Little Board™/486-II

Technical Manual

P/N: A74130

Revision: B

Ampro Computers, Incorporated
990 Almanor Avenue Sunnyvale, CA 94086
Tel (408) 522-2100 FAX (408) 720-1305

NOTICE

DISCLAIMER

Ampro Computers, Incorporated makes no representations or warranties with respect to the contents of this manual or of the associated Ampro products, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Ampro shall under no circumstances be liable for incidental or consequential damages or related expenses resulting from the use of this product, even if it has been notified of the possibility of such damages. Ampro reserves the right to revise this publication from time to time without obligation to notify any person of such revisions. If errors are found, please contact Ampro at the address listed on the title page of this document.

TRADEMARKS

The Ampro logo is a registered trademark, and Ampro, Little Board, StackPlane, MiniBackplane, MiniModule, and CoreModule are trademarks of Ampro Computers, Inc. All other marks are the property of their respective companies.

TECHNICAL SUPPORT

Technical Support is available from 8:00 AM to 5:00 PM, Pacific time. Please have the product you wish to discuss at hand when calling. The number is 800 966-5200

Revision History

Revision	Reason for Change	Date
1	First Release	4/95
A	Production Release	7/95
B	DRAM TO 70nS	10/95

PREFACE

This manual is for integrators and programmers of systems based on the Ampro Little Board™/486-II single board system. It contains information about hardware requirements, interconnects, and details of how to program the system.

There are three chapters, organized as follows:

- **Chapter 1**—Introduction. General information pertaining to the Little Board/486-II, its features, and specifications.

- **Chapter 2**—Hardware Configuration. A description of how to configure and connect the Little Board/486-II for use with a variety of onboard and external devices. Included are tables listing the pinouts of each of the board's connectors, jumper options, and considerations and specifications regarding peripheral devices.

- **Chapter 3**—Software Configuration. An overview of the system features, configuration options, and utilities that are available under the Disk Operating System (PC-DOS, MS-DOS, or DR DOS), including SETUP guidelines.

- **Chapter 4**—Advanced Topics. Detailed technical information on Little Board/486-II onboard hardware and peripheral interfaces.

TABLE OF CONTENTS

CHAPTER 1—INTRODUCTION

CHAPTER 2—HARDWARE CONFIGURATION

CHAPTER 3—SOFTWARE CONFIGURATION

CHAPTER 4—ADVANCED TOPICS

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The Little Board™/486-II provides a complete, high performance, 80486-based AT-compatible system in the space of a 5-1/4 inch disk drive. It includes all standard motherboard functions, plus memory, serial and parallel ports, and disk controllers.

The Little Board/486-II is ideal for a wide range of performance-oriented embedded microcomputer applications, including those in which IBM PC/AT software, hardware, and bus compatibility are desired. It runs standard PC software, including operating systems such as DOS, UNIX, Windows, OS/2, QNX, VRTX, OS9000, FlexOS, and so forth. It also runs languages such as C, Pascal, Fortran, Basic, and industrial and commercial application software. The Little Board/486-II is also particularly suited to those applications in which compactness, low power consumption, +5V-only operation, wide operating temperature range, and high reliability are crucial.

You can easily expand the system by stacking Ampro MiniModule™ products on the Little Board/486-II. 8-bit PC/104 expansion boards attach directly to the bus interface connector P1. Addition of a single MiniModule creates an assembly that fits entirely within the Little Board's space envelope. Additional modules increase system thickness by 0.6 inches each. Ampro also offers devices to facilitate adding conventional PC and AT expansion cards to the Little Board/486-II.

The Little Board/486-II is offered in two versions:

- 66 MHz, using a 66 MHz 80486DX2, with a motherboard clock rate of 33 MHz, and a CPU running at double speed,
- 100 MHz, using a 100 MHz 80486DX4, with a motherboard clock rate of 33 MHz, and a CPU running at triple speed.

DRAM Memory

Four 30-pin SIMM™ (Single In-line Memory Module) sockets are provided for main system memory. You can configure the Little Board/486-II to have 1M bytes, 4M bytes, 16M bytes, or 32M bytes, depending on the memory modules you add. Ampro offers a custom 8M byte module to support the 32M byte configuration.

Solid State Disk

A unique feature of the Little Board/486-II is a set of three byte-wide memory device sockets. You can add EPROM, Flash EPROM, or battery-backed SRAM devices in these sockets to embed your operating system and application programs. They can be used as regular paged-memory devices, or, more commonly, as a solid-state disk (SSD) drive, in place of conventional disk drives. Using Ampro's optional SSD Support Software, you can easily program the solid-state devices to act like one or more disk drives, to store the operating system, drivers, and application programs. To the operating system, the memory devices look like one or more floppy disk drives. The three sockets can be configured to accommodate a broad variety of memory devices, up to 1M bytes in size, providing up to 3M bytes of storage.

Enhanced BIOS

The Little Board/486-II is equipped with an enhanced BIOS, providing additional services and functions that meet the unique requirements of embedded microcomputer systems. These enhancements include full SCSI support for the onboard SCSI controller, an enhanced SETUP function allowing full control of system features such as the serial ports, parallel port, ROM-BIOS and video-BIOS shadowing options, and configuration of the byte-wide sockets. It supports a watchdog timer function, implemented with the real time clock's alarm function, and supports the use of a serial console. It also provides for automatically downloading program code via a serial connection at boot time. A more complete list of these BIOS enhancements appears in the specifications section of this chapter.

Enhanced Reliability

The Little Board/486-II is equipped with two features to monitor the system environment and provide for clean reset or software recovery of the board should an unstable state occur. The first is the watchdog timer. It can be used to monitor the boot sequence and can be incorporated into your application program to indicate when a process has failed to take place. It can be jumpered to cause an NMI interrupt or reset should a fault condition be detected. The second is a supply voltage sensor that triggers a hard reset of the board when the supply voltage falls below threshold. This prevents erratic operation if the power supply fluctuates.

1.2 LITTLE BOARD/486-II FEATURES

- A complete AT-compatible system on one board: all the functions of a motherboard and 3 or 4 expansion cards in the space of a half-height 5-1/4 inch disk drive
- Runs standard PC software, including:
  - — Disk Operating Systems
  - — Drivers
  - — Languages (C, Pascal, Fortran, Basic, …)
  - — Industrial and commercial application software
  - — Your own applications for your embedded system
- Choice of 66 MHz 80486DX2 or 100 MHz 80486DX4 CPU
- Standard AT DMA, timers, and interrupt controllers
- Complete onboard system memory
  - — 1M, 4M, 16M, or 32M bytes onboard DRAM
  - — Up to 3M bytes of EPROM
  - — Up to 1.5M bytes of Flash EPROM
  - — Up to 512K bytes of SRAM, backed up with an onboard battery
- Complete set of AT-compatible peripheral ports and controllers:
  - — Floppy controller, supporting 250K and 500K byte/sec, 3-1/2 inch and 5-1/4 inch floppy drives
  - — Two RS232C serial ports, parallel printer port, keyboard port, and speaker port
  - — Battery-backed real time clock
  - — IDE hard disk drive interface

— Fast SCSI-2 controller and BIOS support for SCSI peripherals including hard disks

- PC/104-compatible expansion bus. Provides for onboard expansion with PC/104 modules from Ampro or other sources
- Industrialized ROM-BIOS, with support for a variety of SCSI devices (hard disk, tape, CD ROM, and so forth), bootable Solid State Disk, watchdog timer, voltage-level sensor, serial console, serial downloader.

Ideal for Embedded Applications

- Low power CMOS design:
  - — Typically uses only 12.7 watts of power with 4M bytes of DRAM installed
  - — Single +5 VDC supply operation
- Wide operating temperature range (0º – 70º C)
- Reliable:
  - — Low component count
  - — No backplanes or edgecard connectors required
- Onboard bootable Solid State Disk drive (EPROM/NOVRAM/Flash EPROM) option allows diskless DOS operation
- Two Industry Standard Expansion Buses:
  - — Expansion bus for one onboard PC/104 module stack or cabling to external PC and PC/AT bus-connected devices cards
  - — 16-bit SCSI (SCSI-2 compatible) bus for addition of Disk/Tape/Optical drives, scanners, and other peripheral devices

1.3 LITTLE BOARD/486-II SPECIFICATIONS

CPU

- Choice of:
  - — 66 MHz 80486DX2 (internally clock-doubled)
  - — 100 MHz 80486DX4 (internally clock-tripled)
- Standard 8 MHz AT bus

Onboard Memory

- 1M, 4M, 16M, or 32M bytes DRAM with parity
- Award ROM-BIOS with Ampro extensions

- Three 32-pin byte-wide sockets. Two PLCCs, one 32-pin DIP

  PLCC sockets usable with:

  — 128K to 1M EPROMs

  — 32K to 512K Flash EPROMs

  DIP usable with:

  — 32K to 1M EPROMs

  — 32K to 512K Flash EPROMs

  — 32K to 512K SRAMs. Onboard battery converts SRAM to NOVRAM

  — Configurable as 32K, 64K, or 128K byte window, at address D0000-EFFFFh

  — Software-controlled +12V programming power for Flash EPROMs

  — Software-controlled write protect (for DIP socket only)
- 2K bit serial EEPROM for system parameter storage, with 512 bits for OEM use

AT-Compatible Controllers

- Standard DMA, interrupt and counter-timer support:

  — 7 DMA channels

  — 15 interrupt channels

  — 3 programmable counter-timers
- Two RS232C serial ports
- Parallel printer port with 8 bi-directional data lines for digital I/O:

  – Data and Control lines will sink 12 mA (at .4 v).

  – Data lines will source 2 mA @ 2.4 v

  – Control lines will source 1-5 mA at .8 v
- AT keyboard port
- Speaker port—100 mW
- Standard battery-backed real time clock and CMOS RAM
- AT-compatible 5¼ inch and 3½ inch floppy controller:

  — 2 drive selects, 1-2 sided, 250/500K byte/sec data rates

  — BIOS supports all standard formats (360K/720K/1.2M/1.44M bytes)

  — Reliable all-digital phase-locked loop and write precompensation

  — Disk change support
- IDE hard disk drive interface

SCSI Interface

- 16-bit interface to CPU, 8-bit SCSI bus
- Full ANSI X3.131 (SCSI) compatible
- Hardware and software compatible with SCSI-2 specification

- Uses Adaptec AIC-6360 SCSI asynchronous controller
- Up to 5M bytes/sec transfer rate using the Ampro SCSI BIOS

Physical

- 8.0 x 5.75 x 1.1 inches (46 x 203 x 30 mm)
- Provision for system expansion with one or more onboard Ampro MiniModule products
- Power required (typical, with 4M byte DRAM):
  - — 66 MHz 80486DX2 CPU version: +5V ±5%, at 2.6 Amps
  - — 100 MHz 80486DX4 CPU version: +5V ±5%, at 2.6 Amps
- 8-layer PCB using latest surface mount technology
- Operating environment:
  - — 0° to 70° C (with adequate airflow)
  - — 5 to 95% relative humidity (non-condensing)
- Storage temperature: -55° to +85° C
- Weight: 11.5 oz. (326 gm)

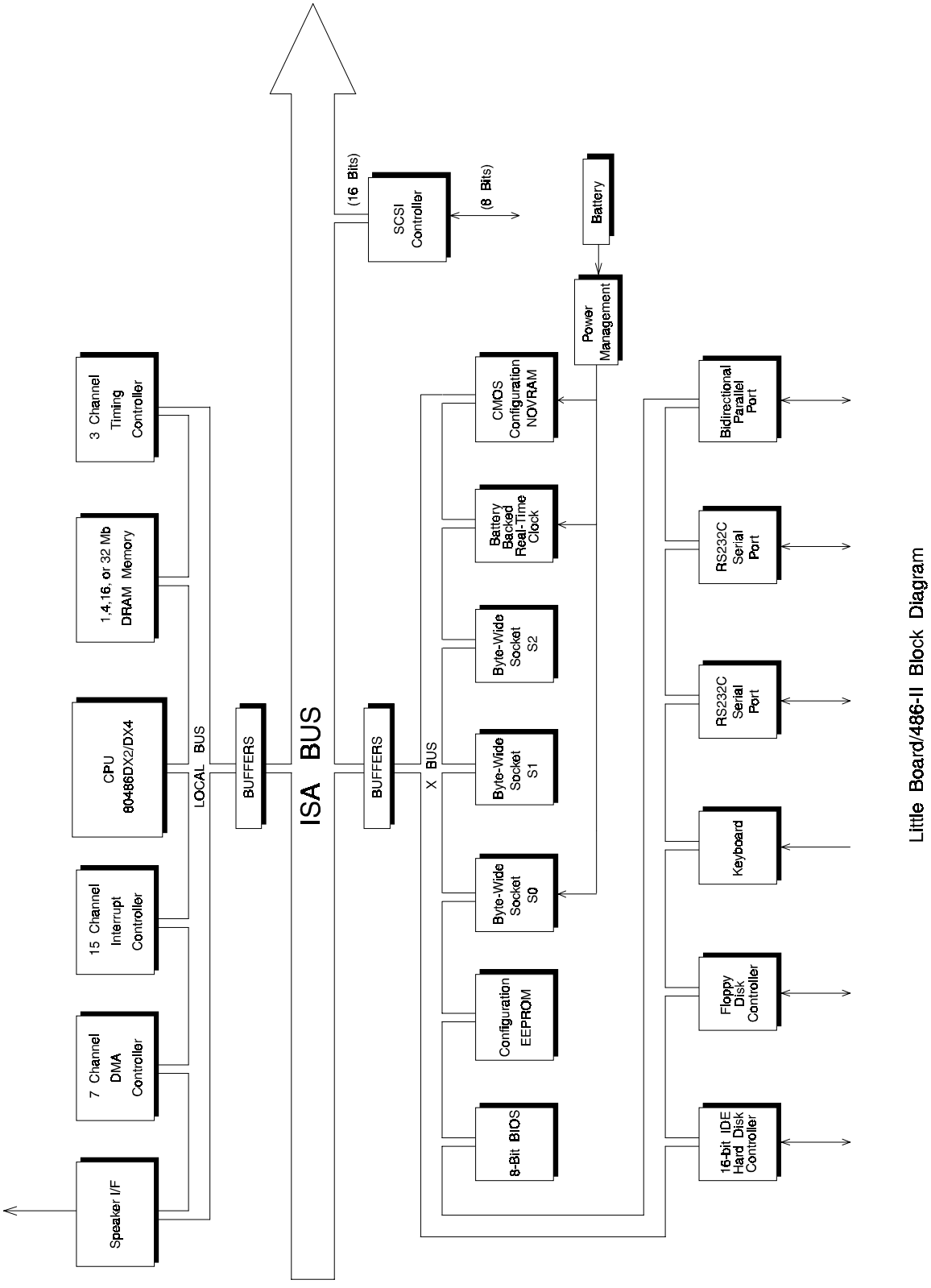


Figure 1-1 Little Board/486-II Block Diagram

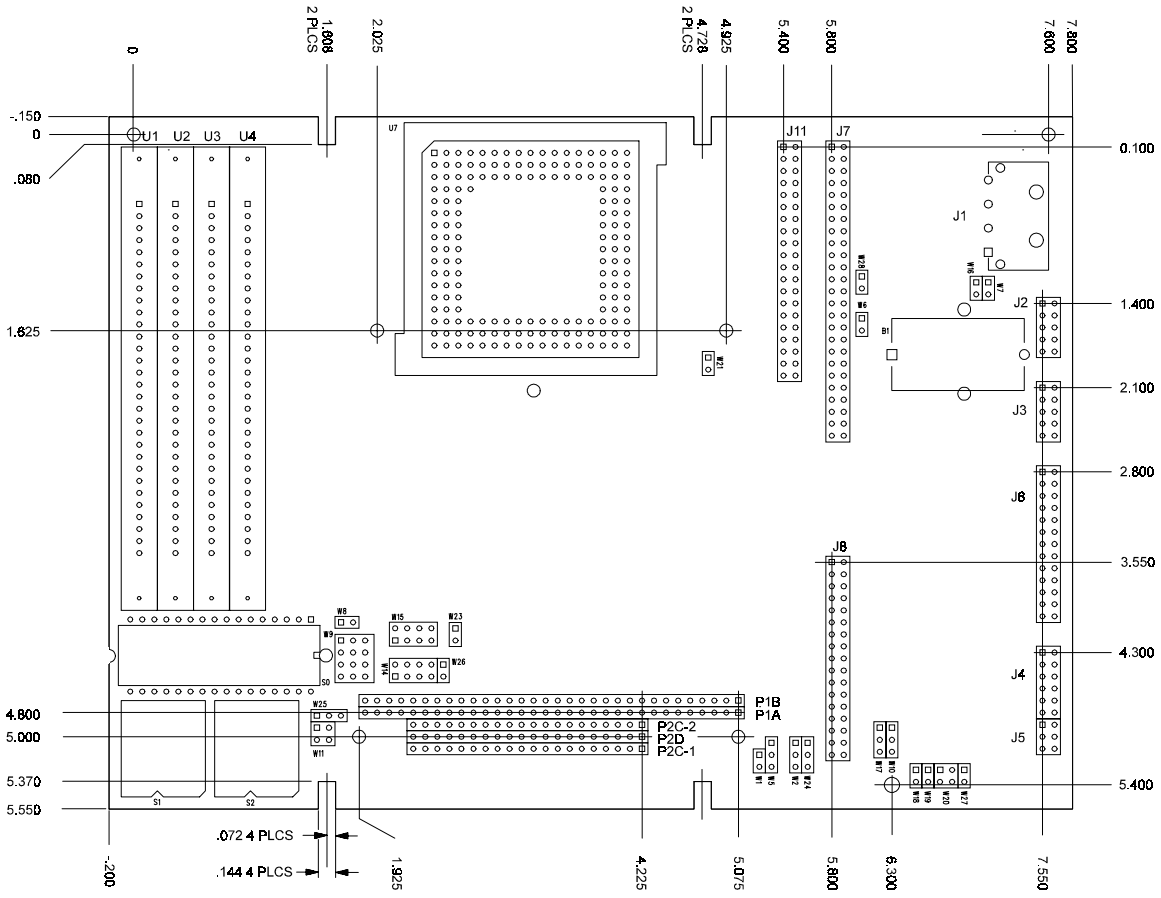


Figure 1-2 Mechanical Dimensions

CHAPTER 2

HARDWARE CONFIGURATION

2.1 INTRODUCTION

This chapter covers configuring the Little Board/486-II and using onboard and peripheral devices. The Little Board/486-II standard features include a complete set of ports including serial, parallel, keyboard, speaker and floppy disk interfaces, both IDE and SCSI hard disk interfaces, standard interrupt and DMA controllers, the usual timers, and a battery-backed real-time clock. Special features include three byte-wide sockets and the PC/104 expansion bus, used for system expansion with Ampro MiniModule products and other PC/104-compliant add-on modules.

