigned int seeData = 0x1234;
/* Inline assembly code for Borland C++ 3.1*/
asm {
    mov ax,0fc01h
    mov bx,03h /* Write word 3*/
    mov cx,seeData/* Get write data from c
environment */
    mov dx,0ffffh
    int 17h
      }
```

Read multiple words from the serial EEPROM

Function:	fch
Subfunction:	02h
Purpose:	To read multiple words from the on-board serial EEPROM.
Calling registers:	AH fch
	AL 02h
	BX Word address (zero based)
	CX Word count
	DX ffffh
	ES:DI Destination pointer

Return registers: Carry flag cleared if successful
AX Word read
Carry flag set if error
AL Error code

Error Code	Meaning
ffh	Unknown error
01h	Function not implemented
02h	Defective serial EEPROM
03h	Illegal access

Comments: This function reads multiple words from the user area of the serial EEPROM.

Programming example:

```
/* Read 10 words starting at word 5*/
unsigned int far*seeDataPtr = new unsigned
int[10];
/* Allocate storage */
/* Inline assembly code for Borland C++ 3.1*/
asm {
    mov ax,0fc02h
    mov bx,05h /* Read starts at
           word 5*/
    mov cx,10  /* Read 10 words */
    mov dx,0ffffh
    les di,seeDataPtr
    int 17h
    }
```

Write multiple words to the serial EEPROM

Function: fch
Subfunction: 03h
Purpose: To write multiple words to the on-board serial EEPROM.
Calling registers: AH fch
AL 03h
BX Word address (zero based)
CX Word count
DX ffffh
DS:SI Source pointer
Return registers: Carry flag cleared if successful
Carry flag set if error
AL Error code

Error Code	Meaning
ffh	Unknown error
01h	Function not implemented
02h	Defective serial EEPROM
03h	Illegal access

Comments: This function writes multiple words to the user area of the serial EEPROM.

Programming example:

```
/* Write 8 words starting at word 6*/
unsigned int far*seeDataPtr = new unsigned
int[8];
/* Allocate storage */
    unsigned int far* tmpPtr = seeDataPtr;
    for(int I=0;I<8;I++)
```

```
    *seeDataPtr = I; /* initialize data */
 /* Inline assembly code for Borland C++ 3.1*/
    asm {
    push   ds
    mov ax,0fc03h
    mov bx,06h /* Write starts at
          word 6*/
    mov cx,8   /* Write 8 words */
    mov dx,0ffffh
    lds si,seeDataPtr
    int 17h
    pop ds
       }
```

Return serial EEPROM size

Function:	fch
Subfunction:	04h
Purpose:	To obtain the size of the on-board serial EEPROM.
Calling registers:	AH fch
	AL 04h
	DX ffffh
Return registers:	Carry flag cleared if successful
	AX Size of the serial EEPROM (in words)
	BX Size available to user (in words)
	Carry flag set if error
	AL Error code

Error code	**Meaning**
ffh	Unknown error
01h	Function not implemented
02h	Defective serial EEPROM
03h	Illegal access

Comments: This function returns the size (in words) of the serial EEPROM. Since the user cannot access all of the serial EEPROM, this function determines how much space is available to the user. This avoids the user from accessing unavailable addresses.

Programming example:

```
unsigned int seeUserSize;
/* Inline assembly code for Borland C++ 3.1*/
asm {
    mov ax,0fc04h
    mov dx,0ffffh
    int 17h
    mov seeUserSize,bx
    }
```

Chapter 18: Temperature sensor

Description

The temperature sensor is a thermometer located on the 5070 card.

Temperature sensor INT17h function definitions

The temperature sensor allows you to set over and under temperature limits, and to read the ambient temperature. There is also a configuration register. For complete details on the configuration register, refer to the Dallas Semiconductor DS1775 SOT23 Digital Thermometer and Thermostat Product Review sheet.

Digital data is written to/read from the component with MSb first, in two's complement format, with the MSb denoting positive or negative temperature. All temperatures are in Celsius.

Write TEMP SENSOR register pointer

```
Function:              0edh
Subfunction:           00h
Purpose:               To set the TEMP SENSOR pointer to internal
                       registers
Calling Registers:     AH     0edh
                       AL     00h
                       BH     Number of bits in internal register 8 or
                              16. Only the configuration register is 8
                              bit, all others are 16 bit.
                       BL     TEMP SENSOR register pointed to:
                              0=TEMPERATURE
                              1=CONFIGURATION
                              2=HYST over temp
                              3=OS under temp
                              4=0x0ff, reserved
                       DX     ffffh
Return Registers:             Carry flag cleared if successful
                              Carry flag set if error
                       AL     Error code
Comments:              This function shall be used to set the TEMP
                       SENSOR internal register pointer.
Programming example 1: /* Inline assembly code for Borland C++ 3.1*/
                       unsigned char aData;
                       asm {
```

```
                mov ax,0ed00h
                mov bx,0801h
                mov dx,ffffh
                int 17h
                mov aData,al
            }
```

Read TEMP SENSOR current register

```
Function:               0edh
Subfunction:            01h
Purpose:                Reads the register currently pointed to by
                        the TEMP SENSOR register pointer.
Calling Registers:      AH     0edh
                        AL     01h
                        DX     ffffh
Return Registers:       Carry flag cleared if successful
                        Carry flag set if error
                        AL     Error code
                        BX     Data read from the TEMPERATURE SENSOR
Comments:               This function shall be used to read the TEMP
                        SENSOR register currently pointed to.
Programming example 1: /* Inline assembly code for Borland C++ 3.1*/
                        unsigned char aData;
                        unsigned int bData;
                        asm {
                               mov ax,0ed01h
                               mov dx,ffffh
                               int 17h
                               mov aData,al
                               mov bData,bx
                             }
```

Write TEMP SENSOR current register

```
Function:               0edh
Subfunction:            02h
Purpose:                Writes the register currently pointed to by the
                        TEMP SENSOR register pointer.
Calling Registers:      AH     0edh
                        AL     02h
                        BX     Data to write
                        DX     ffffh
Return Registers:       Carry flag cleared if successful
                        Carry flag set if error
                        AL     Error code
Comments:               This function shall be used to write the TEMP
                        SENSOR register currently pointed to.
Programming example 1: /* Inline assembly code for Borland C++ 3.1*/
                        unsigned char aData;
                        unsigned int bData;
                        asm {
                               mov ax,0ed01h
                               mov bx,bData
```

```
  mov dx,ffffh
  int 17h
  mov aData,al
}
```

Chapter 19: CPU clock, ISA bus interrupt routing

CPU clock

The CPU clock speed can be configured to run at 33, 50, 66, 99, 100 or 128 MHz by changing the W9, W12, and W13 jumpers. Table 19-1 shows the clock speed jumper configuration.

Table 19-1 CPU clock speed jumpers: W9, W12, W13

W9, W12, W13 – CPU Clock Speed			
Clock speed	**W9**	**W12**	**W13**
33 MHz	None	[9-10][11-12]	[1-2]
50 MHz	[1-2][3-4]	[9-10][11-12]	[1-2]
66 MHz	None	[11-12]	[1-2]
99 MHz	None	None	[1-2]
100 MHz	[1-2][3-4]	[11-12]	[1-3]
128 MHz*	[1-2]*	[11-12]*	[1-3]*

*** = default**

ISA bus interrupt routing

ISA bus and on-card-source IRQs 3 and 4 can be routed to IRQs 10 and 11 with jumper block W14. This allows the lower-ordered ISA interrupts to be connected to the unused higher-order interrupts. The application software must be aware of the interrupt that is set for the particular resource. Table 19-1 shows the interrupt routing.

Table 19-2 Interrupt routing jumper: W14

W14 – IRQ interrupt routing		
Source	**Destination**	**W14**
Bus IRQ3	IRQ3	[5-7]*
	IRQ10	[3-5]
Bus IRQ4	IRQ4	[6-8]*
	IRQ11	[4-6]

*** = default**

Chapter 20: System jumpers, user jumper, system LEDs

System jumpers

Various system function options are selected with jumper W12.

The “S” jumper selects whether the card boots from user defined parameters (defined in the *Setup Programs* chapter), or the BIOS defaults. Removing this jumper allows the user to return to factory programmed defaults.

The “X” enables or disables the BIOS extension area. The default is enabled, booting from the on-card flash device, and to using INT17 calls.

The “V” jumper enables or disables the on-card video, allowing an external video card, or the serial console to be used.

The “U” jumper is user defined and can be used for program control.

Jumpers “0” and “1” are used to set the CPU clock speed. Refer *to Chapter 19: CPU clock, ISA bus interrupt routing*.

Table 19-3 System configuration jumper: W12

W12 – System Configuration		
Label	**Description**	**W12**
S	System parameters option jumper: Installed = enable User Setup options* Removed = enable BIOS Setup default	[1–2]*
X	BIOS extension enable Installed = enable extended BIOS* required to boot from on-board flash required to use INT17 calls Removed = disables extended BIOS	[3–4]*
V	Video jumper: Installed = enable on-card video* Removed = disable on-card video	[5–6]*
U	User jumper	**[7–8]***
0	CPU clock jumper – see CPU clock speed table	[9–10]
1	CPU clock jumper – see CPU clock speed table	[11–12]*
R	Reserved	[13–14]
Z	Reserved	[15–16]

*** = default**

User jumper

The user jumper is W12[7–8] and is associated with bit 4 of the GPIO register. The INT17 functions provide an easy method to implement software routines according to whether or not a jumper has been installed. Refer to the *INT17 calls for user jumper and LEDs* section in this chapter. The user jumper can also be read by reading I/O port 8104h, bit 4.

System LEDs

CR3 and CR5 are system activity LEDs. CR3 amber indicates IDE device access from the CompactFlash or an IDE device connected to J1. CR3 green is a BIOS status LED that can also be user defined. CR5 amber is a BIOS status LED that can also be user defined. CR5 green indicates floppy disk access from a floppy disk drive connected J1.

Table 19-4 System LEDs

System LEDs			
LED	**Function**	**Color**	**Description**
CR3	IDE activity	Amber	Activated by IDE access
	BIOS / user LEDA	Green	BIOS status / user defined
CR5	BIOS / user LEDB	Amber	BIOS status / user defined
	FDD activity	Green	Activated by floppy access

The CR3 amber LEDA and CR5 green LEDB signals are used by the system BIOS to indicate the BIOS processing state. When the card first starts up the CR3 amber LED is on and the CR5 green LED is off. Once the card boots, the CR3 amber LED turns off and the CR5 green LED is on.

The bicolor LEDs also indicates memory suspend status. Upon entering memory suspension, the CR5 green LED turns off and the CR3 amber LED turns on. On a resume condition, the CR3 amber LED turns off and the CR5 green LED turns on.

When the system enters the cool down clocking state, the CR5 green LED and the CR3 amber LED turn on. When the system exits the cool down clocking state, the CR3 amber LED turns off and the CR5 green LED remains on.

If the BIOS finds an error during the power on self test (POST) the CR3 amber LED is flashed in a pattern indicating the POST code failure. Refer to the *Troubleshooting* chapter.

User access to the LEDs

The LEDs are controlled by bits 6 and 7 of the GPIO system on the 5070. The base I/O address for controlling the GPIO outputs is 8100h. Bit 6 is CR3 green (LEDA) and bit 7 is CR5 amber (LEDB). The user can modify the state of these LEDs to indicate system or program status. Refer to the *INT17 calls for user jumper and LEDs* section in this chapter.

INT17 calls for user jumper and LEDs

This section provides definitions for the functions using the INT17 handler, I17HNDLR.EXE.

I17HNDLR.EXE is a TSR program and is called out by the 5070 BIOS. By default, when the "X" jumper is on, the INT17 extended BIOS is operational. If the "X" jumper is removed and DOS is the operating system, the I17HNDLR.EXE TSR can be used. Once executed, the TSR is active, but it must be executed each time the system is rebooted. Copy the I17HNDLR.EXE utility to your boot device and add it to your AUTOEXEC.BAT.

Note The INT17 functions can only be used with DOS operating systems. If you use a different operating system, the INT17 functionality can still be used by your application but must be integrated into your software.

GPIO register

The GPIO register is used to control the digital inputs and outputs, the user jumper, and the CR3 and CR5 system activity LEDs.

IMPORTANT

To assure that output bits are not inadvertently changed, always use the read-modify-write method to change this register.

Table 19–5 GPIO register

GPIO register								
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Name	CR5 green LEDB	CR3 amber LEDA	Reserved	User jumper	GPIO3	GPIO2	GPIO1	GPIO0
Function	Status LED	Status LED		User defined	Digital output	Digital output	Digital input	Digital input
Direction	Output	Output		Input	Output	Output	Input	Input

Write LEDs