This chapter includes data on the board's connector signals and pinouts, external device requirements, interconnection cable wiring, and board configuration.

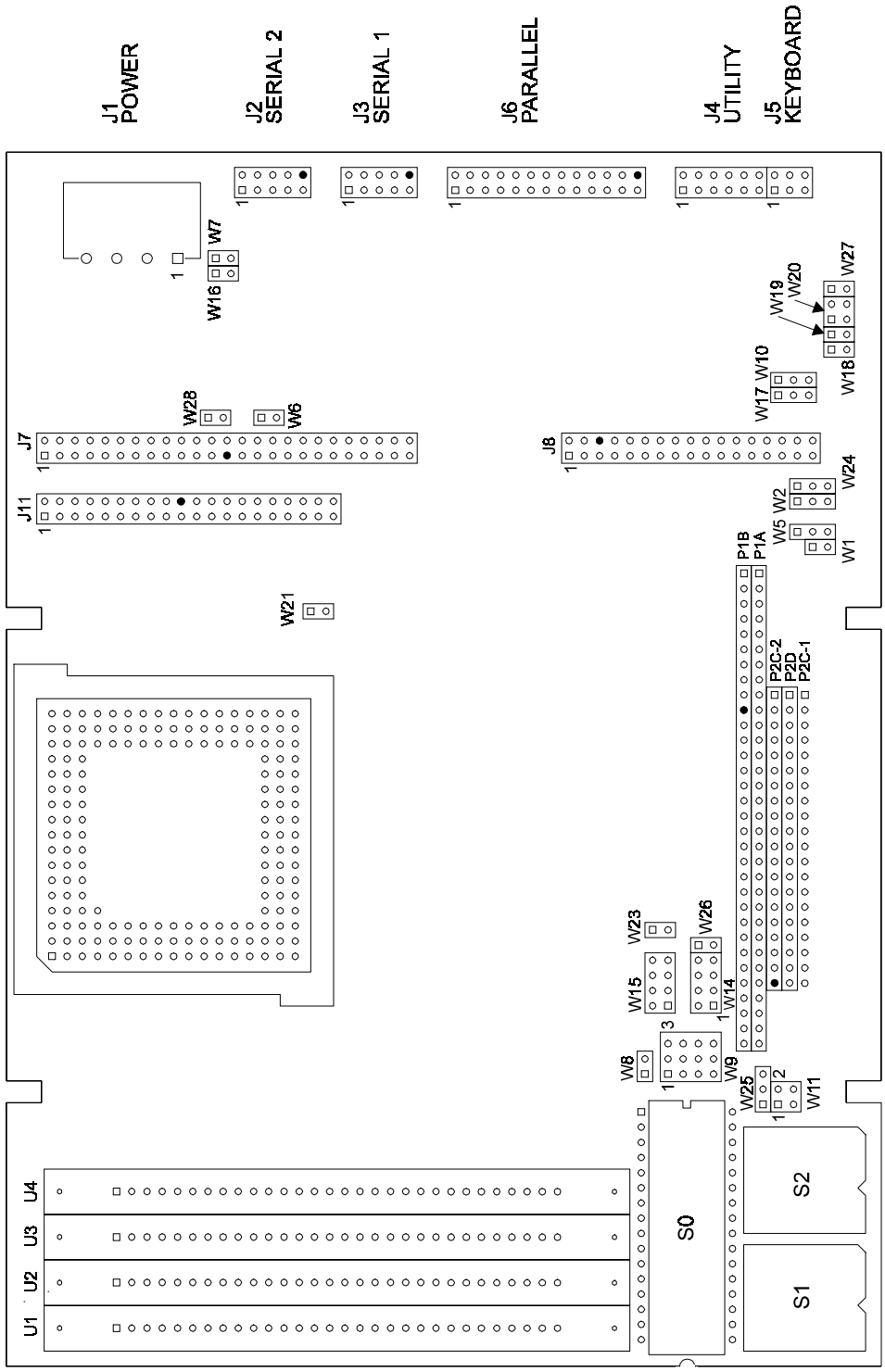
2.1.1 Interface Connectors

Figure 2-1 shows the location of the connectors (J1-J11) and configuration jumpers (W1-W28). Table 2-1 summarizes use of the connectors. Table 2-2 summarizes use of the configuration jumpers. Each interface is described in its own section, showing connector pinouts, signal definitions, required mating connectors, and configuration jumper options.

Many of the connectors have a key pin removed. This allows you to block the corresponding connector socket to help prevent improper assembly. Table 2-1 lists key pins, and Figure 2-1 shows their locations.

Connector	Function	Size	Key Pin
P1	AT Bus	64-pin	B10
P2	AT Bus	40-pin	C19
J1	Power	4-pin	None
J2	Secondary serial port	10-pin	10
J3	Primary serial port	10-pin	10
J4 (*)	Utility	12-pin	None
J5 (*)	Keyboard	6-pin	None
J6	Parallel port	26-pin	26
J7	SCSI port	50-pin	25
J8	Floppy drive	34-pin	6
J9, J10	Unused		
J11	IDE drive interface	40-pin	20
(*) J4 and J5 are a single 18-pin array			

Table 2-1 Connector Usage Summary



(Shaded connector pins indicate key pins.)

Figure 2-3 Connector and Jumper Locations

The I/O connectors are dual-row headers for use with flat ribbon (IDC) or discretely wired connectors. You can also use specialized connectors or a dedicated PC board assembly to connect directly to these connectors. You might do this to eliminate cables, meet packaging requirements, add EMI filtering, or customize your installation in other ways. The PC/AT expansion bus appears on two connectors (P1 and P2). You can expand the system with onboard MiniModules that stack directly on the connectors, or use conventional or custom expansion hardware, including solutions available from Ampro.

2.1.2 Jumper Configuration Options

The Little Board/486-II requires no special jumpering for standard AT operation. You can connect the peripherals and operate it immediately. The only jumpers of concern are those that configure the byte-wide sockets for the devices you install. Jumpers are also provided to disable various interfaces, so that you can use their I/O addresses, DMA channels, and interrupts for other purposes.

The jumper arrays are designated W1, W2, and so forth. A square solder pad identifies pin 1 of each jumper array. Table 2-2 is a summary of jumper use. A slash (1/2) means to short the indicated pins.

Jumper Group	Function	Default
W1	Enables SCSI chip select	1/2
W2	Byte-wide S0 chip select	2/3
W4	SCSI DRQ6/DRQ0 select	Open
W5	SCSI IRQ15/IRQ11 selection	Open
W6	SCSI termination power	Open
W7	Power connection, J1-1 to +12V Bus	Open
W8	Address 19 to byte-wide socket S0-1	Open
W9	Byte-wide socket S0 configuration (128K SRAM)	1/2, 4/7, 5/8, 10/11
W11	BIOS/Byte-wide select	1/2, 3/4
W14	Byte-wide S1 configuration	7/10, 8/9, 11/12, 13/14
W15	Byte-wide S2 configuration	7/10, 8/9, 11/12, 13/14
W16	Flash EPROM Programming +12 volts	Open
W17	Watchdog Timer reset/NMI select	Open
W19	Floppy precompensation (PCVAL) selection	Open
W20	Floppy IRQ6, DRQ2 enable	1/2, 3/4
W23	Power Fail NMI enable	Open
W24	Printer interrupt select	1/2
W25	Socket S0 battery backup enable	1/2
W26	+12 volt power for BIOS programming	Open
W27	Floppy Drive DACK2 enable	1/2
W28	SCSI Active Terminator enable	1/2
W3, W10, W12, W13, W18, W21, W22	Not used, or factory setting	

Table 2-2 Configuration Jumper Summary

2.1.3 SETUP Options

The SETUP function, included in the ROM BIOS and on the Common Utilities diskette, stores the configuration parameters in a 2 Kbit nonvolatile EEPROM and in the battery backed CMOS RAM in the real time clock. These two memories comprise the Configuration Memory. During the boot process, the ROM BIOS initializes system parameters based on the contents of these memories. The contents of the CMOS RAM (except time and date) are also stored in the EEPROM. Even without battery power, the SETUP parameters are saved.

You can enter SETUP two ways. One is a "hot-key" entry. Press CTRL-ALT-ESC at the same time just prior to boot. The other is through SETUP.COM, a program provided on the utilities diskette. Use of SETUP is covered in detail in Chapter 3. Using the SETUP function, you can also load system setup parameters from a disk file. This is useful when installing a standard configuration, for instance, when configuring many systems during production.

2.2 DC POWER

The pinout of the power connector J1 is identical with the power connectors on 5-1/4 inch floppy disk drives. Refer to Table 2-3 for power connections, Table 2-4 for mating connector information, and Figure 2-2 for typical connector wiring. Table 2-4 gives the part numbers for mating connectors for J1.

If you intend to program Flash devices installed in the byte-wide sockets, you can connect the +12 volt programming voltage through the power connector. (Power cables from standard PC power supplies often supply +12 volts to their power connectors.) A jumper is provided to connect the +12 volt power connector pin to the +12 volt bus on the board. Shorting W7-1/2 connects +12 volts from the power connector (J1) to the board and the AT expansion bus.

Caution

Be sure the power plug is wired correctly before applying power to the board! See Figure 2-2.

Pin	Signal Name	Function
1	+12VDC	+12VDC ±5% input
2, 3	Ground	Ground return
4	+5VDC	+5VDC ±5% input

Table 2-3 Power Connector (J1)

Connector Type	Mating Connector
DISCRETE WIRE	AMP HOUSING 1-480424-0 AMP PIN 60619-1

Table 2-4 J1 Mating Connector

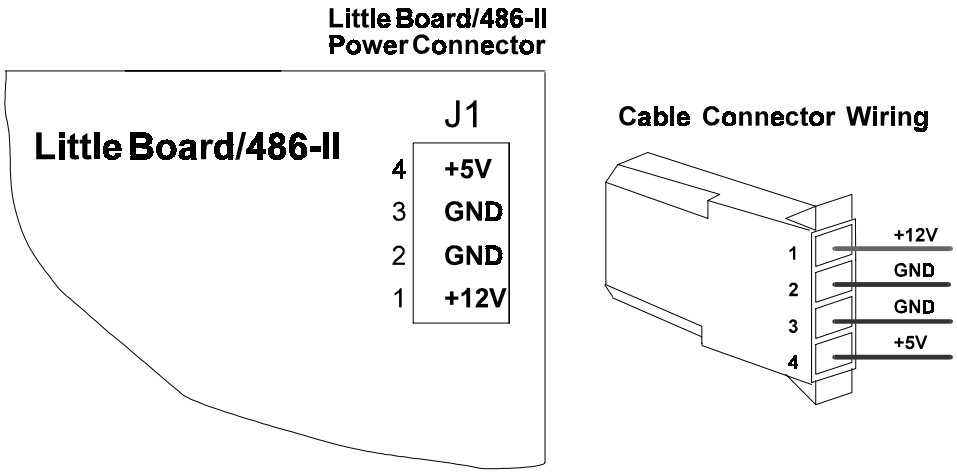


Figure 2-4 Power Connector Wiring

2.2.1 Power Requirements

The Little Board/486-II requires only +5 VDC (±5%) for operation. The ±12 volts for the RS232 ports is generated onboard from the +5 VDC supply. However, if you use +12 volt Flash EPROMs, you must supply them with 12 volts during programming. Consult the documentation for your Flash EPROMs for their programming current requirements.

The exact power requirement of the Little Board/486-II system depends on several factors: the quantity of DRAM, byte-wide memory devices; SCSI bus termination; CPU speed; the peripheral connections, and which, if any, MiniModules or other expansion boards are attached to the PC/104 bus. For example, AT keyboards draw their power from the board, and there can be some loading from the serial and parallel ports.

A fully populated Little Board/486-II with 4 MBytes of DRAM, but no expansion modules installed, and no connections to peripherals draws approximately 2.6 amps for both the 66 MHz 486DX2 version; and the 100 MHz 80486DX4 version.

If you use a switching power supply, be sure it regulates properly with the load for your system configuration. Some switching power supplies do not regulate properly unless they are loaded to some minimum value. If this is the case with your supply, consult the manufacturer about additional loading, or use a linear power supply or batteries.

2.2.2 Powerfail Options

The Little Board/486-II includes a circuit to monitor for a power failure. Configuring the board for the power failure options involves jumper arrays W23 and W25. To enable the powerfail circuitry, short W23 with a jumper. (If you don’t want the power fail NMI, leave W23 open.)

If the +5 volt power supply falls below 4.7 volts, the powerfail logic produces a non-maskable interrupt (NMI). The BIOS detects the NMI and displays the message "Power Fail NMI" on the screen. You have two options at this point (made by keyboard selections). One is to mask the NMI and continue. The other is to reboot the system. This, of course, requires operator intervention. If you want an automatic response to the NMI, you can provide an NMI handler in your application, and patch the NMI interrupt vector address to point to your routine. This is covered in Chapter 4.

If the supply voltage falls below (approximately) 4.5 volts, the powerfail logic initiates a hardware reset (like pressing the RESET button). A "clean" reset during a low voltage period prevents "flakey"

operation or uncontrolled crashes. Reset is asserted for the duration of the low-voltage period plus 100 mS after the voltage returns to above 4.5 volts

Jumper array W25 determines whether backup battery power is routed to byte-wide socket S0 during a power failure. If you jumper W25-2/3, the power monitor circuit switches the power supply for byte-wide socket S0 from the +5 volt power supply (Vcc) to the backup battery. This effectively makes an SRAM in S0 into non-volatile RAM (NOVRAM). However, if you use an EPROM or Flash device in S0, jumper W25 to short pins 1/2. With this setting the power supply to S0 is always Vcc, and battery power is not wasted.

Byte-wide S0 is write protected while power is below 4.7 volts. (Its chip select is held to a logic 1.) This is to prevent writing bad data to a SRAM in S0 when the voltage is low.

2.2.3 Backup Battery

With only the real-time clock drawing current, the backup battery on the Little Board/486-II should last 10 years. If it supplies only the clock, replace the battery every 10 years as a routine maintenance procedure.

Caution

Lithium batteries can explode if mistreated. Do not attempt to recharge a Lithium battery. See manufacturers instructions for proper disposal of used batteries. Be careful not to overheat the battery when changing it.

If the battery supports an SRAM, calculate the battery life using the formula below, and replace the battery as necessary. To calculate battery life, add the SRAM current and the clock current (5 uA) and divide 750 milliamp-hours by the sum. Then, multiply that result by the duty cycle of the battery. That is, estimate the percentage of time the battery supplies power (while the system is off).

Here is the formula for calculating battery life (in hours):

Battery life = (750 milliamp-hours ÷ (5 uA + SRAM backup current)) × Duty Cycle

2.3 COOLING REQUIREMENTS

The 80486 CPU, DRAM, and core logic chip draw most of the power and generate most of the heat. A heat sink is provided for the CPU, but in most cases you must provide additional air flow. Table 2-5 shows how much air flow, in meters per second, is required for various ambient operating temperatures. As indicated, the required airflow differs according to which CPU speed version (66 or 100 MHz) you are using.

Air Flow (meters/sec)	0	1	2	3	4	5
80486DX2 (66 MHz) version Ambient temperature, °C	19	41	57	63	66	69
80486DX4 (100 MHz) version Ambient temperature, °C	35.5	57	70			

Table 2-5 Airflow and Ambient Temperature

2.4 ONBOARD DEVICE CONFIGURATION

This section includes the configuration and installation of onboard devices and options. These include the DRAM, Cache RAM, the byte-wide memory devices, the serial ports, the parallel port and the floppy, SCSI, and IDE controllers. It includes system expansion with MiniModules and conventional expansion boards.

2.4.1 DRAM

The board has positions for four single in-line memory modules (SIMMs). Because the DRAM is organized as a 32-bit data bus for the CPU, you must install memory in all four positions, as each SIMM is 8 bits wide (plus an additional bit for DRAM parity). Use either 256 KBytes x 9, 1 MBytes x 9, 4 MBytes x 9, or 8 MBytes x 9 SIMM modules, depending on the amount of memory desired. Use Fast Page Mode DRAMs with access times of 70 nS or less. (The 8 MByte SIMM modules are available from Ampro.)