```
Function:            0ech
Subfunction:         01h
Purpose:             To write a value to the GPIO LED outputs at
                     GPIO[6] and GPIO[7].
Calling Registers:   AH     0ech
                     AL     01h
                     DI     mmxxmmxx ddxxddxx
                            The m bits are mask bits. If set to 1, the
                            corresponding d bit will be written. If
                            set to 0, the corresponding d bit will
                            remain at current value.
                                          DI mask bit   DI data bit
                                   GPIO[7]   15            7
                                   GPIO[6]   16            6
                                   GPIO[3]   11            3
                                   GPIO[2]   10            2
                     DX     ffffh
Return Registers:    Carry flag cleared if successful
                     Carry flag set if error
                     AL     Error code
Comments:            This function is used to write to GPIO output
                     pins.
Programming example 1:
                     /* Inline assembly code for Borland C++ 3.1*/
                     asm {
                            mov ax,0ec01h
                            /* Set GPIO[6] to zero and set GPIO[7] to
                            one */
                            mov di,0c080h
                            mov dx,0ffffh
                            int 17h
                             }
```

Read user jumper

```
Function:            0fbh
Sub-Function:        0bh
Purpose:             To read user jumper
Calling Registers:   AH     0fbh
                     AL     0bh
                     DX     0ffffh
Return Registers:    Carry flag cleared if successful
                     AL     Jumper data
                            bit 0 user jumper A. 1=on, 0=off
                            bit 1 user jumper B. 1=on, 0=off
                     Carry flag set if error
                     AL     Error code
Comments:            This function is used to read the user jumper
Programming example 1: /* Inline assembly code for Borland C++ 3.1 */
                     unsigned char aData;
                     asm {
                            MOV AX, 0fb0bh
```