Onboard memory is allocated as follows:

- The first 640K bytes of DRAM are assigned to the DOS region 00000h to 9FFFFh. No user DRAM is addressed to the top 384K bytes of the first 1M byte. DRAM is often mapped into a portion of this 384K region to *shadow* (see below) a video BIOS when a VGA video board is installed, and the top 128K, to shadow the ROM BIOS. The minimum DRAM installation is 1M byte.

- When more than 1M byte of DRAM is installed, the remaining memory is mapped to extended memory starting at the 1 megabyte boundary, 100000h.

When the system boots, the BIOS measures the amount of memory installed and configures the internal memory controller for that amount. (No jumpering or manual configuration is required.) The amount it measured can be displayed by running SETUP. Saving SETUP automatically stores this figure in the Configuration Memory. If you change the amount of memory installed, you must run SETUP again to save the new value in the Configuration Memory.

Memory above the 1 megabyte boundary is called "extended" memory. Some programs require that memory be available as EMS memory. EMS memory is available as pages rather than as a contiguous block. The exact manner for accessing EMS memory is defined in the LIM 4.0 specification. You can convert the board's extended memory into expanded memory using MS-DOS or DR DOS EMS emulation utilities conforming to the LIM 4.0 specification. Refer to the DOS technical documentation for instructions for using their EMS emulation utility.

Shadowing

One way to improve system performance is to shadow the ROM BIOS and video BIOS. When the system operates directly from ROM, it accesses an 8-bit memory device. When the ROM contents are shadowed, the contents are copied into system DRAM where they are accessed as 32-bit wide data. Shadowing a BIOS ROM substantially enhances system performance, especially when an application or operating system repeatedly accesses the ROM. Shadowing the ROM BIOS is built into the Ampro Extended BIOS. There is no user setting. Shadowing the video BIOS is an option, available in SETUP.

2.4.2 Math Coprocessor

The 80486 CPU (DX2 and DX4) contains a built-in floating point math coprocessor. There are no configuration jumpers or options for the math coprocessor.

2.4.3 Byte-Wide Sockets

The Little Board/486-II has three onboard byte-wide memory sockets. One is a 32-pin DIP, and the other two are 32-pin PLCC (Plastic Leaded Chip Carrier) sockets. All of these sockets accept a variety of EPROM and Flash EPROM devices. The DIP socket (S0) also accepts SRAMs and nonvolatile RAM (NOVRAM) devices. You can make an SRAM in S0 "non-volatile" with the connection of battery backup power, a jumper option.

You can use these devices for simple program storage, BIOS extensions, or as Solid State Disk (SSD) drives. Table 2-6 shows some byte-wide devices and in which sockets they can be installed.

You can use either 28- or 32-pin devices in socket S0. If you use a 28-pin device, install it in accordance with Figure 2-3, with pin 1 of the device installed in the socket's pin 3.

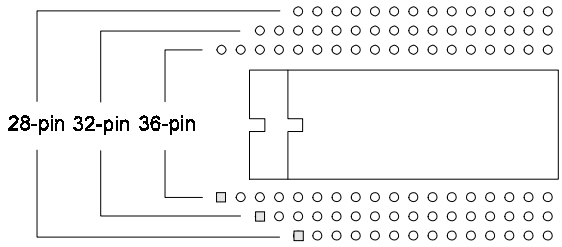


Figure 2-5 Using a 28-Pin Device in a 32-Pin Socket

Device Type	Size	Typical Part Number	Usable In
EPROMs			
EPROM32	32K byte	27C256	S0
EPROM64	64K byte	27C512	S0
EPROM128	128K byte	27C010	S0, S1, S2
EPROM256	256K byte	27C020	S0, S1, S2
EPROM512	512K byte	27C040	S0, S1, S2
EPROM1024	1024K byte	27C080	S0, S1, S2
STATIC RAMs			
SRAM32	32K bytes	62256, 43256, 51257	S0
SRAM128	128K bytes	628128, 66204	S0
SRAM512	512K bytes	434000	S0
FLASH EPROMs			
EPROM128	128K bytes	28F010	S0, S1, S2
EPROM256	256K bytes	28F020	S0, S1, S2
EPROM512	512K bytes	29F040	S0, S1, S2
Maximum access time for byte-wide devices is 250 nS.			

Table 2-6 Typical Byte-wide Devices

Use the CPU SETUP program to specify the size and starting address of each socket, and which device the BIOS enables upon system initialization. Table 2-7 lists the possible settings for sizes and address ranges of the byte-wide sockets.

Note

When a byte-wide device is enabled, the memory address space it uses is unavailable for other devices. You must disable the byte-wide sockets in SETUP before you can use the memory space for other purposes.

Window	Address
DISABLE	N/A
32K	D0000-D7FFFh
32K	D8000-DFFFFh
32K	E0000-E7FFFh
32K	E8000-EFFFFh
64K	D0000-DFFFFh
64K	E0000-EFFFFh
128K	D0000-EFFFFh

Table 2-7 Window Size and Address Selection

Direct Program Access

Application software can access the memory devices in S0, S1, and S2 if the program knows about them. To access a byte-wide socket, you must enable it with a BIOS call. See Chapter 4 for more details on this BIOS call.

The system can run its entire application from memories in the byte-wide sockets, instead of loading it into DRAM from a disk drive. This technique, known as a ROM BIOS extension, is discussed in Ampro Application Note AAN-8702. Note that executing programs directly from the byte-wide sockets can adversely affect system performance. Since all three sockets share the same memory address space, the one you want to execute from must be enabled. Also, byte-wide devices are substantially slower than DRAM, as they are 8-bit devices instead of 32-bit. In addition, they are accessed from the PC bus, which is much slower than the high-speed processor memory bus. You can improve performance substantially by copying the byte-wide devices' contents into RAM and executing the RAM copy.

Solid State Disk (SSD) Drives

Using the Ampro Solid State Disk (SSD) Support Software, you can configure EPROM, Flash EPROM, or SRAM solid-state devices, installed in the byte-wide sockets, to act as one or more disk drives. No custom programming is required. Regular DOS-compliant programs can be used without modification. Ampro's SSD support software creates data image files, based on your application programs and operating system, that can be programmed into the devices you install in the byte-wide sockets. The Ampro ROM-BIOS treats these devices like one or more disk drives. The sockets can serve as a single drive or multiple drives. You can use SSD drives in addition to, or instead of, normal floppy and hard disk drives. Using the three byte-wide sockets, up to three megabytes of storage is available. You can increase system SSD capacity by adding one or more of Ampro's SSD expansion modules.

Jumpering the Byte-Wide Sockets

You must jumper the byte-wide sockets for the devices you install in them. Jumper arrays W8 and W9 configure S0 for a particular device type. W14 configures S1, and W15 configures S2. Figures 2-4 to 2-8 show how to install jumpers for supported memory devices. See Chapter 4 for a description of the byte-wide socket signals that correspond to each jumper pin. Besides jumpering each socket, you must use SETUP to select which one the BIOS will enable on power up and its size and address.

If you use a Flash EPROM in byte-wide socket S0, you must jumper W16 to supply the +12 volts to the software controlled switch before attempting to program the device. The programming voltage is also under software control. A utility program for programming supported Flash devices is included on the utility disk. See Section 4.9.5 for details on programming Flash EPROMs.

If you install an SRAM in S0, you can provide backup power from the battery when power is off by shorting W25-2/3 and W2-1/2.

Note

Some byte-wide devices draw battery backup current through their chip select lines when power is off. When using memory devices that do not require battery backup power, set W2 to 2/3. This prevents the backup battery from being drained prematurely.

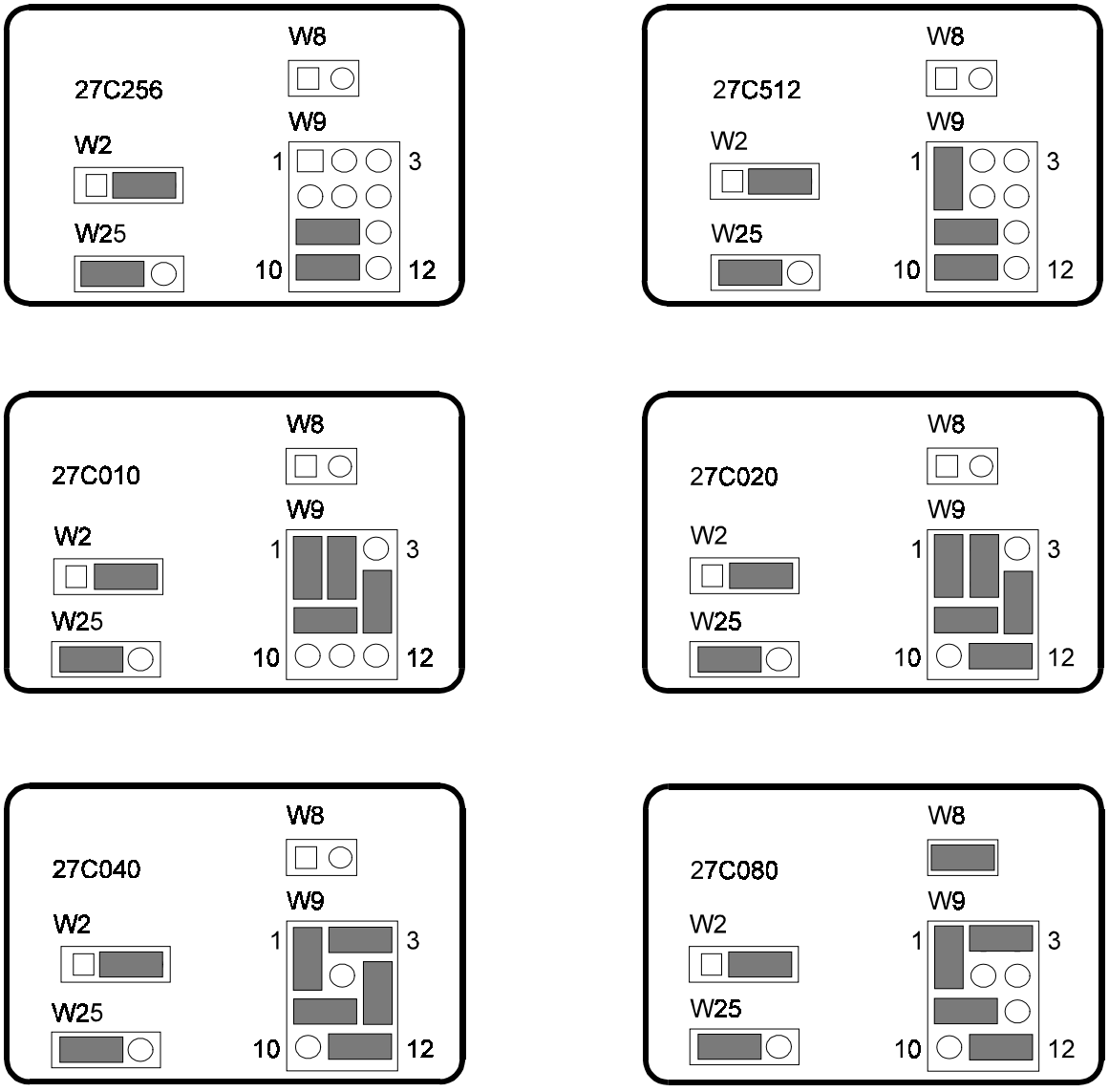


Figure 2-6 EPROM Jumpering for S0

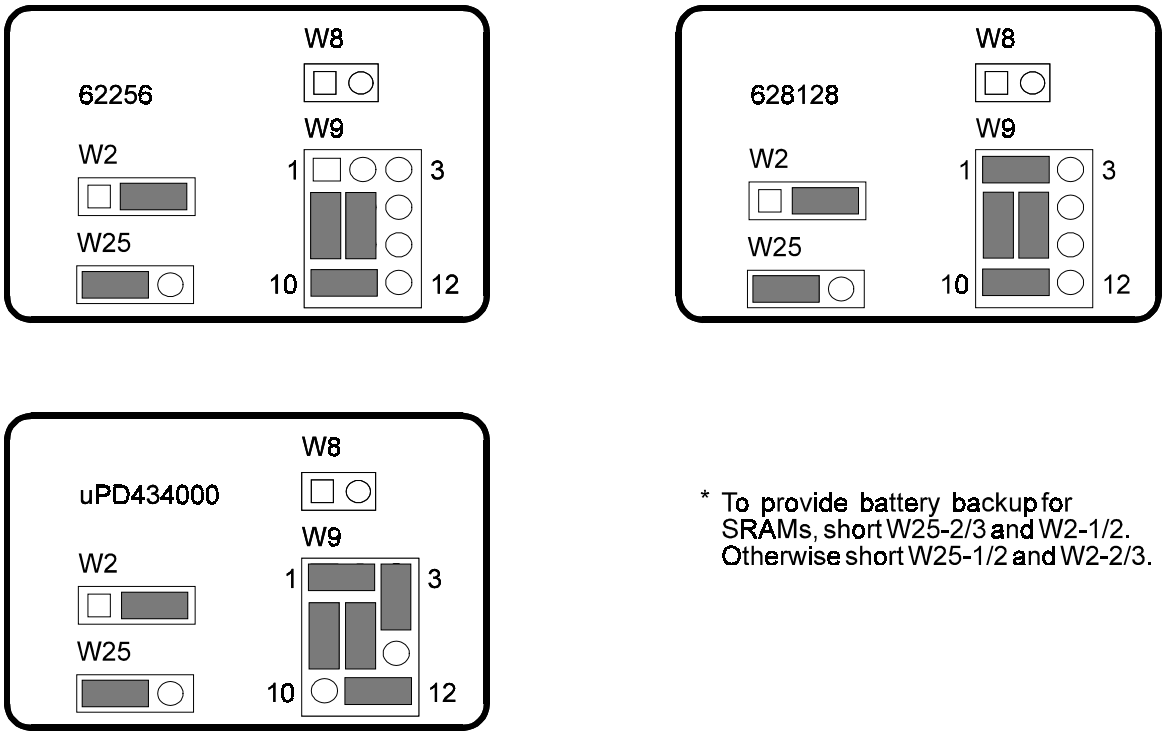


Figure 2-7 SRAM Jumpering for S0

Figure 2-8 Flash EPROM Jumpering for S0

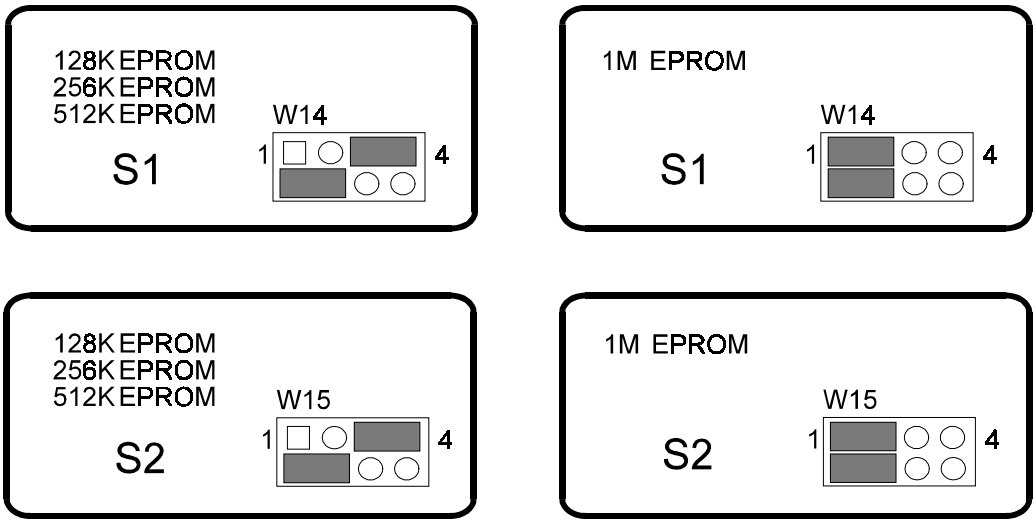
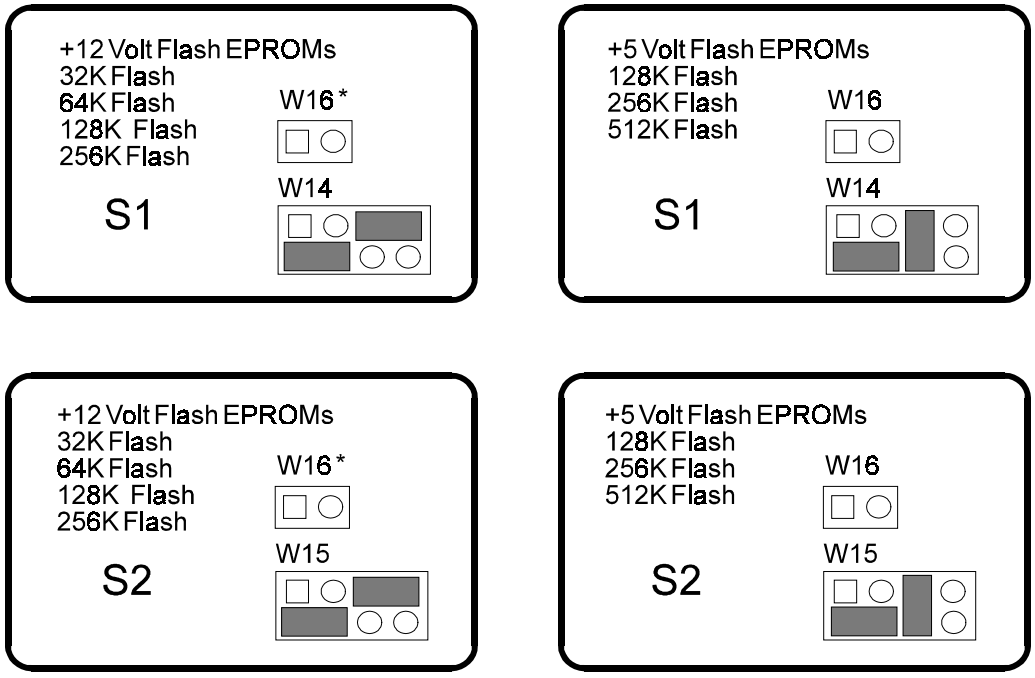


Figure 2-9 EPROM Jumpering for S1 and S2



* To provide +12V Flash programming power, short W7 and W16.

Figure 2-10 Flash EPROM Jumpering for S1 and S2

2.4.4 Battery-Backed Clock

An AT compatible battery-backed real time clock (with CMOS RAM) is standard on the Little Board/486-II. The clock is powered by a 3.6 volt 1/2 AA Lithium battery soldered to the board. This battery will support the clock for about 10 years. (This assumes that the battery is not also used to supply backup power to SRAM devices in the byte-wide sockets.)

The factory initializes the real time clock and various parameters in the Configuration Memory for a standard configuration. The factory sets the date and time, but it may not be set for your time zone. Use the Ampro SETUP utility to change these values as needed.

2.4.5 Watchdog Timer Option

A unique feature of the onboard clock circuitry is a watchdog timer. You can program this timer to generate an interrupt or reset signal if the programmed time interval expires before the timer is reinitialized. Use SETUP to select the time interval. The options are: Disable, 30 seconds, 60 seconds, and 90 seconds.

The watchdog timer uses the standard alarm feature of the real time clock. In a standard AT, the alarm output is connected to IRQ8. On the Little Board/486-II you can also jumper the alarm output to I/O Channel Check (-IOCHCK) or Reset with W17. I/O Channel Check is the bus signal that triggers a non-maskable interrupt (NMI). Reset is a hard reset signal, the same as pressing the Reset button. For the watchdog timer to generate I/O Channel Check, short W17-1/2. For Reset, short W17-2/3. To disable the watchdog timer, leave W17 open, and select Watchdog Timer Disable in SETUP. See Table 2-8.

If you enable the watchdog timer in SETUP, but do not install a jumper on W17, IRQ8 will turn off the interrupt and the system will continue, unaffected. If you select I/O Channel Check, the watchdog timer will generate a message on the screen. If you select Reset, no interrupt handler is required.

Selection	W17
Disabled	Open
-IOCHCK	1/2
RESET	2/3

Table 2-8 Watchdog Timer Configuration

2.5 VIDEO DISPLAY MODE

Whatever type of video controller you use—an onboard MiniModule or a board installed on the AT expansion bus—you have to use SETUP to establish the powerup (or reset) video state as "color" or "monochrome". Table 2-9 shows which setting to use for various video modes.

Video Display	Video Mode	Mono/Color Jumper
Mono (MDA or Hercules)	MONO	MONO
Mono (EGA, or VGA)	EGA/VGA	COLOR
Color (CGA, EGA, or VGA)	COLOR 40 COLOR 80 EGA/VGA	COLOR

Table 2-9 Video Mode Options

Be sure to set any appropriate switches or jumpers on the video display controller for the video mode you use. Refer to your video controller's technical manual.

2.6 PERIPHERAL CONNECTIONS

This section covers the interface requirements of external devices. It provides information on interface and device characteristics, connector pinouts, mating connector part numbers, signal definitions, jumper configuration, and Configuration Memory setup.

2.6.1 Utility Connector (J4)

Five functions appear on the 12 pin connector at J4. These are:

- Speaker
- Push-button reset
- Power indicator LED
- Auxiliary power connections
- Power Good status input signal.

Table 2-10 shows the pinout and signal definitions of the Utility Connector. Connectors J4 and J5 appear on a single header array to facilitate use of a single connector for both. Table 2-11 shows manufacturer's part numbers for mating connectors.

Pin	Signal Name	Function
1	Speaker +	Audio signal
2	Speaker -	Ground
3	Ground	To one side of Reset button
4	Reset	To other side of Reset button
5	LED Cathode	Ground return
6	LED Anode	Current source (+5V through 330 ohms)
7	Ground	Ground return
8	+12V power	Connected to P1 pin B9
9	-5V power	Connected to P1 pin B5
10	-12V power	Connected to P1 pin B7
11	Ground	Ground return
12	POWERGOOD	Power supply status

Table 2-10 Utility Connector (J4)

Connector Type	Mating Connector
RIBBON	3M 3473-7010
DISCRETE WIRE	MOLEX HOUSING 22-55-2101 PIN 16-02-0103

Table 2-11 J4 and J5 Mating Connector

Speaker
The board supplies about 100 milliwatts for a speaker on pins 1 and 2 of the Utility Connector. A transistor amplifier buffers the speaker signal. Use a small general purpose 2 or 3 inch permanent magnet speaker with an 8 ohm voice coil. Refer to an AT technical reference manual for custom speaker programming information.

Push-button Reset
Two pins (3 and 4) of the Utility Connector provide connections for an external normally open momentary switch to manually reset the system. Pin 4 is the Reset (active low) input, and pin 3 is the logic ground.

2.6.2 Keyboard Connector (J5)

You can connect an AT (not PC) keyboard to the keyboard port. The first five pins (1-5) of connector J5 provide this function. (Note that connectors J4 and J5 are combined into one connector.) Normally, AT keyboards include a cable that terminates in a male 5-pin DIN plug for connection to an AT. Table 2-12 gives the keyboard connector pinout and signal definitions, and includes corresponding pin numbers of a normal AT DIN keyboard connector.

You can connect pin 6 of J5 to a keyboard inhibit switch. If you ground this pin, the system ignores keyboard inputs. You can use this with key switches for system security.

J5 Pin	Signal Name	DIN Pin
1	Keyboard Clock	1
2	Keyboard Data	2
3	Key pin	N/A
4	Ground	4
5	Keyboard power	5
6	Keyboard inhibit	N/A

Table 2-12 Keyboard Connector (J5)

2.6.3 Parallel Port (J6)

The Little Board/486-II includes an AT-compatible parallel port. You can use the parallel port either as a standard PC/AT printer port, or as a bi-directional data bus. Refer to Chapter 4 for information on bi-directional use of this port. Refer to Chapter 1, Section 3 for drive capability.

Connection to the parallel port is through the connector J6. Table 2-13 gives this connector's pinout and signal definitions. You can use a flat ribbon cable between J6 and a female DB25 connector. The table also gives the connections from the header pins to the DB25 connector. Table 2-14 gives manufacturer's part numbers for mating connectors.

Note

For maximum reliability, keep the cable between the board and the device it drives to 10 feet or less in length.

J6 Pin	Signal Name	Function	In/Out	DB25 Pin
1	-STROBE	Output data strobe	OUT	1
3	Data 0	LSB of printer data	I/O	2
5	Data 1	:	I/O	3
7	Data 2	:	I/O	4
9	Data 3	:	I/O	5
11	Data 4	:	I/O	6
13	Data 5	:	I/O	7
15	Data 6	:	I/O	8
17	Data 7	MSB of printer data	I/O	9
19	-ACK	Character accepted	IN	10
21	BUSY	Cannot receive data	IN	11
23	PAPER OUT	Out of paper	IN	12
25	SEL OUT	Printer selected	IN	13
2	-AUTOFD	Autofeed	OUT	14
4	ERROR	Printer error	IN	15
6	-INIT	Initialize printer	OUT	16
8	SEL IN	Selects printer	OUT	17
26	N/A	Key pin		
10,12, 14,16 18,20 22,24	GROUND	Signal ground	N/A	18-25

Table 2-13 Parallel Port Connector (J6)

Connector Type	Mating Connector
RIBBON	3M 3473-7010
DISCRETE WIRE	MOLEX HOUSING 22-55-2101 PIN 16-02-0103

Table 2-14 J6 Mating Connector

Printer Port Configuration

With SETUP, you can configure the parallel port as the primary port (LPT1), the secondary port (LPT2), or disabled. Table 2-15 lists the parallel port addresses for the primary and secondary port.

Selection	I/O Address
Primary	378-37Fh
Secondary	278-27Fh

Table 2-15 Parallel Printer Port Address Configuration

Normally, the BIOS assigns the name LPT1 to the primary parallel port, and LPT2 to the secondary parallel port (if present). However, the BIOS scans for both choices (primary and secondary) and if it only finds a secondary port, it assigns LPT1 to that one. Configure the parallel port for the primary assignment shown in Table 2-17, unless the system includes another primary parallel port.

Parallel Port Interrupt

Interrupts are seldom used with parallel ports. When they are, the convention is to use IRQ7 with the primary port (LPT1) and IRQ5 with the secondary port (LPT2). You can select the interrupt with W24. Table 2-16 shows the options.

Selection	W24
No Interrupt	OPEN
IRQ5	2/3
IRQ7	1/2

Table 2-16 Parallel Port Interrupt Selection

Bi-directional Parallel Port Use

You can use the parallel printer port as a standard AT printer port, or you can use it for general purpose programmable I/O. You can create interfaces for specialized devices with the port's input and output handshake signals, and its 8-bit bi-directional data lines. You might use it for writing data to LCD display panels, scanning custom keyboards, and so forth.

The bi-directional feature is controlled by software. Refer to Chapter 3 for typical system software configuration information, and to Chapter 4 for hardware details regarding nonstandard uses of this interface.

2.6.4 Serial Ports (J2, J3)

The Little Board/486-II provides two standard RS232 serial ports at J2 and J3. Table 2-17 gives the connector pinout and signal definitions for J2 and J3. In addition, the table indicates the pins to which each signal must be wired for compatibility with DB25 and DB9 connectors. The serial port pinout is arranged so that you can use a flat ribbon cable between the header and a standard DB9 connector. Normally PC serial ports use male DB connectors. Table 2-18 shows the manufacturer's part number for mating connectors.

Both ports support software selectable standard baud rates up to 19.2K bits/second (limited by the RS232C specification), 5-8 data bits, and 1, 1.5, or 2 stop bits. The serial ports appear at the standard port addresses of 3F8-3FFh (primary port) and 2F8-2FFh (secondary port), using interrupts IRQ4 and IRQ3, respectively.

You can use the serial ports for printers, modems, terminals, remote hosts, or other RS232C serial devices. Many devices, such as printers and modems, require handshaking in one or both directions. Consult the documentation for the device(s) you use for information about handshaking and other interface considerations.

Unique to Ampro is ROM BIOS support for using a serial console (keyboard and display) in place of the conventional video controller, monitor, and keyboard. See Chapter 3 for an explanation of the serial console option. Also unique to Ampro is ROM BIOS support for downloading a program from a host computer via a serial port. The program is then run as if it had been loaded from disk. See Chapter 3 for an explanation of the serial download option.

Connector J3 is the primary serial port. The ROM BIOS supports it as the DOS COM1 device. The secondary serial port is J2. The ROM BIOS supports it as the DOS COM2 device. It is possible to disable either or both serial ports with SETUP. If you disable the primary port and enable the secondary port, and there is no other primary port in the system, then the secondary port is installed as COM1 by the BIOS.

Pin	Signal Name	Function	In/Out	DB25 Pin	DB9 Pin
1	DCD	Data Carrier Detect	IN	8	1
2	DSR	Data Set Ready	IN	6	6
3	RXD	Receive Data	IN	3	2
4	RTS	Request To Send	OUT	4	7
5	TXD	Transmit Data	OUT	2	3
6	CTS	Clear to Send	IN	5	8
7	DTR	Data Terminal Ready	OUT	20	4
8	RI	Ring Indicator	IN	22	9
9	GND	Signal Ground	-	7	5
10	N/A	Key pin	-	-	-

Table 2-17 Serial Port Connectors (J3, J2)

Connector Type	Mating Connector
RIBBON	3M 3473-7010
DISCRETE WIRE	MOLEX HOUSING 22-55-2101 PIN 16-02-0103

Table 2-18 J3 and J2 Mating Connector

2.6.5 Floppy Disk Interface (J8)

The onboard floppy disk controller and ROM BIOS support one or two floppy disk drives in any of the standard DOS formats shown in Table 2-19.

Capacity	Drive Size	Tracks	Data Rate
360K	5-1/4 inch	40	250 KHz
1.2M	5-1/4 inch	80	500 KHz
720K	3-1/2 inch	80	250 KHz
1.44M	3-1/2 inch	80	500 KHz

Table 2-19 Supported Floppy Formats

Floppy Drive Considerations

Nearly any type of soft-sectored, single or double-sided, 40 or 80 track, 5-1/4 inch or 3-1/2 inch floppy disk drive is usable with this interface. Using higher quality drives improves system reliability. Here are some considerations about the selection, configuration, and connection of floppy drives to the Little Board/486-II.

- **Drive Interface**—The drives must be compatible with the board's floppy disk connector signal interface, as described below. Any standard PC or AT compatible 5-1/4 inch or 3-1/2 inch floppy drive will work fine.

- **Drive Quality**—Use high quality, DC servo, direct drive motor floppy disk drives.

- **Drive Select Jumpering**—Both drives must be jumpered to the second drive select. Use a floppy cable with conductors 10-16 twisted between the two drives. This is standard practice in PC-compatible systems.

- **Drive Termination**—Resistive terminations should be installed only on the drive connected to the last interface cable connector (farthest from the board).

- **Head Load Jumpering**—When using drives with a Head Load option, jumper the drive for head load with motor on rather than head load with drive select. This is the default for PC-compatible drives.

- **Drive Mounting**—If you mount the Little Board directly on a disk drive, you may need to place a thin metal shield between the disk drive and the CPU board, to reduce the possibility of electromagnetic interactions.

- **Dual-Capacity Drives**—The Little Board/486-II's ROM BIOS allows dual-capacity use of 1.2M byte and a 1.44M byte high-density drives: that is, 360K byte diskettes can be read on a 1.2M byte drive, and 720K byte diskettes can be read on a 1.44M byte drive. However, drive manufacturers recommend that you do not write to low-density diskettes using a high-density drive, as incomplete erasure may occur, causing higher error rates. (The heads on a high-density drive make narrower tracks.)

Floppy Interface Configuration

In SETUP, set the number and type of floppy drives connected to the system.

If you don't use the floppy interface, disable it in SETUP. Remove the jumpers from W20-1/2, 3/4 and W27 to free IRQ6, DACK2 and DRQ2 for use by other devices.

Floppy Interface Connector

Table 2-20 shows the pinout and signal definitions of the floppy disk interface connector, J8. The pinout of J8 meets the AT standard for floppy drive connectors. Table 2-21 shows the manufacturer's part numbers for mating connectors.

Pin	Signal Name	Function	In/Out
2	RPM/-RWC	Speed/Precomp	OUT
4	N/A	(Not used)	N/A
6	N/A	Key pin	N/A
8	-IDX	Index Pulse	IN
10	-MO1	Motor On 1	OUT
12	-DS2	Drive Select 2	OUT
14	-DS1	Drive Select 1	OUT
16	-MO2	Motor On 2	OUT
18	-DIRC	Direction Select	OUT
20	-STEP	Step	OUT
22	-WD	Write Data	OUT
24	-WE	Write Enable	OUT
26	-TRKO	Track 0	IN
28	-WP	Write Protect	IN
30	-RDD	Read Data	IN
32	-HS	Head Select	OUT
34	-DCHG	Disk Change	IN
1-33	(all odd)	Signal grounds	N/A

Table 2-20 Floppy Disk Interface Connector (J8)

Connector Type	Mating Connector
RIBBON	3M 3473-7034
DISCRETE WIRE	MOLEX HOUSING 22-55-2341 PIN 16-02-0103

Table 2-21 J8 Mating Connector

2.6.6 IDE Hard Disk Interface (J11)

The Little Board/486-II provides an interface for one or two Integrated Device Electronics (IDE) hard disk drives. IDE drives have the hard disk controller built-in. The IDE interface appears at connector J11, a 40-pin, dual-row connector. Table 2-22 shows the interface signals and pin outs for the IDE interface connector. Table 2-23 shows manufacturer's part numbers for mating connectors.

Note

For maximum reliability, keep IDE drive cables less than 18 inches long.

Pin	Signal Name	Function	In/Out
1	-HOST RESET	Reset signal from host	OUT
2	GND	Ground	OUT
3	HOST D7	Data bit 7	I/O
4	HOST D8	Data bit 8	I/O
5	HOST D6	Data bit 6	I/O
6	HOST D9	Data bit 9	I/O
7	HOST D5	Data bit 5	I/O
8	HOST D10	Data bit 10	I/O
9	HOST D4	Data bit 4	I/O
10	HOST D11	Data bit 11	I/O
11	HOST D3	Data bit 3	I/O
12	HOST D12	Data bit 12	I/O
13	HOST D2	Data bit 2	I/O
14	HOST D13	Data bit 13	I/O
15	HOST D1	Data bit 1	I/O
16	HOST D14	Data bit 14	I/O
17	HOST D0	Data bit 0	I/O
18	HOST D15	Data bit 15	I/O
19	GND	Ground	OUT
20	KEY	Keyed pin	N/C
21	RSVD	Reserved	N/C
22	GND	Ground	OUT
23	-HOST IOW	Write strobe	OUT
24	GND	Ground	OUT
25	-HOST IOR	Read strobe	OUT
26	GND	Ground	OUT
27	RSVD	Reserved	N/C
28	HOST ALE	Address latch enable	OUT
29	RSVD	Reserved	N/C

30	GND	Ground	OUT
31	HOST IRQ14	Drive interrupt request	IN

Table 2-22 IDE Drive Interface Connector (J11)

Pin	Signal Name	Function	In/Out
32	-HOST IO16	Send/receive 16-bit data	IN
33	HOST A1	Drive address 1	OUT
34	-HOST PDIAG	Pass diagnostic	IN
35	HOST AD0	Drive address 0	OUT
36	HOST AD2	Drive address 2	OUT
37	-HOST CS0	Chip select	OUT
38	-HOST CS1	Chip select	OUT
39	-HOST SLV/ACT	Drive active/drive slave	IN
40	GND	Ground	OUT

Table 2-22 IDE Drive Interface Connector (J11) (cont.)