```
        MOV DX, 0ffffh
        INT 17h
        MOV aData, AL
        }
if (aData & 1)
  printf("U jumper is ON\n");
else
  printf("U jumper is OFF\n");
```

Chapter 21: Troubleshooting

If your system is not working properly, check the following items.

No system LED activity

If there is no LED activity at CR3 or CR5, check the following:

- Make sure all PC/104 expansion cards are removed from the 5070. This ensures that other cards are not interacting with the 5070.
- Remove the jumper from the “S” position at W12[1–2].
- Check all power connections to the 5070 card.
- Measure the supply voltage at the J7 power connector and verify that the voltage at the 5070 card is +5V (+/–0.25V).
- Make sure your power module provides +5V (+/–0.25V) and at least 2.5A of current.

No CRT or flat panel video

If the LEDs appear to be functioning properly, but there is no video activity, check the following:

- Make sure all PC/104 expansion cards are removed from the 5070. This ensures that other cards are not interacting with the 5070.
- Remove the jumper from the “S” position at W12[1–2].
- If using a CRT monitor, check the cable and connections going from the J5 connector to the monitor.
- If using a flat panel display, check the following:

    1. Cable and connections going from the J4 connector to the flat panel display.
    2. Check the power and cables going to the flat panel display.
    3. Make sure that the power module has enough current capacity to power both the 5070 card and the flat panel.

Video is present but is distorted

If video is present but is distorted, check the following:

- Make sure all PC/104 expansion cards are removed from the 5070. This ensures that other cards are not interacting with the 5070.
- Remove the jumper from the “S” position at W12[1–2].
- If using a CRT monitor, check the cable and connections going from the J5 connector to the monitor.
- If using a flat panel display, check the following:

    1. Cable and connections going from the J4 connector to the flat panel display
    2. Signal cable going to the flat panel display. If the cable length is too long, the distortion may be caused by noise. 18” or less is the recommended length. Cable shielding may be required.
    3. Power cable going to the flat panel display.
    4. Power module for the flat panel. Make sure that the power module has enough current capacity to power both the 5070 card and the flat panel.

No serial console activity

If the serial console does not appear to be functioning correctly, check the following:

- Make sure all PC/104 expansion cards are removed from the 5070. This ensures that other cards are not interacting with the 5070.
- Remove the jumper from the “S” position at W12[1–2].
- Make sure the COM1 connector on the 5070 HDC-18-MPC-MULTIPORT cable is used.
- Make sure a null modem adapter is installed between COM1 of the HDC-18-MPC-MULTIPORT cable and the serial port of your PC.
- Make sure that your terminal emulator (such as HyperTerminal) on your PC is set-up properly. Refer to the *Console devices* chapter. Refer to the HyperTerminal manual for information on setting up communication parameters.
- After verifying the above conditions, you can monitor voltage levels by connecting an oscilloscope between the TxD* line on COM1 and ground.

After power–up, you should see a burst of activity on the oscilloscope screen. The voltage level should switch between +/–8V.

Garbled console screen activity

If you do get activity on your console screen but the message is garbled, check the following:

- Remove the jumper from the "S" position at W12[1–2] to ensure the default settings for COM1.
- Make sure that your terminal emulator (such as HyperTerminal) on your PC is set-up properly. Refer to the *Console devices* chapter. Refer to the HyperTerminal manual for information on setting up communication parameters.

System generates a BIOS message but locks up when booting from SSD1

- Remove the jumper from the "S" position at W12[1–2] and reboot.
- Display the directory of SSD1 and verify that all the necessary boot files exist. Copy any missing files to SSD1.
- If no files are missing, remake SSD1 to overwrite any files which may have become corrupted. In addition, you may want to do a **FXFMT** and **SYS** to SSD1.

Note FORMAT requires a floppy disk drive to restore system files.

System will not boot from CompactFlash

Many CompactFlash devices as shipped from the factory are not bootable devices. Refer to the *SSD1, CompactFlash, SDRAM, battery backup, and Z–tag Interface* chapter to make your CompactFlash bootable.

System locks up on power–up; may or may not respond to reset switch

A common cause is using a non–Octagon power supply such as a PC desktop supply. Most of these PC supplies are rated at 5V at 20A or more. Switching

supplies usually requires a 20% load to operate properly, that is, 4A or more. Since a typical Micro PC system takes less than 2A, the supply does not regulate properly. Output drift up to 6–7V and/or 7–8 voltage spikes have been reported. If the power supply comes up slowly (that is, longer than 10 ms), the sequencing of ICs on the board may be out of sync, thus, causing the system to lock up.

Octagon supplies are designed to ramp up fast, discharge fast on power–down and to regulate properly under a no load condition.

System locks up after power–down/power–up

If the power supply does not drain below 0.7V, the CMOS components on the card will act like diodes and forward bias. This is typically caused by using power supplies that have large output capacitors. Either use a different power supply that discharges faster, leave the power off until the supply has adequate time to discharge or place a 100 ohm, large wattage resistor across the output capacitor.

Octagon supplies are designed to ramp up fast, discharge fast on power–down and to regulate properly under a no load condition.

LED signaling of “beep” codes

Description

The 5070 has bicolor LEDs (CR3 green and CR5 amber) that are used by the BIOS to indicate the BIOS processing state.

Immediately after the 5070 powers on, the CR5 amber LED is on and the CR3 green LED is off. Once the card boots, the CR5 amber LED turns off and the CR3 green LED is on.

If the BIOS finds an error during the power on self test (POST) the CR5 amber LED is flashed in a pattern indicating the POST code failure. The visual beep codes are defined in Table 20–1.

Count the number of flashes in each of four sets. Subtract one from each set, the resulting number matches the POST error found in the Table 19–1.

For example:

Flash-Flash pause

Flash-Flash-Flash-Flash pause

Flash-Flash-Flash-Flash-Flash pause

Flash-Flash-Flash-Flash

Is counted as 2-4-5-4. After subtracting one from each set the result is 1-3-4-3. This is a failure of the first 64K of base RAM.

Table 20–1 BIOS beep codes

Port 80 Code	Beep Sequence	POST Routine Description
02h		Verify Real Mode
03h		Disable Non–Maskable Interrupt (NMI)
04h		Get CPU type
06h		Initialize system hardware
07h		Disable shadow and execute code from the ROM
08h		Initialize chipset with initial POST values
09h		Set IN POST flag
0Ah		Initialize CPU registers
0Bh		Enable CPU cache
0Ch		Initialize caches to initial POST values
OEh		Initialize I/O component
0Fh		Initialize the local bus IDE
10h		Initialize Power Management
11h		Load alternate registers with initial POST values
12h		Restore CPU control word during warm boot
13h		Initialize PCI Bus Mastering devices
14h		Initialize keyboard controller
16h	1–2–2–3	BIOS ROM checksum
17h		Initialize cache before memory Auto size
18h		8254 timer initialization
1Ah		8237 DMA controller initialization
1Ch		Reset Programmable Interrupt Controller
20h	1–3–1–1	Test DRAM refresh
22h	1–3–1–3	Test 8742 Keyboard Controller
24h		Set ES segment register to 4 GB
28h		Auto size DRAM
29h		Initialize POST memory manager
2Ah		Clear 512 KB base RAM
2Ch	1–3–4–1	ROM failure on address line xxxx
2Eh	1–3–4–3	RAM failure on data bits xxxx of low byte of memory bus
2Fh		Enable cache before system BIOS shadow
32h		Test CPU bus–clock frequency
33h		Initialize Phoenix Dispatch Manager
36h		Warm start shutdown
38h		Shadow system BIOS ROM
3Ah		Auto size cache
3Ch		Advanced configuration of chipset registers
3Dh		Load alternate registers with CMOS values
41h		Initialize extended memory for ROMPilot
42h		Initialize interrupt vectors
45h		POST device initialization
46h	2–1–2–3	Check ROM copyright notice
47h		Initialize I20 support
48h		Check video configuration against CMOS
49h		Initialize PCI bus and devices
4Ah		Initialize all video adapters in system
4Bh		QuietBoot start (optional)
4Ch		Shadow video BIOS ROM
4Eh		Display BIOS copyright notice
4Fh		Initialize MultiBoot
50h		Display CPU type and speed
51h		Initialize EISA board
52h		Test keyboard
54h		Set key click if enabled

55h		Enable USB devices
58h	2–2–3–1	Test for unexpected interrupts
59h		Initialize POST display service
5Ah		Display prompt “Press F2 to enter Setup”
5Bh		Disable CPU cache
5Ch		Test RAM between 512 and 640 KB
60h		Test extended memory
62h		Test extended memory address lines
64h		Jump to UserPatch1
66h		Configure advanced cache registers
67h		Initialize Multi Processor APIC
68h		Enable external and CPU caches
69h		Setup System Management Made (SMM) area
6Ah		Display external L2 cache size
6Bh		Load custom defaults (optional)
6Ch		Display shadow–area message
6Eh		Display possible high address for UMB recovery
70h		Display error messages
72h		Check for configuration errors
76h		Check for keyboard errors
7Ch		Set up hardware interrupt vectors
7Dh		Initialize Intelligent System Monitoring
7Eh		Initialize coprocessor if present
80h		Disable onboard Super I/O ports and IRQs
81h		Late POST device initialization
82h		Detect and install external RS232 ports
83h		Configure non–MCD IDE controllers
84h		Detect and install external parallel ports
85h		Initialize PC–compatible PnP ISA devices
86h		Re–initialize onboard I/O ports
87h		Configure Motherboard Configurable Devices (optional)
88h		Initialize BIOS Data Area
89h		Enable Non–Maskable Interrupts (NMIs)
8Ah		Initialize Extended BIOS Data Area
8Bh		Test and initialize PS-2 mouse
8Ch		Initialize floppy controller
8Fh		Determine number of ATA drives (optional)
90h		Initialize hard–disk controllers
91h		Initialize local–bus hard–disk controllers
92h		Jump to UserPatch2
93h		Build MPTABLE for multi–processor boards
95h		Install CD-ROM for boot
96h		Clear huge ES segment register
97h		Fix up Multi Processor table
98h	1–2	Search for option ROMs. One long, two short beeps on checksum failure.
99h		Check for SMART drive (optional)
9Ah		Shadow options ROMs
9Ch		Set up Power Management
9Dh		Initialize security engines (optional)
9Eh		Enable hardware interrupts
9Fh		Determine number of ATA and SCSI drives
A0h		Set time of day
A2h		Check key lock
A4h		Initialize typematic rate
A8h		Erase F2 prompt
Aah		Scan for F2 keystroke
Ach		Enter Setup
Aeh		Clear Boot flag

B0h		Check for errors
B1h		Inform ROMPilot about the end of POST
B2h		POST done – prepare to boot operating system
B4h	1	One short beep before boot
B5h		Terminate QuietBoot (optional)
B6h		Check password (optional)
B7h		Initialize ACPI BIOS
B9h		Prepare Boot
Bah		Initialize SMBIOS
BBh		Initialize PnP Option ROMs
BCh		Clear parity checkers
BDh		Display MultiBoot menu
Beh		Clear screen (optional)
BFh		Check virus and backup reminders
C0h		Try to boot with Int 19h
C1h		Initialize POST Error Manager (PEM)
C2h		Initialize error logging
C3h		Initialize error display function
C4h		Initialize system error handler
C5h		PnP dual CMOS (optional)
C6h		Initialize note dock (optional)
C7h		Initialize note dock late
C8h		Force check (optional)
C9h		Extended checksum (optional)
Cah		Redirect Int15h to enable remote keyboard
CBh		Redirect Int 13h to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk
CCh		Redirect Int 10h to enable remote serial video
CDh		Re–map I/O and memory for PCMCIA
Ceh		Initialize digitizer and display message
D2h		Unknown interrupt
		The following are for boot block in Flash ROM
E0h		Initialize the chipset
E1h		Initialize the bridge
E2h		Initialize the CPU
E3h		Initialize system timer
E4h		Initialize system I/O
E5h		Check force recovery boot
E6h		Checksum BIOS ROM
E7h		Go to BIOS
E8h		Set Huge Segment
E9h		Initialize Multi Processor
Eah		Initialize OEM special code
Ebh		Initialize PIC and DMA
Ech		Initialize Memory type
Edh		Initialize Memory size
Eeh		Shadow Boot Block
Efh		System memory test
F0h		Initialize interrupt vectors
F1h		Initialize Run Time Clock
F2h		Initialize video
F3h		Initialize System Management Manager
F4h		Output one beep
F5h		Clear Huge Segment
F6h		Boot to Mini DOS
F7h		Boot to full DOS

Technical assistance

Carefully recheck your system before calling Technical Support. Run as many tests as possible; the more information you can provide, the easier it will be for Technical Support staff to help you solve the problem. For additional technical assistance, try the following:

Technical Support telephone: 303–426–4521

E–mail Technical Support: support1@octagonsystems.com

Applications Notes (via web):
http://www.octagonsystems.com/Solutions/appnotes.html

FAQ (via web): http://www.octagonsystems.com/pages/faqs.html

Overview: *Section 4 – Appendices*

Section 4 contains a series of appendices which provides additional information about the 5070.

Appendix A:	Technical data
Appendix B:	Software utilities
Appendix C:	Accessories

Appendix A: 5070 technical data

Technical specifications

CPU

586 128 MHz

SYSCLK

33, 50, 66, 99, 100, or 128 MHz (jumper selectable)

BIOS

AT compatible with industrial extensions

SDRAM

32MB surface mount SDRAM supplied

Solid–state disk

2 MB flash supplied
OEM option – 4 or 8 MB

Floppy drive

Floppy drive support with on–card floppy drive controller. Accessed via 34–pin connector on HDC-18-HDD/FDD drive cable.

Hard drive

EIDE hard drive support with on–card hard drive controller and BIOS. Accessed via 34–pin connector on HDC-18-HDD/FDD drive cable.

Video

Supports CRT, LCD and EL displays up to 1024 x 768 x 16 bpp (1280 x 1024 on some selected displays).

Ethernet

10/100 BaseT

Serial I/O

COM1, COM2 - 16C550 compatible
RS-232, RS-422, and RS-485 supported on both ports

Digital I/O

Two input and two output differential lines, optically isolated

USB

Two USB ports

Parallel port

LPT1 is PC compatible with multifunctional capability

Keyboard and Mouse ports

PS-2 compatible

Watchdog timer

Time–out is from 3 μ seconds to 2 seconds, software enabled and strobed. Enhanced INT17 function calls increase timeout.

Bus mastering

Bus mastering is not supported

DOS

ROM–DOS 7.1 included

Other operating system

Compatible with Windows NT, Windows 98, Windows CE, Linux, and QNX

Power requirements

5V ±0.25V

128 MHz operation:	1250 mA typical
Standby:	1170 mA typical
Suspend:	580 mA typical
33 MHz operation:	840 mA typical
Standby:	680 mA typical
Suspend:	500 mA typical

The power supply for the 5070 must meet the startup risetime requirements specified in the ATX Power Design Guide, version 1.1, section 3.3.5. This assures that all the circuitry on the 5070 sequences properly and avoids system lockup.

Environmental specifications

–40° to 85°C, operating, 99 MHz and below
–40° to 70°C, operating, 128 MHz
–55° to 90° C, nonoperating
RH 5% to 95%, noncondensing

Size

PCB size 4.5” x 4.9” x .92”

Weight

4.4 oz.

Mating connectors

J1, J200 – 80-pin connector
Hirose #CL572-0677-7