Connector Type	Mating Connector
RIBBON	3M 3417-7040
DISCRETE WIRE	MOLEX HOUSING 22-55-2401 PIN 16-02-0103

Table 2-23 J11 Mating Connector

IDE Interface SETUP

Use SETUP to specify your IDE hard disk drive type. If you do not find a drive type whose displayed parameters match the drive you are using, use drive type 48 or 49. These two types allow you to enter the drive parameters in the parameter fields. The drive manufacturer provides the drive parameters—check the drive's documentation for the proper parameters. See the section on SETUP in Chapter 3 for additional information.

Note

The Ampro ROM BIOS allows you to use both IDE and SCSI drives on the same system.

2.6.7 SCSI INTERFACE (J7)

The Little Board/486-II features a 16-bit Small Computer System Interface (SCSI) controller compatible with SCSI-2 software and peripherals. The SCSI port uses a 50-pin male header connector (J7) to interface with peripherals. (This connector provides an 8-bit path to the peripheral device. The controller subsystem has a 16-bit path to the CPU.) Table 2-24 shows the pinout and signal definitions of this interface. Refer to your SCSI device documentation, or the ANSI X3.131 SCSI specification for detailed information on the signal functions. Be sure that the maximum SCSI bus cable length, from the board to the most distant SCSI peripheral, is less than 18 feet. If you don't use the SCSI, disable it in SETUP. Table 2-25 shows manufacturer's part numbers for mating connectors.

Pin	Signal	Function
2	-DB0	Data Bit 0 (LSB
4	-DB1	Data Bit 1
6	-DB2	Data Bit 2
8	-DB3	Data Bit 3
10	-DB4	Data Bit 4
12	-DB5	Data Bit 5
14	-DB6	Data Bit 6
16	- DB7	Data Bit 7
18	-DBP	Data Parity
26	TERM PWR	Termination +5VDC
32	-ATN	Attention
34	GROUND	Signal Ground
36	-BSY	Busy
38	-ACK	Transfer Acknowledge
40	-RST	Reset
42	-MSG	Message
44	-SEL	Select
46	-C/D	Control/Data
48	-REQ	Transfer Request
50	-I/O	Data direction
25	N/A	Key pin
1-49(odd) 20,22,24 28,30	GROUND	Signal Grounds

Table 2-24 SCSI Interface Connector (J7)

Connector Type	Mating Connector
RIBBON	3M 3425-7050
DISCRETE WIRE	MOLEX HOUSING 22-55-2501 PIN 16-02-0103

Table 2-25 J7 Mating Connector

2.6.8 Normal Use of SCSI

The SCSI interface can serve many purposes, including controlling hard disk drives, tape drives, text scanners, and printer and communications servers. The ROM BIOS supports booting DOS from a SCSI device such as a hard disk. With Ampro's ROM BIOS support, you can use any device compatible with the SCSI Common Command Set for “direct access devices”.

The Little Board/486-II Development Kit comes with a diskette containing SCSI utilities for use with DOS. It includes a powerful SCSI hard disk formatting utility that allows low-level formatting, changing the disk interleaving, and mapping out bad sectors. Refer to the Ampro Common Utilities manual for details about using the SCSI utilities.

Older versions of PC-DOS, for instance Version 3.x, requires you to divide drives larger than 32M bytes into more than one partition. Under PC-DOS or MS-DOS 3.x, you can logically divide each drive into as many as four partitions of 32 megabytes each or smaller. This allows the use of physical drives as large as 128 megabytes. MS-DOS 4.x, PC-DOS 4.x and DR DOS (any version) support a maximum drive size of 512M bytes without partitioning. Recent versions of MS-DOS remove these restrictions. The Ampro SCSI Common Command Set implementation permits partitions as large as 2G bytes. Four 2G byte partitions allow a physical drive size as large as 8G bytes.

Besides direct access, SCSI devices include sequential access devices (tape), printer devices, read-only devices (CD-ROM), and processor devices (CPUs). These device types require special application programs, utilities, or driver software not included on the Ampro Utility diskette.

Hard disk support for operating systems other than DOS may or may not be available through the ROM BIOS hard disk driver. This depends on two things: whether the operating system in question uses BIOS calls exclusively for the hard disk function; and whether the operating system has any special ROM BIOS constraints, such as reentrancy. Some operating systems—multitasking ones in particular such as UNIX—bypass the BIOS and attempt to program the hard disk controller directly. With such systems, you must modify the operating system to add an appropriate SCSI hard disk driver that can take advantage of the SCSI interface. An alternative is to use the IDE interface instead of SCSI, as IDE drives are usually supported by PC implementations of UNIX.

2.6.9 The Ampro SCSI BIOS

You can use a variety of mass storage devices with the SCSI universal bus interface and command protocols. Ampro has added a further layer of universality, the SCSI BIOS.

The SCSI BIOS, a set of low level functions in the ROM BIOS, is a hardware independent interface between system software and SCSI peripherals. The advantage of the Ampro SCSI BIOS is in interfacing to devices. Programmers can write software for SCSI devices without concern for the operational details of the SCSI interface. Also, the SCSI BIOS enables you to import software from other environments more safely, quickly, and easily.

Chapters 3 and 4 discuss the SCSI interface, and the Ampro SCSI BIOS (in the ROM BIOS) in greater detail. In addition, Application Note AAN-8804, available from Ampro, provides details of the SCSI BIOS functions.

2.6.10 SCSI Interface Configuration

Configure the SCSI interface according to your system's needs. This is covered in the following paragraphs.

Active Terminators

The SCSI interface uses active terminators for the SCSI bus. Active terminators draw less current than 330/220 ohm terminators (the normal method of termination), and are less susceptible to noise. Only the SCSI devices on each end of the SCSI bus should be terminated. To enable the SCSI terminators on the Little Board/486-II, install a jumper on W28.

External Termination Power Option

You can power external SCSI terminations from the Little Board/486-II. A jumper option (W6), connects power (+5V) to the SCSI bus TERMPWR signal (J7, pin 26). The board includes a Schottky protection diode to prevent damage to the board by current flowing *from* the SCSI bus.

The default jumpering of W6 is open; that is, termination power is not normally supplied by the Little Board/486-II.

SCSI ID

Every SCSI device must be configured for a specific SCSI bus ID, between 0 and 7. Normally, set the SCSI initiator ID to 7. Set the disk drives' and other SCSI target devices' IDs to 0, 1, and so forth. Set the SCSI ID using SETUP. Details are provided in Chapter 3.

Enabling the SCSI Interface

Using SETUP, disable or enable the SCSI BIOS services. This SETUP option disables access to the SCSI BIOS services only. The SCSI hardware can still function normally. In this situation, SCSI services are provided by an external device driver. Disabling the SCSI BIOS might be desirable for several reasons:

- To speed system booting when you don’t use SCSI. Otherwise, there is a delay while the system waits for a SCSI device.
- To disable SCSI BIOS control of the SCSI hardware, when you have a non-standard use for the interface, implemented by a device driver.

Details on using SETUP to configure the SCSI interface are provided in Chapter 3.

2.7 AT EXPANSION BUS

The PC/AT expansion bus appears on a pair of header connectors at P1 and P2. P1 is a 64-pin male dual-row header. P2 is a 40-pin male dual-row header. The PC-bus subset of the expansion bus connects to the first 62 pins of P1; the two additional pins of P1 (A32 and B32) are added grounds, to enhance system reliability. Connector P2 replaces the 36-pin edgecard connector of a conventional AT expansion bus. It has extra ground pins at each end of the connector (C0, D0, D19). (C19 is a key pin.) The layout of signals on P1 and P2 is compliant with the PC/104 bus specification. PC/104-compliant expansion modules can be installed on the Little Board/486-II expansion bus.

The buffered output signals to the expansion bus are standard TTL level signals. All inputs to the Little Board/486-II operate at TTL levels and present a typical CMOS load to the expansion bus. The current sinking rating for most output signals driving the AT expansion bus is shown in Tables 2-25 through 2-28, along with how the signals are terminated on the Little Board/486-II. Certain control signals have 33 ohms in series to limit ringing. Other signals are pulled up with resistors with values indicated in the tables. You can find further information about these signals in many publications, including the IBM technical reference manuals for the PC and AT computers, and from the reference documents listed in this manual.

2.7.1 Onboard MiniModule Expansion

You can install one or more MiniModule products on the Little Board/486-II expansion connectors. When installed on P1 and P2, the expansion modules fit within the Little Board/486-II’s outline dimensions. Ampro offers a wide variety of MiniModule products, including display controllers, serial/parallel interfaces, modems, LAN interfaces, and others. Most Ampro MiniModule products have stackthrough connectors compatible with the PC/104 Version 2.1 specification. You can stack

several modules on the headers. Each additional module increases the thickness of the package by 0.6 inches (17 mm). See Figure 2-9.

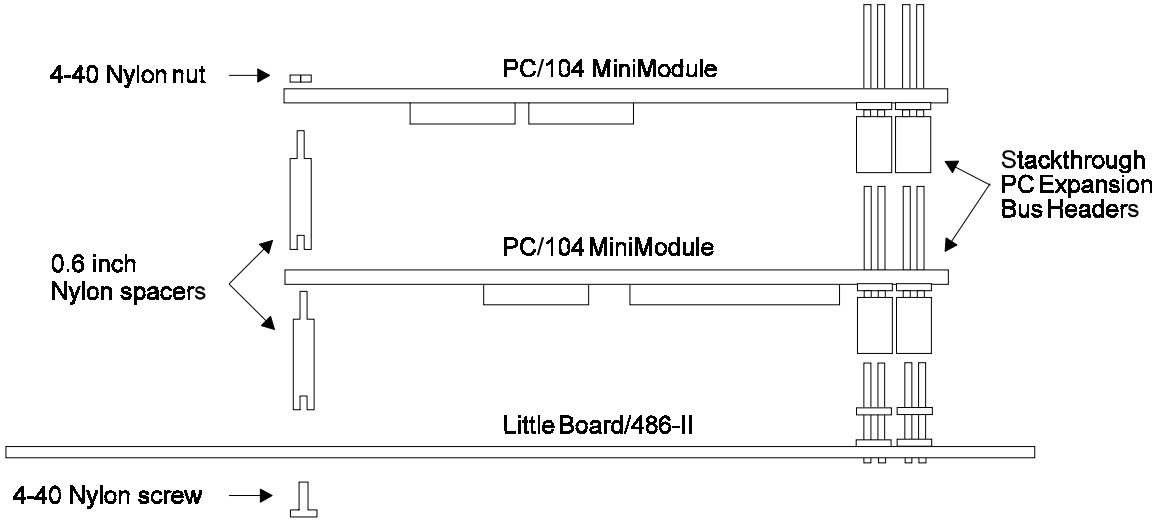


Figure 2-11 Stacking MiniModules on the Little Board/486-II

2.7.2 Using Standard PC and AT Bus Cards

Ampro offers several options that allow you to add conventional PC and PC/AT expansion cards to the Little Board/486-II system. Contact Ampro for further information on optional bus expansion products.

2.7.3 Bus Expansion Guidelines

One way to expand the AT bus is by connecting cables to the header connectors. Several options, available from Ampro and others, allow expansion with standard PC or AT bus plug-in cards. Use the following guidelines about the length and quality of the cables.

- **Cable Length and Quality**—In general, keep the bus expansion cable as short as possible. Long cables reduce system reliability.
  - For cables up to 6 inches, use a high quality standard cable, such as 3M 3365/64 (64 conductor) and 3365/40 (40 conductor).
  - For cables between 6 to 12 inches long, use a high quality ground plane cable, such as 3M part number 3353/64 (64 conductor) and 3353/40 (40 conductor).
  - Do not use cables over 12 inches long.

- **Backplane Quality**—Be sure to use a high quality backplane that minimizes signal crosstalk. Use of power and ground planes, and guard traces between bus signals will improve system reliability.

- **Eliminating Reset and TC Noise**—Some cards have asynchronous TTL logic inputs that are susceptible to noise and crosstalk. The active high RESET and TC bus lines are especially vulnerable. You can make these signals more reliable by adding a 200 pF to 500 pF capacitor between the signal and ground to prevent false triggering by filtering noise on the signals. These RESET and TC filters are included on most Ampro backplane expansion products.

- **Bus Termination**—Many backplanes include bus termination to improve system reliability by matching backplane impedance to the rest of the system. The IEEE-P996 draft specification for the AT expansion bus recommends the use of AC termination rather than resistive termination. The recommended AC termination is a 50 to 100 pF capacitor, in series with a 50 to 100 ohm resistor, from each signal to ground. Ampro provides positions for OEM addition of AC termination on most bus expansion products.

Caution

Do not use resistive bus termination! If the signal requires termination, use AC termination only.

Here are some manufacturer part numbers for 9-pin, eight-terminator devices with 100 pF capacitors in series with 100 ohm resistors:

- Dale CSRC-09C30-101J-101M
- Bourns 4609H-701-101/101

The actual requirements for signal termination depend on system configuration, interconnecting bus cable, and on the number and type of expansion modules used. It is the system engineer's responsibility to determine the need for termination.

2.7.4 Expansion Bus Connector Pinouts

Tables 2-26 through 2-29 show the pinout and signal functions on the AT expansion bus connectors. Further information about these signals is available in various publications, including reference documents listed in this manual.

The Little Board/486-II does not generate ±12VDC or -5VDC for the AT expansion bus. If devices on the AT expansion bus require these voltages, they can be supplied to the AT expansion bus connector from the utility connector (J4). Some Ampro expansion products provide for DC-to-DC converters for all bus voltages except +5VDC.

The AT bus pin numbers used in the following tables correspond to the scheme normally used on the AT expansion bus card sockets. Rather than numerical designations (1, 2, 3) they have alpha-numeric designations (A1, A2…, B1, B2…, etc.) .

Pin	Signal Name	Function	In/Out	Current	PU/PD/Ser*
A1	-IOCHCK	bus NMI input	IN	N/A	4.7K PU
A2	SD7	Data bit 7	I/O	12 mA	10K PU
A3	SD6	Data bit 6	I/O	12 mA	10K PU
A4	SD5	Data bit 5	I/O	12 mA	10K PU
A5	SD4	Data bit 4	I/O	12 mA	10K PU
A6	SD3	Data bit 3	I/O	12 mA	10K PU
A7	SD2	Data bit 2	I/O	12 mA	10K PU
A8	SD1	Data bit 1	I/O	12 mA	10K PU
A9	SD0	Data bit 0	I/O	12 mA	10K PU
A10	IOCHRDY	Processor Ready Ctrl	IN	N/A	1K PU
A11	AEN	Address Enable	I/O	12 mA	10K PU
A12	SA19	Address bit 19	I/O	12 mA	10K PU
A13	SA18	Address bit 18	I/O	12 mA	10K PU
A14	SA17	Address bit 17	I/O	12 mA	10K PU
A15	SA16	Address bit 16	I/O	24 mA	10K PU
A16	SA15	Address bit 15	I/O	24 mA	10K PU
A17	SA14	Address bit 14	I/O	24 mA	10K PU
A18	SA13	Address bit 13	I/O	24 mA	10K PU
A19	SA12	Address bit 12	I/O	24 mA	10K PU
A20	SA11	Address bit 11	I/O	24 mA	10K PU
A21	SA10	Address bit 10	I/O	24 mA	10K PU
A22	SA9	Address bit 9	I/O	24 mA	10K PU
A23	SA8	Address bit 8	I/O	24 mA	10K PU
A24	SA7	Address bit 7	I/O	24 mA	10K PU
A25	SA6	Address bit 6	I/O	24 mA	10K PU
A26	SA5	Address bit 5	I/O	24 mA	10K PU
A27	SA4	Address bit 4	I/O	24 mA	10K PU
A28	SA3	Address bit 3	I/O	24 mA	10K PU
A29	SA2	Address bit 2	I/O	24 mA	10K PU
A30	SA1	Address bit 1	I/O	12 mA	
A31	SA0	Address bit 0	I/O	12 mA	
A32	GND (**)	Ground	N/A	N/A	

*PU=pull up; PD=pull down; SER=resistance in series. All values in ohms.

**Added ground. Not needed with conventional expansion cards.

Table 2-26 AT Expansion Bus Connector, A1-A32 (P1)

Pin	Signal Name	Function	In/Out	Current	PU/PD/Ser*
B1	GND	Ground	N/A	N/A	
B2	RESETDRV	System reset signal	OUT	12 mA	
B3	+5V	+5 volt power	N/A	N/A	
B4	IRQ9	Interrupt request 9	IN	N/A	10K PU
B5	-5V	To J4 pin 8	N/A	N/A	
B6	DRQ2	DMA request 2	IN	N/A	10K PU
B7	-12V	To J4 pin 10	N/A	N/A	
B8	-ENDXFR	Zero wait state	IN	N/A	
B9	+12V	To J4 pin 8	N/A	N/A	
B10	N/A	Keyed pin	N/A	N/A	
B11	-SMEMW	Mem Write(lwr 1MB)	I/O	12 mA	33 SER
B12	-SMEMR	Mem Read(lwr 1MB	I/O	12 mA	33 SER
B13	IOW	I/O Write	I/O	12 mA	33 SER
B14	IOR	I/O Read	I/O	12 mA	33 SER
B15	-DACK3	DMA Acknowledge 3	OUT	6 mA	
B16	DRQ3	DMA Request 3	IN	N/A	10K PU
B17	-DACK1	DMA Acknowledge 1	OUT	6 mA	
B18	DRQ1	DMA Request 1	IN	N/A	10K PU
B19	-REFRESH	Memory Refresh	I/O	24 mA	330 PU
B20	SYSCLK	Sys Clock(e.g. 8MHz)	OUT	12 mA	
B21	IRQ7	Interrupt Request 7	IN	N/A	10K PU
B22	IRQ6	Interrupt Request 6	IN	N/A	10K PU
B23	IRQ5	Interrupt Request 5	IN	N/A	10K PU
B24	IRQ4	Interrupt Request 4	IN	N/A	10K PU
B25	IRQ3	Interrupt Request 3	IN	N/A	10K PU
B26	-DACK2	DMA Acknowledge 2	OUT	6 mA	
B27	TC	DMA Terminal Count	OUT	12 mA	
B28	BALE	Address latch enable	OUT	12 mA	33 SER
B29	+5V	+5V power	N/A	N/A	
B30	OSC	14.3 Mhz clock	OUT	6 mA	
B31	GND	Ground	N/A	N/A	
B32	GND**	Ground	N/A	N/A	

*PU=pull up; PD=pull down; SER=resistance in series. All values in ohms.

**Added ground. Not needed with conventional expansion cards.

Table 2-27 AT Expansion Bus Connector, B1-B32 (P1)

Pin	Signal Name	Function	In/Out	Current	PU/PD/Ser*
C0	GND**	Ground	N/A	N/A	
C1	SBHE	Bus High Enable	I/O	12 mA	
C2	LA23	Address bit 23	I/O	24 mA	
C3	LA22	Address bit 22	I/O	24 mA	
C4	LA21	Address bit 21	I/O	24 mA	
C5	LA20	Address bit 20	I/O	24 mA	
C6	LA19	Address bit 19	I/O	24 mA	
C7	LA18	Address bit 18	I/O	24 mA	
C8	LA17	Address bit 17	I/O	24 mA	
C9	-MEMR	Memory Read	I/O	12 mA	33 SER
C10	-MEMW	Memory Write	I/O	12 mA	33 SER
C11	SD8	Data Bit 8	I/O	12 mA	10K PU
C12	SD9	Data Bit 9	I/O	12 mA	
C13	SD10	Data Bit 10	I/O	12 mA	
C14	SD11	Data Bit 11	I/O	12 mA	
C15	SD12	Data Bit 12	I/O	12 mA	
C16	SD13	Data Bit 13	I/O	12 mA	
C17	SD14	Data Bit 14	I/O	12 mA	
C18	SD15	Data Bit 15	I/O	12 mA	
C19	Key	Key Pin	N/A	N/A	

*PU=pull up; PD=pull down; SER=resistance in series. All values in ohms.

**Added ground. Not needed with conventional expansion cards.

Table 2-28 AT Expansion Bus Connector, C0-C19 (P2)

Pin	Signal Name	Function	In/Out	Current	PU/PD/Ser*
D0	GND**	Ground	N/A	N/A	
D1	-MEMCS16	16-bit Mem Access	IN	N/A	330 PU
D2	-IOCS16	16-bit I/O Access	IN	N/A	330 PU
D3	IRQ10	Interrupt Request 10	IN	N/A	10K PU
D4	IRQ11	Interrupt Request 11	IN	N/A	10K PU
D5	IRQ12	Interrupt Request 12	IN	N/A	10K PU
D6	IRQ15	Interrupt Request 15	IN	N/A	10K PU
D7	IRQ14	Interrupt Request 14	IN	N/A	10K PU
D8	-DACK0	DMA Acknowledge 0	OUT	6mA	
D9	DRQ0	DMA Request 0	IN	N/A	
D10	-DACK5	DMA Acknowledge 5	OUT	6mA	
D11	DRQ5	DMA Request 5	IN	N/A	
D12	-DACK6	DMA Acknowledge 6	OUT	6mA	
D13	DRQ6	DMA Request 6	IN	N/A	
D14	-DACK7	DMA Acknowledge 7	OUT	6mA	
D15	DRQ7	DMA Request 7	IN	N/A	
D16	+5 V	+5 Volt Power	N/A	N/A	
D17	-MASTER	Bus Master Assert	IN	N/A	330 PU
D18	GND	Ground	N/A	N/A	
D19	GND**	Ground	N/A	N/A	

*PU=pull up; PD=pull down; SER=resistanDe in series. All values in ohms.

**Added ground. Not needed with Donventional expansion Dards.

Table 2-29 AT Expansion Bus Connector, D0-D19 (P2)

2.7.5 Interrupt and DMA Channel Usage

The AT bus provides several interrupt and DMA control signals. When you expand the system with MiniModules or plug-in cards that require either interrupt or DMA support, you must select which Interrupt or DMA channel the added device uses. Typically this involves switches or jumpers on the module. It is important that you configure the added module to use an Interrupt or DMA channel not already in use. For your convenience, Tables 2-30 and 2-31 provide a summary of the normal assignment of interrupt and DMA channels on the Little Board/486-II.

Interrupt	Function
IRQ0*	ROM BIOS clock tick function, from Timer 0
IRQ1*	Keyboard interrupt
IRQ2*	Cascade input for IRQ8-15
IRQ3	Secondary serial port (if present)
IRQ4	Primary serial port (if present)
IRQ5	Reserved for secondary parallel printer
IRQ6	Floppy controller (if present)
IRQ7	Parallel printer (if present)
IRQ8*	Reserved for battery-backed clock alarm
IRQ9**	EGA or VGA controller (if present)
IRQ10	Available
IRQ11	Available
IRQ12	Available
IRQ13*	Reserved for coprocessor
IRQ14	Hard disk controller
IRQ15	Available
* Unavailable on PC/AT bus.	
** Corresponds to IRQ2 on a PC's expansion bus.	

Table 2-30 Interrupt Channel Assignment

Channel	Function
0	Available for 8-bit transfers
1	Available for 8-bit transfers
2	Floppy controller (if present)
3	Available for 8-bit transfers
4	Cascade for channels 0-3
5	Available for 16-bit transfers
6	Available for 16-bit transfers
7	Available for 16-bit transfers

Table 2-31 DMA Channel Assignment

CHAPTER 3

SOFTWARE CONFIGURATION

3.1 INTRODUCTION

This chapter provides an overview of the system features, configuration options, and utilities that are available under a disk operating system and discusses configuration of the Little Board/486-II using the SETUP function. A combination of standard DOS and Ampro-supplied utilities and drivers allows you to create a highly customized embedded computer system based on the Little Board/486-II.

This manual presumes you have some familiarity with DOS (PC-DOS, MS-DOS, or DR DOS). It does not attempt to describe the standard DOS and ROM BIOS functions. Refer to the appropriate DOS and PC reference manuals for further information on the DOS, its drivers and utilities, and the ROM BIOS. Where Ampro has added to or modified standard functions, these will be described. Note that the Ampro Common Utilities manual contains detailed descriptions of the Ampro driver and utility programs (for instance, the SCSI utilities).

3.2 OPERATION WITH DOS

The Little Board/486-II's ROM BIOS allows the use of IBM's PC-DOS or Microsoft's MS-DOS, Version 3.3 or later, or of any version of Digital Research's DR DOS as the disk operating system. Throughout this chapter, "DOS" refers to any of these operating systems. Any differences between these similar operating systems are noted in the text where applicable.

Caution

Sometimes MS-DOS is customized by a manufacturer for a specific system and may not work on the Little Board/486-II. Use DR DOS (supplied by Ampro), IBM PC-DOS (supplied by IBM), or the generic version of MS-DOS (supplied by Microsoft on an OEM basis).

Serial Ports—DOS normally supports the board's two 16C450-compatible RS232C serial ports as the COM1 and COM2 ports.

Parallel Port—The Parallel Printer port is normally the DOS LPT1 device.

Real Time Clock—The ROM BIOS maintains the real-time clock. It is incremented approximately 18.2 times per second by an interrupt from timer/counter 0. The ROM BIOS automatically initializes it, upon system reset or powerup, to the correct time and date stored in the onboard battery-backed clock.

IDE Port—Older versions of DOS require you to divide disk drives larger than 32M bytes into more than one partition. They will allow four partitions, enabling the use of physical drives as large as 128M bytes. MS-DOS or PC-DOS Versions 4.x, or any version of DR DOS support hard disks as large as 512M bytes. DOS 6.x supports drives as large as 2G bytes, and third party products can be added that support even larger drives.

SCSI Port—Like with the IDE port, older versions of DOS require you to divide disk drives into partitions. The same rules apply for each version of DOS as for the IDE interface.

3.2.1 Configuration Options

All of the subsystems on the Little Board/486-II can be configured using options provided by hardware jumpers and by the SETUP program (provided by Ampro, the disk operating system (and its drivers and utilities) and the Ampro Common Utilities. In addition, expansion boards such as LAN adapters, modems, etc. usually have their own configuration programs or jumpers, described in their own technical manuals.

The following section briefly summarizes the features of the software included in the Ampro Little Board/486-II SETUP function, and the Ampro utilities. The Ampro Utilities manual contains detailed descriptions of the programs supplied on the Ampro Utilities diskette.

3.2.2 EMS Option

The Little Board/486-II can emulate the Lotus-Intel-Microsoft Expanded Memory Specification Version 4.0 (LIM EMS 4.0), with the memory management capability of the 80486DX2/4 CPU, under control of a device driver. Such drivers are available with the newer versions of DOS.

3.3 UTILITY SOFTWARE OVERVIEW

The Little Board/486-II Development Kit provides a number of software utilities on a diskette called the Common Utilities diskette. Some of the programs provided on this diskette are:

- **SETUP**—A utility used to access the ROM BIOS SETUP function from the DOS command line.
- **SCSICOMP**—A SCSI utility that compares data from two SCSI direct access devices.
- **SCSICOPY**—A SCSI copy utility that copies data between two SCSI direct access devices
- **SCSIFMT**—A hard disk utility for low level SCSI drive formatting.
- **SCSITOOL**—A SCSI debugger that issues low level commands to any SCSI bus device.
- **SERLOAD**—A serial loader utility for downloading files from a remote host prior to system boot.
- **SERPROG**—A utility to program byte-wide devices from a serial port.
- **TVTERM**—A Televideo 900-series terminal emulator.
- **WATCHDOG** — Used to stop, start, or retrigger the watchdog timer function.

These utilities and others are described in the Ampro Common Utilities manual.

3.4 THE AMPRO SETUP FUNCTION

Use the Ampro SETUP function to initialize or modify the contents of the nonvolatile Configuration Memory. The Configuration Memory comprises the NOVRAM in the battery-backed clock chip, and the Configuration EEPROM. SETUP takes effect at system boot time. Note that, should the backup battery fail, the contents of the EEPROM are used to initialize the board. The only information not available would be the time of day and date.

The SETUP function is located in the ROM BIOS. It can be accessed using CTRL-ALT-ESC while the computer is in the Power On Self Test (POST) when booting up. The screen will display a

message indicating when you can enter CTRL-ALT-ESC to access the SETUP function. You may also enter the SETUP function using the SETUP.COM program (provided on the Ampro Common Utilities diskette) from the DOS command line.

The SETUP function displays four pages of setup information. Table 3-1 shows which SETUP choices are on each page.

Page	Menu Name	Functions
1	Standard (CMOS/EEPROM) Configuration	Date and time; floppies; IDE hard disks; video; RAM; error halt; video shadow RAM
2	Options/Peripheral Configuration	Extended BIOS; serial ports; parallel port; byte-wide sockets; floppy enable; IDE enable; serial boot loader; watchdog timer; hot key SETUP enable
3	Extended SCSI and Hard Disk configuration	SCSI parameters; SCSI disk maps; DOS disk maps
4	Extended Serial Console Configuration	Console input and output device configuration
* SETUP pages 3 and 4 are available when you enable Extended BIOS from SETUP page 2, LB/486-II Options/Peripheral Configuration.		

Table 3-1 Functions on Each SETUP Page

3.5 SETUP PAGE 1—STANDARD CONFIGURATION

This section describes the options on the first page of the SETUP function.

3.5.1 Date and Time

Enter the correct date and time in the provided fields. The time shown on the screen is continuously updated by the SETUP function and reflects the current state of the real-time clock. The new time that you enter is immediately written to the real-time clock.

3.5.2 Floppy Drives

The ROM BIOS supports all of the popular DOS-compatible floppy disk formats. This includes all the 5-1/4 inch and 3-1/2 inch floppy formats—360K, 720K, 1.2M, and 1.44M. (Note that some formats are not supported by early versions of DOS.) In addition, the ROM BIOS supports dual capacity use of high density floppy drives. That is, you can read and boot from 360K floppies in a 1.2M 5-1/4 inch drive, and from 720K floppies in a 1.44M 3-1/2 inch drive.

When the Little Board/486-II is shipped, the Configuration Memory is initialized to expect the first physical floppy drive (A) to be a 360K 5-1/4 inch drive. However, the ROM BIOS automatically adapts to whatever type of drive you have connected as drive A. Therefore, you can boot your system from any standard drive.

Drive Parameter Setup

The Configuration Memory contains the information on the number and type of floppy drives in the system. If the drives connected to the system do not match the parameters in the Configuration

Memory, an error message is displayed indicating the mismatch and instructing you to Run SETUP. Once you have booted DOS, you can reconfigure the system for your floppy drive configuration with SETUP.

Single-Floppy Configurations

A handy feature of DOS is built-in support for single-drive systems. If you specify one floppy in SETUP, DOS automatically assigns drive letters A and B to that drive. You can copy files between two diskettes on one drive; DOS will prompt you to change diskettes when needed.

3.5.3 IDE Hard Disk Drives

The ROM BIOS supports one or two hard disk drives connected to the IDE interface. You can use IDE drives exclusively, or with SCSI hard disk drives. In DOS applications, the total number of drives used (SCSI and IDE) can be up to eight under DR DOS, but is limited to two under older versions of PC-DOS or MS-DOS. Check the documentation for the DOS you use.

One advantage of using the IDE interface is that you can use operating systems or programs that are designed to talk directly to AT hard disk controllers. This includes multitasking operating systems such as UNIX, Xenix, QNX, etc. To configure the system for use with one or two IDE drives, set the drive parameters with SETUP, as outlined here:

- **Drive Types**—The Configuration Memory contains the parameters that specify the physical format of the each IDE drive. It includes the total number of cylinders, number of heads, cylinder to begin precompensation, landing zone cylinder number, and the number of sectors per cylinder. The drive manufacturer supplies these parameters. If no built-in drive type matches your drive, select drive type 48 or 49 and enter the drive parameters in the fields provided.

- **Drive Selection**—besides specifying the physical characteristics of each IDE drive, you also must specify how they are to be used by the ROM BIOS. Use the SETUP Extended SCSI and Hard Disk Configuration menu (third menu page) for this. Specify one or two IDE drives in the DOS disk map as DOS drives to be installed by the ROM BIOS at system boot time.

- **BOOT Device Specification**—The choice to boot from hard or floppy disk is on the Extended Options/Peripheral Configuration menu in SETUP. Select either Floppy Drive or DOS Hard Drive. When you select DOS Hard Drive, the 1st Hard Disk in the DOS Disk Map becomes the boot device.

Once you have set the system's Configuration Memory, the IDE drive(s) can be formatted and otherwise prepared normally. Refer to your operating system and disk drive documentation for specific procedures and requirements.

3.5.4 Video

Specify the initial video mode. Select MDA, Hercules, CGA, EGA, or VGA. If your video display card is VGA or SuperVGA, specify VGA no matter how it is configured to come up.

3.5.5 DRAM Memory

The ROM BIOS automatically sets the amount of memory during POST. If you change the amount of memory installed on the board, however, you must run SETUP and save the new memory size. Until you do this, an error message will appear during POST.

3.5.6 Error Halt

Select which kinds of errors will halt the power on self test.

3.5.7 Video Shadow RAM

This option, when enabled, allows the ROM BIOS to copy the contents of a video BIOS into DRAM, so that video BIOS accesses are speeded up. The actual video BIOS ROM on the video controller is disabled, and DRAM is mapped into the address space it occupied.

3.6 SETUP PAGE 2—OPTIONS/PERIPHERAL CONFIGURATION

3.6.1 Extended BIOS

Normally, the Ampro Extended BIOS is enabled. This allows access to SETUP pages three and four. If you do not want to use the BIOS extensions, you can disable them using this parameter.

3.6.2 Serial Ports

This section discusses use of the two RS232C serial ports, and gives some examples of typical installations.

Serial Port Initialization

Before you can use the board's serial ports, they must be enabled and properly initialized. Enable/disable is handled by the SETUP function. You can enable or disable either port with the Serial Port 1 and Serial Port 2 parameters. (When you use SETUP to enable or disable a port, the change does not take effect until you reboot the system.)