J2 – Ethernet — Standard Ethernet interface

J3 – Z-Tag — Z-Tag dongle

J4 – Flat panel — 50 pin shrouded header
Samtec -#TCSD-25-01-N

J5 – VGA CRT — 16-pin shrouded
AMP - #746288-3 receptacle
#499252-8 strain relief

J6 – PC/104 interface
For 8–bit – Samtec #ESQ–132–14–G–D
For 16–bit – Samtec #ESQ–120–14–G–D

J7 – Power — 4-pin header
Molex - 22–01–3047, female housing 4 position
08–52–0123, crimp to wire terminal

HDC-18-HDD/FDD Drive Cable – Octagon part #6239
80-pin – Hirose # CL572-0677-7
IDE – 40-pin hard drive connector, 3M #3417-7600
Floppy – 34-pin floppy connector, 3M #3414-7600
Stereo jacks – not used on 5070

HDC-18-MPC-Multiport cable – Octagon part #6240
80-pin – Hirose #CL572-0677-7
LPTA – 25-pin D-type female, AMP #747052-2
LPTB – 26-pin IDC-type female, AMP# 746288-6
COM1/2 – 9-pin D-type male, AMP #747306-4
USB1/2 – female panel mount, Assman #AK674
Keyboard – PS-2, 6-pin mini-DIN, Singatron #2MJ-20416000
Mouse – PS-2, 6-pin mini-DIN, Singatron #2MJ-20416000
Battery – 4-pin connector, AMP #644861-4
Speaker – 8 ohm, CUI Stack GC0351M
Reset switch – SPST MOM switch, C&K #8532MZQE2

Maps

Table A–1 5070 DMA map

Channel	Description
Channel 0	Reserved for bus memory refresh
Channel 1	Reserved for ECP parallel port
Channel 2	Floppy drive interface
Channel 3	IDE interface
Channel 4	Slave
Channel 5	Not available
Channel 6	Not available
Channel 7	Not available

Table A–2 5070 I/O map

Hex range	Function
000h to 0A7h	System I/O
0A8h to 0AFh	System I/O
0B0h to 0FFh	System I/O
100h to 1Efh	Off card
1F0h to 1F7h	IDE controller
208h to 20Fh	System I/O
218h to 21Ch	ZF Micro (reserved)
278h to 27Bh	Optional LPT1
2E8h to 2EFh	Optional COM2
2F8h to 2FFh	COM2 (default)
320h to 327h	System I/O
328h to 32Fh	System I/O
340h to 377h	System I/O
378h to 37Bh	LPT1 (default)
3E8h to 3EFh	Optional COM1
3F0h to 3F7h	Floppy controller
3F8h to 3FFh	COM1 (default)
8100h to 81FFh	Digital I/O

Table A–3 5070 interrupt map

IRQ	Default Device	Alternate
IRQ0	System Timer	
IRQ1	Keyboard	
IRQ2	Cascade to IRQ9	
IRQ3	COM2	PC/104
IRQ4	COM1	PC/104
IRQ5	Available	PC/104
IRQ6	Floppy	PC/104
IRQ7	LPT1	PC/104
IRQ8	RTC alarm	
IRQ9	Cascade from IRQ2	PC/104
IRQ10	Ethernet	PC/104 (also Bus IRQ3 via jumper
IRQ11	System USB	PC/104 (also Bus IRQ4 via jumper
IRQ12	Mouse	PC/104
IRQ13	Reserved for FPU	
IRQ14	Primary IDE controller	PC/104
IRQ15	Available	PC/104

Table A–4 5070 memory map

Address	Size	Description	Shadowing
0000H to 9FFFH	640KB	System SDRAM memory	
A0000H to BFFFH	128KB	Video memory	
C0000H to C9FFFH	40KB	Video BIOS shadow region	Setup option On when “V” is present
CA000H to CDFFFH	16KB	Reserved	
CE000H to CFFFFH	8KB	SSD window area (enabled when Flash FX is in use)	SSD1 window when X jumper installed
D0000H to D7FFFH	32KB	Off card memory	Setup option
D8000H to DFFFFH	32KB	INT 17 and FlashFX extended BIOS area	On when X jumper installed
E0000H to FFFFFH	128KB	System BIOS Area (Phoenix)	ENABLED
10000H to 1FFFFFFH	31 MB	System extended SDRAM memory	

Jumper settings by function

Table A–5 Flat panel jumpers: W1, W2, W7

W1, W2, W7 – Flat Panel		
Function	**Jumper**	**Description**
SHFCLK polarity	W1[1-3]	Inverted polarity
	W1[2-4]*	Normal polarity
Flat panel voltage	W2[1-2]*	5V panel
	W2[2-4]	3V panel
Latch pulse /	W7[2-4]*	Route latch pulse to J4, pin 8
Blank pulse routing	W7[1-3]	Route blank pulse to J4, pin 8
Back-light voltage	W7[7-9]	Route 12V to J4, pin 2
select	W7[8-10]*	Route 5V to J4, pin 2

* = default

Table A–6 CompactFlash configuration jumpers: W5, W6

W5, W6 – CompactFlash	
Configuration	**Jumper**
Master	W6[1–3]
Slave	W6[1–2]*
5V	W5[1–2]*
3V	W5[3–4]

* = default

Table A–7 CPU clock speed jumpers: W9, W12, W13

W9, W12, W13 – CPU Clock Speed			
Clock speed	**W9**	**W12**	**W13**
33 MHz	None	[9-10][11-12]	[1-2]
50 MHz	[1-2][3-4]	[9-10][11-12]	[1-2]
66 MHz	None	[11-12]	[1-2]
99 MHz	None	None	[1-2]
100 MHz	[1-2][3-4]	[11-12]	[1-3]
128 MHz*	[1-2]*	[11-12]*	[1-3]*

*** = default**

Table A–8 5070 COM port jumpers: W3, W8, W10, W11

W3, W8, W10, W11 – COM Ports		
COM Port	**Communication Mode**	**Jumper Settings**
COM1	RS–232C*	W3[4–6][10–12]* W10[1–2]* W11[1–2][4–6][5–7]*
	RS–422 no termination	W3[4–6][10–12] W10[1–3] W11[1–2][4–6][5–7]
	RS–422 with termination	W3[2–4][8–10] W10[1–3] W11[1–2][4–6][5–7]
	RS–485 no termination	W3[4–6][10–12] W10[2–4] W11[1–2][4–6][5–7]
	RS–485 with termination	W3[4–6][10–12] W10[2–4] W11[1–3][7–9][8–10]
COM2	RS–232C*	W3[3–5][9–11]* W8[1–2][4–6][5–7]* W10[7–8]*
	RS–422 no termination	W3[3–5][9–11] W8[1–2][4–6][5–7] W10[7–9]
	RS–422 with termination	W3[1-3][7-9] W8[1–2][4–6][5–7] W10[7–9]
	RS–485 no termination	W3[3–5][9–11] W8[1–2][4–6][5–7] W10[8-10]
	RS–485 with termination	W3[3–5][9–11] W8[1–3][7-9][8–10] W10[8-10]

* = default jumper installed

Table A–9 System configuration jumper: W12

W12 – System Configuration		
Label	**Description**	**W12**
S	System parameters option jumper: Installed = enable User Setup options* Removed = enable BIOS Setup default	[1–2]*
X	BIOS extension enable Installed = enable extended BIOS* required to boot from on-board flash required to use INT17 calls Removed = disables extended BIOS	[3–4]*
V	Video jumper: Installed = enable on-card video* Removed = disable on-card video	[5–6]*
U	User jumper	[7–8]*
0	CPU clock jumper – see CPU clock speed table	[9–10]
1	CPU clock jumper – see CPU clock speed table	[11–12]*
R	Reserved	[13–14]
Z	Reserved	[15–16]

*** = default**

Table A–10 Interrupt routing jumper: W14

W14 – IRQ interrupt routing		
Source	**Destination**	**W14**
Bus IRQ3	IRQ3	[5-7]*
	IRQ10	[3-5]
Bus IRQ4	IRQ4	[6-8]*
	IRQ11	[4-6]

*** = default**

Jumper settings by jumper designation

Table A–11 W1 – SHFCLK polarity

W1 – SHFCLK polarity	
Setting	**Function**
[1-3]	Inverted
[2-4]*	Non-inverted

*** = default**

Table A–12 W2 – Flat panel voltage select

W2 – Flat panel voltage select	
Setting	**Function**
[1-2]	5 volt panel
[2-4]*	3 volt panel

*** = default**

Table A–13 W3 – COM1/2, RS-422 line termination select

W3 – COM1/2, RS-422 line termination select	
Setting	**Function**
[4-6][10-12]*	COM1 – RS-232 (no termination)
[2-4][8-10]	COM1 – RS-422 100Ω termination
[3-5][9-11]*	COM2 – RS-232 (no termination)
[1-3][7-9]	COM2 – RS-422 100Ω termination

*** = default**

Table A–14 W4 – Factory jumper

W4 – Factory jumper	
Setting	**Function**
[1-3]*	Factory use only –DO NOT CHANGE

*** = default**

Table A–15 W5 – CompactFlash operating voltage select

W5 – CompactFlash operating voltage select	
Setting	**Function**
[1-2]*	5 volts
[3-4]	3 volts

*** = default**

Table A–16 W6 – CompactFlash Master/Slave select

W6 – CompactFlash Master/Slave select	
Setting	**Function**
[1-2]*	CompactFlash is slave device
[1-3]	CompactFlash is master device

*** = default**

Table A–17 W7 – Flat panel Safe voltage/LP or M select

W7 – Flat panel Safe voltage/LP or M select	
Setting	**Function**
[2-4]*	LP (route latch pulse to J4, pin 8)
[1-3]	M (route blank pulse to J4, pin 8)
[7-9]	12 volt safe
[8-10]*	5 volt safe

*** = default**

Table A–18 W8 – COM2, RS-485 line termination

W8 – COM2, RS-485 line termination	
Setting	**Function**
[1-2][4-6][5-7]*	RS-232 (no termination)
[1-3][7-9]	XMIT / RCV connected for RS-485
[1-3][7-9][8-10]	Connected and terminated 100Ω

*** = default**

Table A–19 W9 – SYSCLK frequency

W9 – SYSCLK frequency	
Setting	**Function**
[1-2]*	64 MHz
[3-4]	66 MHz
[1-2][3-4]	50 MHz
No jumpers	33 MHz

*** = default**

Table A–20 W10 – COM1/2, RS-232/422/485 select

W10 – COM1/2, RS-232/422/485 select	
Setting	**Function**
[1-2]*	COM1 – RS-232
[1-3]	COM1 – RS-422
[2-4]	COM1 – RS-485
[7-8]*	COM2 – RS-232
[7-9]	COM2 – RS-422
[8-10]	COM2 – RS-485

*** = default**

Table A–21 W11 – COM1 RS-485 line termination

W11 – COM1 RS-485 line termination	
Setting	**Function**
[1-2][4-6][5-7]*	RS-232 (no termination)
[1-3][7-9]	XMIT / RCV connected for RS-485
[1-3][7-9][8-10]	Connected and terminated 100Ω

*** = default**

Table A–22 W12 – User jumpers/Clock multiplier

W12 – User jumpers/Clock multiplier	
Setting	**Function**
[1-2]*	"S" – User Setup jumper
[3-4]*	"X" – Enable extended BIOS jumper
[5-6]*	"V" – Enable Video jumper
[7-8]*	"U" – User jumper
[9-10]	INVALID - DO NOT USE
[11-12]*	SYSCLK X2
[9-10][11-12]	SYSCLK X1
No jumpers	SYSCLK X3
[13-14]	Reserved for factory use
[15-16]	Reserved for factory use

*** = default**

Table A–23 W13 – CPU core voltage select

W13 – CPU core voltage select	
Setting	**Function**
[1-3]*	2.7 volts (CPU clock speeds 100 MHz and above)
[1-2]	2.2 volts (CPU clock speeds below 100 MHz)

*** = default**

Table A–24 W14 – IRQ routing

W14 – IRQ routing	
Setting	**Function**
[5-7]*	BUS IRQ 3 to CPU IRQ3
[3-5]	BUS IRQ 3 to CPU IRQ10
[6-8]*	BUS IRQ 4 to CPU IRQ4
[4-6]	BUS IRQ 4 to CPU IRQ11

*** = default**

Connector pin-outs

The following tables show the pin-outs for the connectors on the 5070 CPU control card.

Table A–25 Ethernet connector: J2

J2 – Ethernet connector			
Pin#	**Pin name**	**Pin name**	**Pin#**
1	Shield	Shield	8
2	Link LED anode	NC	9
3	Link LED cathode	Transmit +	10
4	NC	Transmit –	11
5	NC	Activity LED cathode	12
6	Receive +	Activity LED anode	13
7	Receive –	Shield	14

Table A–26 CRT connector: J5

J5 – CRT connector			
Pin#	**Pin name**	**Pin name**	**Pin#**
1	**Red**	Green	2
3	Blue	NC	4
5	GND	GND	6
7	GND	GND	8
9	+5V Safe	GND	10
11	NC	DDC SDA	12
13	HSYNC	VSYNC	14
15	DDC CLK	GND	16

Table A–27 Power connector: J7

J7 - Power connector	
Pin#	**Pin name**
1	+5V
2	GND
3	+12V
4	– 12V

Table A–28 Flat panel connector: J4

J4 – Flat panel connector			
Pin#	**Pin name**	**Pin name**	**Pin#**
1	**Flat panel VCC**	Back-light VCC	2
3	**Flat panel VCC**	GND	4
5	ENAVEE	PD24	6
7	M	LP/M	8
9	PD25	LP	10
11	FLM	GND	12
13	BSHFCLK	PD26	14
15	PD0	PD1	16
17	PD27	PD2	18
19	PD3	PD28	20
21	PD4	PD5	22
23	GND	PD6	24
25	PD7	PD29	26
27	PD8	PD9	28
29	PD30	PD10	30
31	PD11	PD31	32
33	PD12	PD13	34
35	GND	PD14	36
37	PD15	PD32	38
39	PD16	PD17	40
41	PD33	PD18	42
43	PD19	PD34	44
45	PD20	PD21	46
47	PD35	PD22	48
49	PD23	GND	50

Table A–29 Digital I/O connector: J11

J11 – Digital I/O connector		
Function	**Pins**	**Name**
Input 0 (GPIO0)	1	DIN0 –
	2	DIN0 +
Input 1 (GPIO1)	3	DIN1 –
	4	DIN1 +
Output 0 (GPIO2)	5	DOUT0 –
	6	DOUT0 +
Output 1 (GPIO3)	7	DOUT1 –
	8	DOUT1 +
	9	Not used
	10	Not used

Table A–30 IDE / floppy connector: J1

J1 – IDE / floppy connector			
Side A Pin#	**Pin name**	**Pin name**	**Side B Pin#**
1	**IDE RST***	**GND**	**1**
2	**GND**	NC	2
3	IDED7	GND	3
4	IDED8	NC	4
5	IDED6	GND	5
6	IDED9	NC	6
7	IDED5	GND	7
8	IDED10	INDEX*	8
9	IDED4	GND	9
10	IDED11	MTR0*	10
11	IDED3	GND	11
12	IDED12	DR0*	12
13	IDED2	GND	13
14	IDED13	DR0*	14
15	IDED1	GND	15
16	IDED14	DIR*	16
17	IDED0	GND	17
18	IDED15	STEP*	18
19	**GND**	GND	19
20	NC	WDATA*	20
21	IDE	GND	21
22	**GND**	WGATE*	22
23	IDEIOW*	GND	23
24	**GND**	TRK0*	24
25	IDEIOR	GND	25
26	**GND**	WP*	26
27	IDEIORDY	GND	27
28	NC	RDATA*	28
29	IDEDACK*	GND	29
30	**GND**	RDSEL*	30
31	IRQ14	GND	31
32	NC	DSKCHG*	32
33	IDEA1	**GND**	33
34	PDIAG	NC	34
35	IDEA0	NC	35
36	IDEA2	NC	36
37	IDESC0*	NC	37
38	IDECS1*	NC	38
39	ACT LED	NC	39
40	**GND**	NC	40

*** = indicates an inverted signal**

Table A–31 Multipurpose I/O connector: J200

J200 – Multipurpose I/O			
Side A Pin#	**Pin name**	**Pin name**	**Side B Pin#**
1	**OSTB***	**DCD1***	**1**
2	**AFD***	DSR1*	2
3	PD0	RX1	3
4	ERR*	RTS1*	4
5	PD1	TS1	5
6	INIT*	CTS1*	6
7	PD2	DTR1*	7
8	SLIN*	RI1*	8
9	PD3	NC	9
10	GND	**DCD2***	10
11	PD4	DSR2*	11
12	GND	RX2	12
13	PD5	RTS2*	13
14	GND	TS2	14
15	PD6	CTS2*	15
16	GND	DTR2*	16
17	PD7	RI2*	17
18	GND	NC	18
19	**ACK***	DIN0–	19
20	GND	DIN0+	20
21	BUSY	DIN1–	21
22	**GND**	DIN1+	22
23	PE	DOUT0–	23
24	**GND**	DOUT0+	24
25	SLCT	DOUT1–	25
26	**GND**	DOUT0+	26
27	BATTERY +	USBPWR0	27
28	5V SAFE	USBD0–	28
29	SPEAKER +	USBD0+	29
30	**GND**	USBD0GND	30
31	RESET	USBD0GND	31
32	GND	USBPWR1	32
33	MDATA	USBD1–	33
34	5V SAFE	USBD1+	34
35	MCLK	USBD1GND	35
36	GND	USBD1GND	36
37	NC	KBCLK	37
38	NC	5V SAFE	38
39	NC	KBDATA	39
40	**NC**	GND	40

*** = indicates an inverted signal**

Table A–32 CompactFlash connector: J201

J201 – CompactFlash connector			
Pin#	**Pin name**	**Pin name**	**Pin#**
1	GND	IDED3	2
3	IDED4	IDED5	4
5	IDED6	IDED7	6
7	IDECS0*	GND	8
9	GND	GND	10
11	GND	GND	12
13	+5V	GND	14
15	GND	GND	16
17	GND	IDEA2	18
19	IDEA1	IDEA0	20
21	IDED0	IDED1	22
23	IDED2	NC	24
25	NC	NC	26
27	IDED11	IDED12	28
29	IDED13	IDED14	30
31	IDED15	IDECS1*	32
33	NC	IDEIOR*	34
35	IDEIOW	+5V	36
37	IRQ14	+5V	38
39	M/S select	NC	40
41	RST*	IDEIORDY	42
43	NC	+5V	44
45	ACTLED	PDIAG	46
47	IDED8	IDED9	48
49	IDED10	GND	50

*** = indicates an inverted signal**

Table A–33 ZTAG connector: J3

J3 – JTAG connector			
Pin#	**Pin name**	**Pin name**	**Pin#**
1	DSKCHG*	PWRGOOD	2
3	DR0*	GND	4
5	VCC3	SA23	6
7	STEP*	RDATA*	8
9	VCC3	RDSEL*	10
11	WP*	WDATA*	12
13	DIR*	TRK0*	14

* = indicates an inverted signal

Table A–34 PC/104 connector: J6

J6 – PC/104 connector				
Pin#	**Row A**	**Row B**	**Row C**	**Row D**
0	-----	-----	GND	GND
1	IOCHK*	GND	SBHE*	MCS16*
2	SD7	RSTDRV	LA23	IOCS16*
3	SD6	+5V SAFE	LA22	IRQ10
4	SD5	IRQ9	LA21	IRQ11
5	SD4	NC	LA20	IRQ12
6	SD3	DRQ2	LA19	IRQ15
7	SD2	-12V	LA18	IRQ14
8	SD1	ZWS*	LA17	DACK0*
9	SD0	+12V	MEMR*	DRQ0
10	IOCHRDY	KEY	MEMW*	DACK5*
11	AEN	SMEMW*	SD8	DRQ5
12	SA19	SMEMR*	SD9	DACK6*
13	SA18	IOW*	SD10	DRQ6
14	SA17	IOR*	SD11	DACK7*
15	SA16	DACK3*	SD12	DRQ7
16	SA15	DRQ3	SD13	+5V
17	SA14	DACK1*	SD14	MASTER*
18	SA13	DRQ1	SD15	GND
19	SA12	REF*	Key	GND
20	SA11	SYSCLK	-----	-----
21	SA10	IRQ7	-----	-----
22	SA9	IRQ6	-----	-----
23	SA8	IRQ5	-----	-----
24	SA7	IRQ4	-----	-----
25	SA6	IRQ3	-----	-----
26	SA5	DACK2*	-----	-----
27	SA4	TC	-----	-----
28	SA3	BALE	-----	-----
29	SA2	+5V SAFE	-----	-----
30	SA1	OSC	-----	-----
31	SA0	GND	-----	-----
32	GND	GND	-----	-----

*** = indicates an inverted signal**

Table A–35 ISA bus connector: P1

P1 – ISA bus connector		
Pin#	**Row A**	**Row B**
1	IOCHK*	GND
2	SD7	RSTDRV
3	SD6	+5V SAFE
4	SD5	IRQ9
5	SD4	NC
6	SD3	DRQ2
7	SD2	-12V
8	SD1	ZWS*
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW*
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3*
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REF*
20	SA11	SYSCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	TC
28	SA3	BALE
29	SA2	+5V SAFE
30	SA1	OSC
31	SA0	GND

* = indicates an inverted signal

Appendix B: Software utilities

Introduction

This chapter describes the utilities listed below. The drivers and utilities are in a self-extracting zip file, located at the Octagon Systems web site on the 5070 product page. Download this file to a separate directory on your hard drive, then double click on it to extract the files.

Support commands

- FXCHK.EXE
- FXDOS.SYS
- FXFMT.EXE
- FXINFO.EXE
- FXRECLM.EXE
- FXREMNT.EXE
- GETBIOS.EXE
- GETIMG.EXE
- GETIMGH.EXE
- I17HNDLR.EXE
- LPT1CON.COM
- PGMBIOS.EXE
- PGMIMG.EXE
- PGMIMGH.EXE
- REMDISK.EXE
- REMQUIT.COM
- REMSERVE.EXE
- RESET.COM
- SETSSD.EXE
- TRANSFER.EXE

Support device drivers

- HIMEM.SYS
- FXDOS.SYS
- VDISK.SYS

FXCHK.EXE

Purpose

This support command checks the FlashFX flash media for bad spare units.

Syntax

FXCHK *drive*

Parameter

drive specifies the drive letter to check.

Remarks

FlashFX usually keeps at least one spare unit so that if a flash device develops a bad block, the spare unit takes its place.

FXDOS.SYS

Purpose

An alternate way to access the on–board SSD. By using this driver, it is possible to free up the address area at CE0000h–CFFFFFh by the “X” jumper.

Syntax

FXDOS.SYS

Remarks

This command will not allow booting from an SSD.

FXFMT.EXE

Purpose

This support command formats a FlashFX flash disk.

Syntax