At boot time, DOS initializes the serial ports, assigning them their COM port designations and their communication parameter settings. Although this might vary with different types and versions of DOS, typical communication parameter settings are 2400 baud, even parity, 7 bits, and 1 stop bit.

Usually an application program that uses a serial port will access the port's hardware and reinitialize the communication parameters to other values, based on settings that the user has entered when configuring the application program.

Note

COMn (n=1, 2) is a logical designation, not a physical value. When the system boots, the ROM BIOS scans both serial port addresses, and installs the first port it finds as COM1. If it finds a second one, it installs that one as COM2. If you disable serial port 1, then serial port 2 becomes COM1, not COM2. There must be a COM1 elsewhere in the system for the onboard secondary serial port to become COM2.

3.6.3 Using a Serial Modem

You can use either of the RS232C ports as a modem interface. You will not need to concern yourself with serial port initialization since most PC communications programs control the serial port

hardware directly, rather than using DOS or ROM BIOS functions. If your program does not do this, use the DOS MODE command to initialize the port.

When installing a modem, be sure to connect appropriate input and output handshake signals, depending on what your communications software requires. Standard serial modem cables are commonly available that connect all of the proper signals correctly. The signal arrangement on the serial port connectors J2 and J3 are arranged so that a straight-through cable with a ribbon cable connector on one end and a ribbon cable-compatible DB9 on the other end provides the correct wiring for a standard PC modem cable.

Many powerful communications programs are available to control modem communications. Some of these programs offer powerful “script” languages that allow you to generate complex automatically functioning applications with little effort.

3.6.4 Parallel Printer

No special configuration is required to use the system with a PC compatible parallel printer. SETUP allows you to enable the parallel port as the primary, or secondary port, or to disable it entirely. Most application software uses LPT1 as the default printer port. If you enable the port, printing to it is automatic.

The following DOS commands can be used to test printing with the parallel printer:

`A>COPY CONFIG.SYS LPT1`	Prints contents of CONFIG.SYS
`A>DIR >LPT1`	Prints the directory

In addition, the <PrtSc> (Print Screen) key will print the contents of the video screen to the LPT1 device. Also, you can use the Printer Echo function to print all characters written to the console. The command <Ctrl-P> enables the Printer Echo function. Entering <Ctrl-P> again disables Printer Echo.

Note

LPTn (n=1, 2, 3) is a logical designation, not a physical value. When the system boots, the ROM BIOS scans both parallel port addresses, and installs the first one it finds as LPT1. If it finds a second one, it installs that one as LPT2. Changing the port's designation to secondary with SETUP does not change it from LPT1 to LPT2. There must be an LPT1 elsewhere in the system for the onboard parallel port to become LPT2.

3.6.5 Floppy Interface Enable

This SETUP parameter allows you to enable or disable the onboard floppy interface. When disabled, the I/O ports assigned to the floppy controller are freed, allowing them to be used by other devices installed on the PC expansion bus.

3.6.6 IDE Interface Enable

This SETUP parameter allows you to enable or disable the onboard IDE hard disk interface. When disabled, the I/O ports assigned to the IDE controller are freed, allowing them to be used by other devices installed on the PC expansion bus.

3.6.7 Mono/Color Jumper

On most PC motherboards, there is a Mono/Color jumper. This is the software-equivalent of that jumper. If you use an MGA or Hercules video display, set this parameter to Mono. If you use EGA, VGA, or SuperVGA, set this parameter to Color, even if you plan to run these displays in a monochrome mode.

3.6.8 Byte-wide Sockets Configuration

Each byte-wide socket, S0, S1, and S2 can be independently configured for its starting address and the size of the memory block it resides in. If the sockets are disabled, they do not appear in the memory address space of the computer. Only one socket is enabled at a time. You can specify which socket is enabled at boot time with the Default Socket parameter.

You must also set hardware jumpers to configure the byte-wide sockets for the devices you install in them. Refer to Chapter 2 for jumper positions. If you are using the byte-wide sockets for SSD (Solid State Disk), using Ampro's Solid State Disk software, follow the directions for setting the byte-wide sockets that are in the SSD Technical Manual.

3.6.9 Serial Boot Loader Enable

This parameter enables or disables the Serial Boot Loader option in the Ampro ROM BIOS. A description of the Serial Boot Loader is provided in Chapter 4. If you are not using the Serial Boot Loader, set this parameter to "Disable".

3.6.10 Watchdog Timer Configuration

This parameter allows you to set the time duration of the watchdog timer for monitoring the boot process. You can set it to 30, 60, or 90 seconds, or you can disable this function. A description of the watchdog timer function is provided in Chapter 4.

3.6.11 Hot Key Setup Enable

In some cases with embedded systems, you do not want an end-user to be able to use the hot key sequence (CTRL-ALT-DEL) to enter SETUP. You can enable or disable hot key access to SETUP with this parameter.

3.7 SETUP PAGE 3—SCSI HARD DISK

One unique feature of the Little Board/486-II is that its ROM BIOS contains hard disk support functions that map to the Small Computer System Interface (SCSI), as well as the IDE drive interface. The SCSI interface features interchangeable peripherals, flexible configuration, and easy system upgrading and support.

SCSI hard disks are available to DOS through standard ROM BIOS functions (INT 13). These are in the SCSI BIOS, part of the ROM BIOS. The ROM BIOS hard disk support allows direct system booting from SCSI Common Command Set direct access devices. Other types of SCSI direct access devices can be used to provide a compatible hard disk function. These include SCSI RAM disks, optical disks, tape drives, and other peripherals.

Most DOS applications run normally in this SCSI-based hard disk environment. Programs nearly always use either DOS or ROM BIOS functions for disk drive access. It is extremely rare for DOS environment software to attempt to access hard disk controller hardware directly. If a program does require disk controller hardware access, it will need to be modified to use the board's SCSI hardware or SCSI-BIOS functions. You can use low level SCSI functions (in the ROM BIOS) to simplify the task.

A combination of Ampro and DOS utilities are used in the formatting and preparation of SCSI hard disk drives. Utilities for SCSI drive formatting and parking, and other SCSI functions are included on the Ampro Common Utilities diskette, and discussed in the Ampro Common Utilities manual.

3.7.1 SCSI Drive Parameter Setup

Several SCSI drive parameters need to be set in the Configuration Memory with SETUP:

- **SCSI Initiator ID**—the Ampro Little Board/486-II is the SCSI Initiator in its transactions with SCSI Target devices such as hard disk drives. Every SCSI device (Target or Initiator) must have a unique ID between 0 and 7. Set the CPU ID to 7, because this is the highest priority ID, and the SCSI BIOS resets the SCSI bus on system powerup or reset if the CPU's ID is 7.
- **SCSI Target Device IDs and LUNs**—the specification of SCSI target device IDs and Logical Unit Numbers (LUNs) are stored in the Configuration Memory. The SCSI ID for Target devices can be 0 to 6 (since the CPU is set for ID 7). This is usually set by jumpers or switches on the device. Unless you operate multiple devices from a single SCSI Target controller, use LUN 0. For example, a typical system with a SCSI drive has the drive configured as ID0, LUN0. If you have multiple SCSI drives, the ROM BIOS will install them as multiple DOS hard disk drives when the system boots, controlled by SETUP entries in the SCSI Disk Map and DOS Disk Map, described below.
- **BOOT Device Specification**—the choice to boot the system from a hard or floppy drive is stored in the Configuration Memory. Select the drive you want to boot from using the Default Boot Device parameter, described below. When you select Hard Drive, the drive shown on this menu as 1st Hard Disk becomes the boot drive.
- **SCSI Disk I/O Retries**—you can specify the number of read/write retries when using SCSI drives as DOS drives.
- **SCSI BIOS Services Enable/Disable**—using SETUP, you can enable or disable the SCSI functions in the ROM BIOS with SETUP. If you use SCSI under DOS, you must enable this option. Disabling the SCSI BIOS services will speed up system booting when you don't use the SCSI port.

Note

SETUP screen 1, "Standard (CMOS/EEPROM) Setup" is used for defining hard drives connected to the IDE interface, or connected with conventional AT bus hard disk controller boards. Do not use this menu to define disk drive parameters for SCSI-connected drives.

3.7.2 SCSI Drive Preparation for DOS Use

To use a hard disk drive on the SCSI port, you must properly connect and jumper it, set the appropriate parameters in SETUP, and then format the drive for use with DOS.

Here is a procedure you can use to prepare (format and partition) a SCSI hard disk drive for use with DOS:

1. **Set the SCSI Device IDs**—using the options in SETUP's Extended SCSI and Hard Disk Configuration menu, specify the appropriate SCSI device IDs for both the drive and the Little Board/486-II. This ID must match jumpers on the drive. If you set the CPU board to SCSI ID 7, a SCSI bus Reset will be issued on system powerup or system reset. Typically, SCSI drives come preset to SCSI ID 0, LUN 0. The "SCSI Initiator ID" option sets the CPU board's SCSI Initiator ID. The SCSI Disk Map options are used to specify the ID and LUN of up to seven SCSI drives. DOS Disk Map options assigns SCSI devices as DOS drives. For example, in a system with one SCSI drive, set SCSI Initiator to 7, SCSI Disk 1 to Id 0, LUN 0, and 1st Hard Disk to SCSI Disk 1.

2. **Low Level Formatting**—The low level format erases all data from the drive and prepares it for use. Often, the low level format function eliminates bad blocks from the usable area of the drive. It replaces bad blocks with spare ones when possible. Most manufacturers of embedded SCSI drives format them prior to shipment. Still, it is best to perform a low level format prior to using a drive for the first time. Use the Ampro SCSIFMT utility to perform the low level format of the drive. The drive must be SCSI Common Command Set (CCS) compatible, to be formatted by the SCSIFMT utility.

3. **Drive Partitioning**—Reboot the system from a floppy diskette in drive A containing the operating system, and run the DOS FDISK utility as described in your DOS documentation. You may be creating one or multiple partitions, depending on the size of the drive and the partition limitations of the particular DOS you are using.

4. **Final Preparation for System Access**—Again, reboot the system from a floppy diskette in drive A. What you do next depends on which operating system you use.

   DR DOS: Run the DOS SYS command to copy the operating system to the hard disk drive(s) that you have created in the above steps. Finally, copy anything else you need to the drive(s), and then reboot the system without the floppy diskette in drive A to verify that you have installed everything properly.

   PC-DOS or MS-DOS: Use the FORMAT /S command to copy the operating system to the DOS boot drive (drive C); or the FORMAT command for drives or drive partitions other than the DOS boot partition. Finally, copy anything else you need to the drive(s), and then reboot the system without the floppy diskette in drive A to verify that you have installed everything properly.

3.7.3 SCSI Disk Map

Enter the SCSI ID and LUN for each SCSI drive in your system in the SCSI Disk Map. You may enter the drives in any order.

3.7.4 DOS Disk Map

The DOS Disk Map allows you to assign the order of any disk drive, IDE or SCSI, to DOS. The first hard disk will be drive C:, the second hard disk will be drive D:, and so on.

You can specify which drive, Floppy A:, or the first hard drive, drive C: will be the primary boot device, using the Default Boot Device parameter. Select "Hard Disk" or "Floppy".

3.8 SETUP PAGE 4—SERIAL CONSOLE

You may want to substitute an RS232C serial device (terminal, remote computer, and so forth) for the standard video controller, monitor, and keyboard normally used as the DOS console device. The Ampro ROM BIOS has been enhanced to support serial input and output.

To use this feature, connect the serial console device to one of the serial ports. Use SETUP to configure the Little Board/486-II to use its serial console support feature. The Configuration Memory stores serial console support parameters.

Caution

Be careful when changing the console configuration using SETUP. If you specify "None" for console input and output, there will be no console access to the system. (You can recover from this state by removing the serial console plug from the primary serial port connector and shorting pins 7/8 before applying power.)

SETUP provides separate configurations for serial console input and output. Thus, you can use a serial port (and attached serial device) for either or both input and output. For instance, you can use a modem or other serial device for input, and a standard video display for output. Or you can use a standard AT keyboard with a serial display, or use a standard ASCII terminal for both input and output.

To use an ASCII terminal as the console device for your system, set both the input and output parameters to Serial Port 1, and set the serial baud rate, data length, and stop bits to match the setting of your terminal. For proper display of SETUP and POST messages from the BIOS, you must use IEEE compatible terminals that implement certain cursor commands. The commands required and their hexadecimal codes are given in Table 3-2.

Hex	Command
08	Backspace
0A	Line Feed
0B	Vertical Tab
0C	Non-destructive Space
0D	Carriage Return

Table 3-2 Required Commands

After booting this system, the keyboard and screen of the terminal become the system console. The programs you use this way must use ROM BIOS video functions (rather than direct screen addressing) for their display I/O. You can enter keyboard data from both the external serial device and the standard AT keyboard. You can revert to the standard keyboard and monitor by removing the plug from the serial port connector and shorting pins 7/8 of that connector.

Note

You cannot use DOS programs that write directly to video RAM on a serial console device.

When the system boots DOS, it initializes the serial ports to 2400 baud (typical). To preserve the serial port parameters stored in SETUP, the ROM BIOS deletes the console port(s) from the internal COM port table, normally used by DOS to locate the serial ports. With the port(s) deleted from the COM port table, DOS cannot change the parameters entered in SETUP for the serial console. (There is an option in SETUP to retain the console ports in the COM table, thought this is seldom used.) If you use a serial console, be sure to select the option that deletes the console ports from the COM Table.

The serial console device data format and the Little Board/486-II serial port data format must match for the devices to properly communicate. In addition, the hardware handshake behavior must be compatible. Normally, a serial port's Data Set Ready (DSR) and Clear To Send (CTS) input handshake signals must be true (active) for the ROM BIOS to send data out. On the Little Board/486-II, the hardware handshake can be enabled or disabled with SETUP. When hardware handshaking is enabled, be sure to connect the DSR and CTS signal inputs to appropriate handshake signals on the external serial device's interface connector. As an alternative, loop the Little Board/486-II CPU's serial output handshake signals to its input signals as follows:

- DTR (out) to DSR (in) (J3 or J2, pin 2 to pin 7)
- RTS (out) to CTS (in) (J3 or J2, pin 4 to pin 6)

3.9 THE SETUP.COM PROGRAM

You can use the SETUP.COM utility from the command line to access the same SETUP function as CTRL-ALT-ESC. In addition, SETUP.COM also adds additional functionality, such as the ability to load and store configuration settings to a disk file. This same feature is used to store up to 512 bits of OEM information in the Configuration Memory EEPROM.

3.9.1 Creating Configuration Files with SETUP.COM

The Ampro SETUP utility, SETUP.COM, offers the following options for command line entry:

```
SETUP [-switches]  [ @file.ext  |  Wfile.ext ]
```

The supported switches and their meaning are as follows:

? Display a usage help screen

T Set the (hardware) real-time clock time and date from the current DOS time and date

@file.ext Writes the specified file to the board's Configuration EEPROM. Drive and path are optional in the file name.

Wfile.ext Write NOVRAM and EEPROM contents to the file specified. The file name may contain an optional drive and path.

You can save a copy of the current contents of the board's Configuration Memory to a disk file by using the W switch. The data saved includes the entire contents of the nonvolatile configuration EEPROM, except the current time and date. The first 512 bits are the SETUP information, the last 512 bits are available for OEM storage (See Ampro Application Note AAN-8805). The file you create with this menu option can be used as a source for programming the Configuration Memory of a Little Board/486-II at a later time.

Following is an example of a configuration file's contents:

```
NOVRAM 00:  24 00 35 00 08 00 07 05 - 12 89 26 02 50 80 00 00
NOVRAM 10:  12 00 00 00 41 80 02 80 - 01 00 00 00 00 00 00 00
NOVRAM 20:  00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 01 56
NOVRAM 30:  80 01 19 80 00 00 00 00 - 00 00 00 00 39 00 38 E8
EEPROM 00:  EB 33 10 19 19 1A A8 BF - F4 FF FF FF FF FF 3E 00
EEPROM 10:  81 E1 FF FA FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 20:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 30:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 40:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 50:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 60:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF FF
EEPROM 70:  FF FF FF FF FF FF FF FF - FF FF FF FF FF FF FF 00
```

For information regarding the file's content and format, contact Ampro Technical Support.

As an example, the following command initializes the EEPROM values with a previously saved configuration:

```
C>SETUP  @SYSTEM.A
```

Assuming you created the file SYSTEM.A with SETUP's write option, SETUP will initialize the EEPROM Configuration Memory using the contents of SYSTEM.A.

Note

Word addresses not specifically named are not changed. For example, if you have a file with an entry for EEPROM 20, only the entry at that location is changed. All other contents of Configuration Memory remain the same.

Using SETUP with the write and read parameters can be useful when many boards must be initialized automatically. Another use for this SETUP mode might be to change between several predefined system configurations.

CHAPTER 4

ADVANCED TOPICS

4.1 INTRODUCTION

Except for the unique functions discussed in this chapter, the Little Board/486-II is functionally identical, in terms of hardware and software, with a PC-compatible 486DX2 or 486DX4 computer. This chapter briefly discusses standard PC/AT features and functions. However, it focuses on the non-standard functions unique to the Ampro Little Board/486-II.