```
FXFMT <Drive> { Options }
```

Parameter

<Drive> Specifies the drive to format. This can be either a drive letter i.e. C: (if the drive has already been formatted) or drive descriptor i.e. 80 if the drive is not yet formatted.

Drive descriptors are of the format 8x where x is the number of the hard drive beginning with 0. For example 80 is the first hard drive, 81 is the second hard drive.

/C Don't confirm before formatting.
/B Force a system reboot and do NOT confirm.
/M Do NOT write an MBR to the media.
/Px[K] Preserves (x) bytes from start of flash.

Default is current format, Max is 960.

/Sx Format with x number of spare units.

Default is current format.

/Tx[KM] Formats (x) bytes of flash.

Default is current format, Max is media size.

/Qx Reserves (x) percent of the media size for cushion.

Default is 2%. Min is 1, Max is 25.

/Vx Place volume label "x" on disk.
/Dx Where (x) is the number of root directory entries.

Default is 240. Min is 16, Max is 512.

Remarks

The previous contents of the drive are lost.

After reformatting, the drive can be SYSed to become bootable which requires another boot device with an installed O/S (such as a floppy).

FXFMT will not affect IDE hard drives.

FXFMT uses the current SETSSD information. This defines the order and the amount of space reserved for the BIOS. If the amount of space reserved for the BIOS changes, FXFMT must be re–executed.

See also

SETSSD.EXE.

FXINFO.EXE

Purpose

This support command displays information about a FlashFX disk.

Syntax

FXINFO *drive*

Parameter

drive specifies the drive letter to show.

Remarks

FlashFX keeps a host of information about each drive.

FXRECLM.EXE

Purpose

This support command removes outdated versions of files and reorganizes files on the FlashFX flash media, to improve access time.

Syntax

```
FXRECLM drive count
```

Parameter

drive specifies the drive letter to reclaim.
count specifies the maximum number of operations to perform.

Remarks

FlashFX will reclaim the drive specified, up to the maximum number of operations specified. This may be helpful if files are constantly changed or deleted.

FXREMNT.EXE

Purpose

This support command forces a remount of the FlashFX flash media.

Syntax

```
FXREMNT drive
```

Parameter

drive specifies the drive letter to remount.

Remarks

FlashFX will remount the drive specified. This may be helpful if the drive was just re–programmed using PGMIMG and when a reboot of the system is not wanted.

GETBIOS.EXE

Purpose

This support command stores the BIOS information in a specific file.

Syntax

GETBIOS *filename*

Parameters

- *filename* specifies the output file for saving or programming.

GETIMG.EXE

Purpose 1

This support command captures an image of an SSD and places it into a local file.

Syntax 1

GETIMG SSD*x filename*

Purpose 2

This support command captures an image of an SSD and transfers it to a host PC running **GETIMGH**.

Syntax 2

GETIMG SSD*x* **/COM***x* [**/B***xx*] [**/I***xx*]

Purpose 3

This support command captures an image of an SSD and transfers it to a host PC running GETIMGH. A nonstandard serial port I/O address is used and the IRQ value must be specified. This syntax is required when in serial console mode.

Syntax 3

GETIMG SSD*x* **/U***xxxx* [**/B***xx*] [**/I***xx*]

Parameters

- **SSD***x* specifies the target SSD for file saving or programming. The variable *x* represents a value from 0 to 1.
- *filename* specifies the output file for saving or programming.

- **/COM***x* specifies the PC COM port for serial transfer where *x* represents a value from 1 to 4.
- **/U***xxxx* specifies the UART base address to use for serial transfer. The base address, 100–3FF, is in hexadecimal format.
- **/B***xx* specifies baud rate of transfer where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200). The default is 38400 when using a **/COM***x* switch.
- **/I***xx* specifies the interrupt to which the UART is connected. The value *x* represents a variable from 3 to 15.

Remarks

GETIMG compatibility:

The following devices share the same image and are interchangeable: Atmel's AT29C040 and AT29C040A; 512K EPROM; and 512K SRAM. Use an EPROM to make the drive read–only.

AMD, Intel, and Sharp flash memory are interchangeable. They are not interchangeable with Atmel devices.

GETIMGH.EXE

Purpose 1

This support command transfers an SSD image from a target PC running **GETIMG** and saves the image to a host file.

Note: This utility is not compatible with Windows XP, Windows NT, or Windows 2000.

Syntax 1

GETIMGH *filename* /**COM***x* [/**B***xx*] [/**I***xx*]

Purpose 2

This support command transfers an SSD image from a target PC running GETIMG and saves the image to a host file via a serial UART connection. A non–standard serial port address is used and the IRQ value must be specified.

Syntax 2

GETIMGH *filename* /**U***xxx* [/**B***xx*] /**I***xx*

Parameters

- *filename* specifies the output file for saving or programming and it also represents the host filename.

- **/COM***x* specifies the PC COM port for serial transfer. The variable *x* represents a value from 1 to 4.

- **/U***xxx* specifies the UART base address to use for serial transfer. The base address, 100–3FF, is in hexadecimal format.

- **/B***xx* specifies baud rate of transfer where *xx* can be (300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200)). The default is 38400 when using a **/COM***x* switch. If COM1 is the console, the baud rate defaults to the current console baud rate.

- **/I***xx* specifies the interrupt to which the UART. The variable *x* represents a value from 3 to 15.

See also

GETIMG.EXE for details on image compatibility.

HIMEM.SYS

Purpose

This device driver manages extended memory and the High Memory Area (HMA) in a 286, 386, or greater PS-2 system. HIMEM prevents programs from simultaneously using the same area of memory for two different purposes. HIMEM supports the eXtended Memory Specification (XMS) 2.0. HIMEM is installed as a device driver in CONFIG.SYS.

Syntax

DEVICE=[*d*:] [*path*] **HIMEM.SYS** [/machine:*n*]

Remarks

- The HIMEM driver can be used to allow ROM–DOS to run in High Memory.

- HIMEM supports a default of 32 handles.
- HIMEM should not be used with older versions of VDISK. Current versions of VDISK will use XMS memory if it is available.
- HIMEM recognizes PS-2 style machines A20 line control. HIMEM determines whether to use the PS-2 A20 control or the AT A20 control method automatically by calling INT15h, function C0h (get system configuration).
- The automatic detection can be overridden with the "/Machine:*n*" command line switch. Replacing "*n*" with 1 designates the PC AT A20 control method. Replacing "*n*" with 2 designates the PS-2 method.

Example 1

```
DEVICE=HIMEM.SYS
```

The above command installs the XMS device driver. Once this driver is installed, accessing the HMA and Extended Memory (XMS) areas are legal. The Extended Memory area can contain up to 2 GB of memory. Typical systems have 4, 8, or 16 MB XMS memory installed.

Example 2

```
DEVICE=HIMEM.SYS /machine:1
```

This example forces the use of the AT style A20 line control.

The HIMEM driver will fail to load if either the machine does not have memory above the 1 MB boundary or the BIOS does not provide support for it. It will also fail to load if another XMS manager has been previously installed .

I17HNDLR.EXE

Purpose

This support command is an alternate way to use the INT 17h functions when the extended BIOS area is disabled (i.e., the jumper at the "X" position is removed at W12. Also, use this support command to reprogram the extended BIOS area with another BIOS.

Syntax

I17HNDLR

Remarks

This command is used if the extended BIOS area (CE000–CFFFF) is not used. The I17HNDLR allows the system to use the INT 17h functions.

LPT1CON.COM

Purpose

This support command redirects the video to the LPT1 port.

Syntax

LPT1CON

Remarks

If you have a keypad and display board and an LCD display connected to the AUX I/O port, executing the DISPLAY.EXE and LPT1CON.COM programs allow you to use the display as the system console. You must reset your system to change the video to the original parameters.

PGMBIOS.EXE

Purpose

This support command programs a new system BIOS into the 5070.

Syntax

PGMBIOS [*filename*] [/y] [/**?**]

Parameters

- *filename* specifies the BIOS .DAT file to program into flash.
- **SSD***x* specifies the source SSD for BIOS. The variable *x* represents a value from 0 to 1.

- **SSD***y* specifies the target SSD for BIOS. The variable *y* represents a value from 0 to 1.
- **/?** requests a help menu.
- **/y** Disables “Are you sure?” confirmation message.

Example 1

To program the BIOSFILE.BIN files into the SSD1 BIOS area, enter:

```
PGMBIOS BIOSFILE.BIN SSD1
```

Example 2

To program the BIOS and extended BIOS from SSD0 to SSD1, enter:

```
PGMBIOS SSD0 SSD1
```

PGMIMG.EXE

Purpose 1

This support command programs a local file image to an SSD.

Syntax 1

PGMIMG SSD*x* filename

Purpose 2

This support command programs a local file image to an SSD and transfers it to a host PC running **PGMIMGH**.

Syntax 2

PGMIMG SSD*x* **/COM***x* [**B***xx*] [**I***xx*]

Purpose 3

This support command programs a local file image to an SSD and transfers it to a host PC running PGMIMGH. A nonstandard serial port I/O address is used and the IRQ value must be specified.

Syntax 3

PGMIMG SSD*x* /**U***xxxx* [**B***xx*] /**I***xx*

Parameters

- *filename* specifies the input file programming.
- **SSD***x* specifies the target SSD for image. The variable *x* represents a value from 0 to 1.
- **/COM***x* specifies the PC COM port for serial transfer. The variable *x* represents a value from 1 to 4.
- **/U***xxx* specifies the UART base address to use for serial transfer. The base address, 100–3FF is in hexadecimal format.
- **/B***xx* specifies baud rate of transfer where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200). The default is 38400 when using a **/COM***x* switch. If COM1 is the console, the baud rate defaults to the current console baud rate.
- **/I***xx* specifies the interrupt to which the UART base address is connected. The variable *x* represents a value from 3 to 15.

PGMIMGH.EXE

Purpose 1

This support command programs an image file from a target PC running PGMIMG.

Note: This utility is not compatible with Windows XP, Windows NT, or Windows 2000.

Syntax 1

PGMIMGH *filename* /**COM***x* [**/B***xx*] [/**I***xx*]

Purpose 2

This support command transfers an SSD image to a target computer via a serial UART connection and programs the image to an SSD. A non–standard serial port address is used and the IRQ must be specified.

Syntax 2

PGMIMGH *filename* /**U***xxx* [/**B***xx*] **/I***xx*

Parameters

- *filename* specifies the input file for programming and it also represents the host filename.

- **/COM***x* specifies the PC COM port for serial transfer. The variable x represents a value from 1 to 4.

- **/U***xxx* specifies the UART base address to use for serial transfer. The base address, 100–3FF is in hexadecimal format.

- **/B***xx* specifies baud rate of transfer where *b* can be (300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200)). The default is 38400 when using a **/COM***x* switch. If COM1 is the console, the baud rate defaults to the current console baud rate.

- **/I***xx* specifies the interrupt to which the UART base address is connected. The variable x represents a value from 3 to 15.

See also

See PGMIMG.EXE. See also, GETIMG for image compatibility

REMDISK.EXE

Purpose

This support command allows access to a disk drive on a remote system via a serial cable and standard PC style (8250 UART) serial port.

Syntax

REMDISK [**/U**] [**/?**] [**/B***nnnn*] [**+**] [**/COM***n*]

Parameters

- **/U** tells REMDISK to unload itself from memory, thereby disabling the new drive letter and freeing the memory occupied by REMDISK. The option can only be used when REMDISK is installed from the DOS command line. A Remote Disk installed via CONFIG.SYS cannot be unloaded.

- **/?** displays a short help screen for the REMDISK program. No other arguments are to be included on the command line when the **/?** is used.
- **/B***nnnn* selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115k. The default baud rate is 115k.
- **+** is an optional argument which specifies packet style transmission. This is recommended for any baud rates over 19200. The default for this option is to include the **+** for packet transmission.
- **COM***n* is an optional argument which selects the communication port. Available ports are 1 and 2. COM1 is the default port.

Note To use the Remote Disk, both the REMDISK and the REMSERV programs must be running on their respective systems. Both programs must use the same baud rate and packet or non–packet style transmission. It does not matter which program is installed first.

Remarks

In a Remote Disk setup, one system, the one that will share its drives, is termed the Server. The other system, the one that will access and use the remote drives, is called the Client. The serial ports on both systems must be connected via a null modem cable. A cabling diagram for a standard cable is shown below.

Run REMDISK.EXE on the Client system. This program creates a new drive letter for the Client. REMDISK will use the next available system drive letter. For example, if the last assigned drive was D:, REMDISK will create a drive E:. This drive acts in all ways just like any other drive, except for the fact that it requires the serial port to do its job.

REMDISK.EXE can be installed using a **DEVICE=** command in CONFIG.SYS or from the DOS prompt.

Example 1

To install the REMDISK program from CONFIG.SYS at 19200, on COM1, using packet style transmission, enter the following in CONFIG.SYS and then reboot the system (remember to include the full path to find REMDISK.EXE if not located in the root directory):

```
DEVICE=REMDISK.EXE /B19200 +
```

Example 2

To display a help screen for REMDISK, enter the following at the DOS prompt:

```
REMDISK /?
```

Example 3

To install REMDISK from the DOS prompt or from a batch file (like AUTOEXEC.BAT) at 9600 baud, without packet style transmission, on COM2, enter the following;

```
REMDISK /B9600 /COM2
```

Example 4

To unload the REMDISK installed from the batch file or the DOS prompt, type:

```
REMDISK /U
```

See also

REMSERV.EXE

REMQUIT.COM

Purpose

This support command cancels a REMSERV session on a remote system.

Syntax

```
REMQUIT
```

Remarks

Once a REMDISK/REMSERV connection is no longer needed, the REMQUIT command is used (on the same CPU running REMDISK) to cancel the REMSERV command. You may also press the ESC key if you have access to a local keyboard to the CPU running REMSERV.

See also

REMSERV.EXE, REMDISK.EXE

REMSERV.EXE

Purpose

This support command makes a single drive at a time on the server system available to the Client. The available drive can be changed at any time by quitting the REMSERV program and then running the program again with a new drive letter.

Syntax

REMSERV.EXE *d:* [**/B***nnnn*] [**+**] [**/COM***n*] [**/S**]

Parameters

- *d:* represents the letter of the drive that the Server will make available to the Client.
- **/B***nnnn* selects the baud rate for transmission. Available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115k. The default baud rate is 115k.
- **+** is an optional argument which specifies packet style transmission. This is recommended for any baud rates over 19200. The default for this option is to include the **+** for packet transmission.
- **COM***n* is an optional argument which selects the communication port. Available ports are 1 and 2. COM1 is the default port.
- **/S** instructs REMSERV to run silently, that is without any screen output.
- **/?** is an unlisted option which is used to print a short help screen for the REMSERV program. If the **/?** is used, the drive letter argument is omitted, for example: **REMSERV /?**

Example 1

To select drive B: as the available Server drive at 115K baud, packet style transmission, using COM1, you would enter the following:

```
REMSERV B:
```

Example 2

To set drive C: as the Server disk at 9600 baud, without packet style transmission, on COM2, you would enter the following:

```
REMSERV C: /B9600 /COM2
```

Note The Server program can be terminated at any time by pressing the <ESC> key. The Client can then no longer access the Server's drive until the REMSERV program is run again.

See also

REMDISK.EXE

RESET.COM

Purpose

This support command enables the watchdog timer and allows time–out to expire, thus restarting the system.

Syntax

```
RESET
```

Remarks

The RESET command also restarts all the expansion I/O cards on the bus. This differs from a <CTRL><ALT><DEL> reboot of the system which only restarts the system but not the expansion cards. The RESET button on the PC–520 also accomplishes the same thing as the RESET command.

SETSSD.EXE

Purpose

This support command sets the arrangement of the SSD FlashFX devices.

Syntax

```
SETSSD [SSDn] [/Before | /After] [/Nossd] [/?] [/D]
```

Parameters

SSD*n* specify which SSDs are enabled and the order in which the SSD will be placed.
/After indicates the SSDs are allocated after any hard drives and the hard drive will be the boot device.
/Before indicates the SSDs are allocated before any hard drives and the first SSD specified will be the boot device.
/NoSSD specifies no SSDs are to be allocated (default if either **/A** or **/B** used and no SSDs are specified) .
/D selects default settings. (**SSD0 SSD1 /Before**)
/SSDnBIOS– reserves BIOS area on SSDn (**+** enables)

Note Requires FXFMT after this option is changed.

/WPn+ write protects drive on SSDn (– enables writes)

Remarks

Defines the SSD device order used by the flash file system. If an SSD name is included on the command line, the SSD will be enabled and can therefore be detected and used by FlashFX. If more than one SSD is listed on the command line, the first SSD will be the first drive (usually C:) and the second SSD will be the second (usually D:).

When IDE hard drives are used, the **/BEFORE** and **/AFTER** switches allow the first SSD to appear before or after the hard drives. For example if two IDE drives and one flash device (SSD1) are defined by SETSSD, and the /BEFORE switch is used; SSD1 will be drive C: and the two IDE drives will be D: and E:. However if the /AFTER switch is used, the first IDE drive will be drive C:, SSD1 will be drive D: and the second IDE drive will be E:.

When other devices such as a DOC are used, the DOC extended BIOS may affect the /AFTER and /BEFORE and the "C: ONLY" setup option.

WARNING!

Use this option with caution!

/WPn can be used to disable writes to the part via the drivers. PGMIMG can still write to these devices.

The SETSSD device order can be overridden by removing the "S" jumper. When removed and when the system boots up, prompts will be asked as to which device is first and which is second. Then a prompt asking whether you wish this information to be saved. If the save prompt is answered "Y" the new information overrides the current SETSSD options. If the save prompt is answered "N" the new information is used but not saved. Since the SETSSD command shows the "saved" options not the temporary

"working" options, this may lead to confusion. It is therefore recommended to always "save" your options.

The device order is used by the FlashFX extended BIOS as well as the FXFMT and TESTOEM commands. Other commands such as FXCHK, FXINFO, FXRECLM etc. are indirectly affected as the device order is changed. The BIOS reserve option effects the FXFMT, PGMIMG, GETIMG commands.

See also

See the *SETSSD* section in the *Setup programs* chapter.

TRANSFER.EXE

Purpose

This support command transfers files to or from the 5070 over a serial port.

Syntax

TRANSFER *filepath* [/**S** | /**R**] [/**B***xxxx*] [/**V**] [/**COM***x*]

Parameters

- *filepath* specifies the file pathname to send or receive.
- **/S** specifies to send the file.
- **/R** specifies to receive the file. This is the default.
- **/B***xxxx* specifies baud rate of transfer where *xxxx* can be (300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200)). The default is 9600 when using a **/COM***x* switch. If COM1 is the console, the baud rate defaults to the current console baud rate.
- **/V** enables the display of “R” when a block (128 bytes) is received, or “T” when a block is transmitted.

Note Do not use **/V** when COM1 is the console device.

- **/COM***x* specifies the serial port to use where *x* represents a value from 1–4. The default is 1.

Example 1

To send a file named C:\MPC\DEMO\DEMO.EXE on the 5070 when using COM1 as the console, enter the following command:

```
TRANSFER D:DEMO.EXE
```

On the remote PC executing SmartLINK, press <ALT> <D>, type C:\MPC\DEMO\DEMO.EXE, and then press ENTER.

Example 2

To send a file named D:DEMO.BAS to the file C:\TEST.BAS on the remote PC when using COM1 as the console, enter the following on the 5070:

```
TRANSFER /S D:DEMO.BAS
```

On the remote PC executing SmartLINK, press <ALT> <U>, type C:\TEST.BAS, and then press <ENTER>.

Example 3

To send a file named C:\DEMO\DEMO.EXE from a remote PC to the file named D:\DEMO.EXE on the 5070 at 57600 baud with a serial cable from COM2 on the remote PC and COM1 on the 5070, enter the following command on the 5070:

```
TRANSFER D:\DEMO.EXE /R /V /B57600
```

Then enter the following command on the remote PC:

```
TRANSFER C:\DEMO\DEMO.EXE /S /V /COM2 /B57600
```

To receive a file named D:\MYAPP.EXE from the 5070 and name it C:\APPS\MYAPP2.EXE on the remote PC over a serial cable connected to COM1 on both systems at 9600 baud, enter the following command on the 5070:

```
TRANSFER D:\MYAPP.EXE /S
```

Then enter the following command on the remote PC:

```
TRANSFER C:\APPS\MYAPP2.EXE
```

Remarks

The TRANSFER command communicates with other XMODEM compatible file transfer programs.

The serial port on the 5070 requires a null modem adapter when connected to a serial port on the remote PC. See the *Serial ports* chapter more information.

The maximum baud rate is dependent on the processor speeds of the remote PC and the 5070.

The received file size is rounded up to the nearest 128 byte boundary.

See also

REMDISK.EXE, REMSERV.EXE

Appendix C: Accessories

5070 CPU card accessories

Table C–1 Cables and terminal board

Product	Description	Octagon p/n
HDC-18-MPC-Multiport cable	Multipurpose I/O cable	6240
HDC-18-HDD/FDD cable	Floppy / hard drive cable	6239
Cable, 6225 power	Power cable for 5070	5848
Null Modem Adapter, 9 Pin	9–pin to 9–pin	2470
VGA-12, Cable, Ribbon, 12”	VGA video cable	2776
STB–10	Terminal board, 10–position	

Table C–2 LCD displays and keypads

Product	Description	Octagon p/n
LCD–4 x 20	LCD display w/cable, 40 character	2783
LCD–4 x 40	LCD display w/cable, 80 character	2784
2010	LCD display/keypad interface	3909
KP–1	Keypad w/cable, 16–key, low cost	1218
KP-3	Keypad w/cable, 16–key	1737

Table C–3 Miscellaneous part numbers

Product	Description	Octagon p/n
AT battery	Calendar/clock battery backup	3186
CAMBASIC	Multitasking, industrial control programming language	4059

Warranty

Octagon Systems Corporation (Octagon), warrants that its standard hardware products will be free from defects in materials and workmanship under normal use and service for the current established warranty period. Octagon's obligation under this warranty shall not arise until Buyer returns the defective product, freight prepaid to Octagon's facility or another specified location. Octagon's only responsibility under this warranty is, at its option, to replace or repair, free of charge, any defective component part of such products.

Limitations on warranty

The warranty set forth above does not extend to and shall not apply to:

1. Products, including software, which have been repaired or altered by other than Octagon personnel, unless Buyer has properly altered or repaired the products in accordance with procedures previously approved in writing by Octagon.
2. Products which have been subject to power supply reversal, misuse, neglect, accident, or improper installation.
3. The design, capability, capacity, or suitability for use of the Software. Software is licensed on an "AS IS" basis without warranty.

The warranty and remedies set forth above are in lieu of all other warranties expressed or implied, oral or written, either in fact or by operation of law, statutory or otherwise, including warranties of merchantability and fitness for a particular purpose, which Octagon specifically disclaims. Octagon neither assumes nor authorizes any other liability in connection with the sale, installation or use of its products. Octagon shall have no liability for incidental or consequential damages of any kind arising out of the sale, delay in delivery, installation, or use of its products.

Service policy

1. If a product should fail during the warranty period, it will be repaired free of charge. For out of warranty repairs, the customer will be invoiced for repair charges at current standard labor and materials rates.
2. Customers that return products for repairs, within the warranty period, and the product is found to be free of defect, may be liable for the minimum current repair charge.

Returning a product for repair

Upon determining that repair services are required, the customer must:

1. Obtain an RMA (Return Material Authorization) number from the Customer Service Department, 303–430–1500.
2. If the request is for an out of warranty repair, a purchase order number or other acceptable information must be supplied by the customer.
3. Include a list of problems encountered along with your name, address, telephone, and RMA number.
4. Carefully package the product in an antistatic bag. (Failure to package in antistatic material will VOID all warranties.) Then package in a safe container for shipping.
5. Write RMA number on the outside of the box.
6. For products under warranty, the customer pays for shipping to Octagon. Octagon pays for shipping back to customer.
7. Other conditions and limitations may apply to international shipments.

Note PRODUCTS RETURNED TO OCTAGON FREIGHT COLLECT OR WITHOUT AN RMA NUMBER CANNOT BE ACCEPTED AND WILL BE RETURNED FREIGHT COLLECT.

Returns

There will be a 15% restocking charge on returned product that is unopened and unused, if Octagon accepts such a return. Returns will not be accepted 30 days after purchase. Opened and/or used products, non–standard products, software and printed materials are not returnable without prior written agreement.

Governing law

This agreement is made in, governed by and shall be construed in accordance with the laws of the State of Colorado.

The information in this manual is provided for reference only. Octagon does not assume any liability arising out of the application or use of the information or products described in this manual. This manual may contain or reference information and products protected by copyrights or patents. No license is conveyed under the rights of Octagon or others.