For detailed technical information on the architecture and functions of a PC and its normal software environment, consult publications such as:

- **Intel Microprocessors**—80486DX/DX2 Data Sheet
- **Intel Microprocessors**—80486DX4 Data Sheet
- **Microsoft Disk Operating System**
  Microsoft Corporation
  One Microsoft Way
  Redmond, WA 98052-6399
- **The Peter Norton Programmer's Guide to the IBM PC**
  Microsoft Press
  A Division of Microsoft Corporation
  10700 Northrup Way
  Box 97200
  Bellevue, Washington 98009
- **Interfacing to the IBM Personal Computer**
  Lewis C. Eggebrecht
  Howard W. Sams & Co., Inc.
  A Subsidiary of Macmillan, Inc.
  4300 West 62nd Street
  Indianapolis, IN 46268 USA
- **Personal Computer Bus Standard P996 (Draft)**
  IEEE
  445 Hoes Lane
  Piscataway, NJ 08854
- **Adaptec AIC-6360 data sheet**
  Adaptec
  691 South Milpitas Blvd.
  Milpitas, CA 95035

Many of the devices on the Little Board/486-II are programmable. The following explanations presume use of the standard Ampro ROM BIOS.

4.2 OVERALL ARCHITECTURE

The Little Board/486-II is essentially a complete PC/AT compatible system, including the equivalent of a motherboard and three or four expansion cards. It is constructed on a single board with the same dimensions as a 5-1/4 inch floppy disk drive.

The following sections discuss the Little Board/486-II's onboard subsystems. These features are grouped as standard AT functions and unique functions, those not found in standard AT systems.

4.2.1 Standard AT System Functions

The following functions are compatible with the equivalent standard functions found on PC-compatible desktop systems:

- **AT Motherboard Logic**—this includes the CMOS 80486DX2 or 80486DX4 CPU, system and extended memory (up to 32 megabytes), ROM BIOS, DMA controllers, interrupt controllers, system clocks, programmable timers, keyboard interface, speaker port, battery-backed real-time clock and CMOS RAM.
- **AT Expansion Bus**—standard 8 MHz PC/AT bus expandable with Ampro MiniModules and other PC/104-compliant expansion modules.
- **Floppy Disk Controller**
- **Serial Controllers**
- **Parallel Controller**
- **Integrated Device Electronics (IDE) Hard Disk Interface**

4.2.2 Unique Functions

The following devices and subsystems are unique to Ampro and are not typically available on desktop systems:

- **Small Computer System Interface (SCSI) Host Adapter**—a full-function ANSI X3.131 compatible asynchronous SCSI bus interface. The board's ROM BIOS supports one or more SCSI direct access devices as AT compatible hard disks. Ampro's implementation of its SCSI controller closely matches SCSI host adapters available as expansion boards for desktop systems, but its integration into the board's ROM BIOS is unique.
- **Byte-wide Memory Sockets**—three byte-wide sockets, a 32-pin DIP and two 32-pin PLCCs. All support Flash EPROMs from 32K to 512K and EPROMs from 128K to 1M. The DIP socket (S0) supports SRAMs from 32K to 512K. You can use all three byte-wide devices as directly accessible data or program memory, BIOS extensions, or as DOS solid state disk (SSD) drives (using Ampro SSD Support Software). Byte-wide DIP socket S0 features software controlled write-protection for SRAMs and Flash EPROMs.
- **Flash Programming Control**—Flash programming voltage is hardware and software controlled. Supply +12 volts (typically, 30 mA) during Flash EPROM programming. This power is controlled by jumper W16 and a BIOS call. The BIOS call is discussed later in this chapter. Jumpering is covered in Chapter 2.
- **Configuration EEPROM**—a 2K bit serial Electrically Erasable, Programmable ROM. It stores system configuration data and can also be used to store up to 512 bits of OEM data. Battery-dependent CMOS RAM data (except date and time) is duplicated in the EEPROM, so configuration data is preserved should the battery fail. In that case, the board will function the same as if its battery were good, except for the loss of the system time and date information and any data in a CMOS RAM in byte-wide S0 being backed-up by the battery.
- **Power Fail NMI**—a power monitor circuit that generates a non-maskable interrupt and ultimately switches certain circuits to battery backup power as input voltage falls.
- **Watchdog Timer**—a timer with a SETUP selectable period. Should it time out, various results occur depending on board jumpering.

4.3 AT MOTHERBOARD LOGIC

The Little Board/486-II contains the equivalent of an entire PC/AT motherboard. This section briefly describes the motherboard subsystem in the Little Board/486-II.

A powerful 80486DX2 or -DX4 microprocessor represents the central element in the Little Board/486-II. The 80486DX2 CPU uses internal clock-doubling, to operate at 66 MHz internally and 33 MHz externally. The 80486DX4 uses internal clock-tripling, to operate at 100 MHz internally and 33 MHz externally.

The 80486DX2 and 80486DX4, high performance microprocessors, feature a 32-bit internal architecture and a 32-bit external data bus. They have versatile integrated memory management units that support virtual memory operation systems while remaining compatible with other i86 microprocessors. They include a virtual 8086 mode that permits running multiple 8086 processes with all the protection features needed to handle multiprocessing issues. In addition, the 80486DX2 and 80486DX4 use instruction pipelining, on-chip address translation, and high bus bandwidth to execute code more efficiently than previous 80x86 microprocessors. They also feature built-in math coprocessors and include an 8K byte internal four-way set-associative code and data cache.

The 80486DX4, running at 100 MHz internally, has been designed to operate at 3.3 volts instead of the usual 5 volts for standard microprocessors. The 100 MHz Little Board/486-II has a voltage regulator that reduces the 5 volt supply to 3.3 volts for the processor. All the other logic circuits operate at 5 volts.

4.3.1 ROM BIOS Device

A +12 volt Flash EPROM holds the ROM BIOS code. The BIOS is licensed by Ampro from Award, Inc., and has been modified with substantial augmentations by Ampro. (The Little Board/486-II functions described in this manual presume the use of this BIOS.) Architecturally, the ROM BIOS is located on an 8-bit internal bus (XD), from location 0F0000h to 0FFFFFh and appears also at the very top of the 16M byte address space at FF0000h to FFFFFFh. Unlike some AT implementations, the BIOS is not mirrored at 0E0000h-0EFFFFh, or FE0000h-FEFFFFh.

To improve system performance, the BIOS ROMs are shadowed. That is, the ROM contents are copied to DRAM and run from there rather than from the ROM. DRAM is 32-bits wide with very fast access times, but ROM is only 8 bits wide with multiple microprocessor wait states. Therefore, shadow operation is much faster than non-shadowed operation.

4.3.2 System Memory Map

The 80486DX2/4 CPU has 24 address lines that appear on the expansion bus, enabling it to address 16 megabytes of physical memory. (The CPU architecture allows it to address up to 64 terabytes of virtual memory.) Table 4-1 shows the allocation of the 16 megabytes of memory space.

The DRAM, the byte-wide sockets, and ROM BIOS occupy the first megabyte (starting at 00000h). You can install up to 32 megabytes of DRAM onboard with 256K, 1M, 4M, and 8M byte SIMMs. (The 8M byte SIMM is a custom product made by Ampro.) You can, in principle, use third-party memory expansion boards connected to the AT bus, but this is not usually recommended because it would severely limit system performance, as it is accessed much more slowly than onboard DRAM.

Memory Address	Function
FF0000-FFFFFFh	Duplicates BIOS at 0F0000-0FFFFFh.

100000-FDFFFFh	Extended memory
0F0000-0FFFFFh	64K ROM BIOS.
0D0000-0EFFFFh	Byte-wide memory sockets S0, S1, and S2 if enabled. Otherwise, free.
0C0000-0CFFFFh	Reserved for expansion bus ROM's; May contain DRAM for a video BIOS if shadowing is enabled.
0A0000-0BFFFFh	Normally contains video RAM, as follows: CGA Video: B8000-BFFFFh Monochrome: B0000-B7FFFh EGA and VGA video: A0000-AFFFFh
000000-09FFFFh	Onboard DRAM

Table 4-34 Little Board/486-II Memory Map

4.3.3 System

Table 4-2 is a list of the I/O port assignments used on the Little Board/486-II. Except for the SCSI controller, an Ampro-specific control port, and control ports in the ASIC chip set, the I/O port functions and addresses shown in Table 4-2 are all standard (for PC compatibles) from both a hardware and software perspective.

Typically, the ROM BIOS provides all the services needed to use the onboard devices and devices connected to I/O ports. If you need to directly program the standard functions, refer to a programming reference for the IBM PC and PC/AT.

I/O Address	Function
03F8 - 03FFh	Primary serial port
03F0 - 03F7h	Floppy disk controller ports 3F2: FDC Digital output register 3F4: FDC Main status register 3F5: FDC Data register 3F7: FDC Control register
03D0 - 03DFh	CGA display adapter (option)
03C0 - 03CFh	EGA or VGA display adapter (option)
03B0 - 03BFh	Monochrome display adapter (option)
0378 - 037Fh	Primary parallel printer port
x370 - x377h	Reserved by Ampro
0340 - 035Fh	SCSI
0300 - 031Fh	Reserved for prototype card
02F8 - 02FFh	Secondary serial port
0278 - 027Fh	Secondary parallel printer port
01F0 - 01F7h	IDE
00F0 - 00FFh	Reserved by Ampro
00C0 - 00DFh	DMA controller 2 (8237 equivalent)
00A0 - 00A1h	Interrupt controller 2 (8359 equivalent)
0092h	Fast A20 gate and CPU reset
0080 - 009Fh	DMA page registers (74LS61 equivalent)
0070 - 0071h	Real time clock and NMI mask
0060 - 0064h	Keyboard controller (8042 equivalent)
0040 - 0043h	Programmable timer (8254 equivalent)
0020 - 0021h	Interrupt controller 1 (8359 equivalent)
0000 - 000Fh	DMA controller 1 (8237 equivalent)

Table 4-35 Little Board/486-II I/O Map

Note

Other I/O ports below 100h are reserved for internal system functions and should not be accessed.

4.3.4 Powerfail Handling

If you have jumpered W23, the power management circuitry generates a power-fail NMI if the +5 volt power falls below 4.7 volts. NMI stands for "non-maskable interrupt". When this occurs, the BIOS detects the NMI and displays the message “Power Fail NMI”. At this point you have two options. You can mask the NMI and continue (the PC architecture provides a mask bit for the non-maskable interrupt), or reboot the system.

If you want to do something else with the NMI, you must provide your own power fail NMI handler and patch the NMI interrupt vector address to install it. To discriminate between the power-fail NMI and NMIs from other sources, execute a C0 command to the keyboard controller to read the P1 input port. The power-fail NMI signal lasts only 200 uS, so interrogate it first.

Here is a simple assembly language routine that you can use in a power fail NMI handler. It detects if the source of the NMI is the power fail logic.

```
;-------------------------------------------------------------
; Test for source of I/O channel check.
; Power-Fail NMI lasts only 200 uSec, interrogate it first.
;-------------------------------------------------------------
;
IN    AL,60h      ; Clear keyboard input buffer.
JMP   $+2
MOV   AL, 0C0h    ; Read kbd controller input line.
OUT   64h,AL      ;
;
Kloop:
JMP   $+2         ; Wait for data ready
IN    AL,64h      ;
TEST  AL,1        ;
JZ    Kloop       ; Kloop
IN    AL,60h      ; Get data
AND   AL,01h      ;
JNZ   NOT-PF      ;
;
;-------------------------------------------------------------
; Power fail handler code
;-------------------------------------------------------------
XXX ; XXXX
;
NOT_PF            ; Service I/O Channel check
```

4.3.5 Interrupt Controllers

A pair of 8259A compatible interrupt controllers (within the core logic chip) provide fifteen prioritized interrupt levels. The onboard interfaces and controllers use some of these. You can use the others for peripherals or devices on the expansion bus. Table 4-3 lists the typical interrupt usage.

Interrupt	Function
IRQ0	ROM BIOS clock tick function, from Timer 0
IRQ1	Keyboard controller output buffer full
IRQ2	Cascade input for IRQ8-15
IRQ3	Secondary serial port (if present)
IRQ4	Primary serial port (if present)
IRQ5	Parallel port option
IRQ6	Floppy controller (if present)
IRQ7	Parallel port option
IRQ8	Battery-backed clock alarm (watchdog timer)
IRQ9*	EGA or VGA controller (if present)
IRQ10	Available
IRQ11	Available
IRQ12	Available
IRQ13	Internal use, not available
IRQ14	IDE controller (if present)
IRQ15	Available
* Corresponds to IRQ2 on a PC's expansion bus.	

Table 4-36 Interrupt Channel Assignment

4.3.6 DMA Controllers

The core control logic includes two 8237A compatible DMA controllers providing seven DMA channels. The hardware and software implementation of these controllers, and the additional address generation logic included, are functionally identical to a standard AT. Table 4-4 shows how the Little Board/486-II uses the seven DMA channels.

Channel	Function
0	Available for 8-bit transfers
1	Available for 8-bit transfers
2	Floppy controller (if present)
3	Available for 8-bit transfers
4	Cascade for channels 0-3
5	Available for 16-bit transfers
6	Available for 16- bit transfers
7	Available for 16-bit transfers

Table 4-37 DMA Channel Assignment

As in all AT-compatible computers, several distinctions exist between the features of DMA Channels 0-3 and Channels 5-7:

- **Channels 0-3** —used for 8-bit transfers between 8-bit I/O adapters and both 8-bit and 16-bit system memory. For these channels, the 8237A compatible DMA controller generates addresses A0-A15. DMA page registers generate addresses A16-A23. These channels can transfer data throughout the AT's 16 megabyte address space, with a maximum block transfer size of 64K bytes.

- **Channels 5-7** —These three channels support 16-bit transfers between 16-bit I/O adapters and 16-bit system memory. For these channels, the 8237A compatible DMA controller logic generates A1-A16. Page registers generate A17-A23. These channels can transfer data throughout the AT's 16 megabyte address space, with a maximum block transfer size of 128K bytes. Channels 5-7 do not support address bit A0, so they cannot transfer data on odd byte boundaries.

Note

During a DMA block transfer, addresses cannot automatically cross page boundaries. On Channels 0 to 3, the page boundaries are at 64K. Channels 5 to 7 have 128K boundaries. The ROM BIOS disk services handle boundary crossing automatically.

4.3.7 Programmable Timers

The core control logic includes an 8254 compatible timer/counter in a circuit compatible with the PC/AT architecture. A 1.193 MHz clock, derived from a 14.318 MHz oscillator, drives each channel of the 8254. This oscillator can be internally divided down to provide a variety of frequencies.

Table 4-5 summarizes the standard use of this device's timers. You can also use timer 2 as a general purpose timer if you don't need the speaker function.

Timer	Function
0	ROM BIOS clock tick (18.2 Hz)
1	DRAM refresh request timing (15 uSec)
2	Speaker tone generation time base

Table 4-38 Timer Assignment

4.3.8 Keyboard Interface

The board uses a standard AT keyboard controller. A four-wire cable, carrying +5 volts, ground, data and clock, connects the keyboard to the board at J5.

The system will operate without a keyboard. To prevent the POST (power-on-self-test) from stopping and issuing a keyboard error message, specify No Keyboard Error Halt on the first page of SETUP.

Besides implementing the keyboard interface, the keyboard controller chip provides a digital I/O port used for other system functions:

- It operates the A20 Gate and 80486DX2/4 CPU reset associated with system protected mode control.
- It provides several control signals for onboard logic control, including keyboard disable switch sensing and power fail NMI status.

4.3.9 Speaker Interface

One of the core control logic devices includes a standard AT-compatible speaker port. The speaker logic signal is buffered by a transistor amplifier, and provides about 100 milliwatts to an external 8 ohm speaker.

The audio output is based on two signals: the output of Timer 2; and the programming of two bits, 0 and 1, at I/O port 61h. Bit 1 of I/O port 61h is one term of a 2-input AND gate in U11. The other term is the output from Timer 2. Thus, setting bit 1 to a logic 1 enables the output of Timer 2 to the speaker, and a logic 0 disables it. If you disable Timer 2 by setting bit 0 of port 61h to a 0, then you can use bit 1 of port 61h to control the speaker directly.

4.3.10 Battery-Backed Clock

The date/time clock includes a CMOS static RAM for Configuration Memory. Unique to the Little Board/486-II is automatic back-up of this data in the Configuration Memory EEPROM by the ROM BIOS SETUP function.

The board has a 1/2 AA lithium battery located near the power connector. You can remove the battery and operate the board without it if you choose. As the CMOS RAM's contents are backed up in the configuration EEPROM, configuration data is not lost when the battery is removed. Should battery power be lost, the Ampro ROM BIOS detects that the CMOS RAM's data is invalid and loads the correct data from the configuration EEPROM. With battery failure, only time and date information are lost.

4.3.11 Expansion Bus

The PC bus signals on P1 include an 8-bit bi-directional data bus, 20 address lines, 6 levels of interrupt, three DMA channel handshake lines, several other control lines, and power and ground for expansion cards. The first 62 pins of P1 (A1-31, B1-31) correspond to the standard signals on the 62-pin PC bus edgecard backplane. The last two pins of P1 are extra grounds to enhance system reliability. An independent 14.318 MHz oscillator provides the bus OSC signal. The OSC signal is *not* synchronous to other expansion bus signals.

The AT bus signals on P2 include six high order address lines, the high order data byte, five additional interrupt requests (IRQs), four additional DMA channel handshake signal pairs (DRQ/DACK), and several additional control signals. You can latch the upper six address lines (LA17-LA23) with BALE.

For compatibility with AT standards, wait states are inserted in all expansion bus I/O and memory cycles. I/O cycles include 4 wait states. Memory cycles include 1 wait state. The bus operates at 8 MHz. With certain minor exceptions (for example, bus loading and driver current capacity) it meets the IEEE P996 (draft) specifications for the ISA bus. It is compatible with the PC/104 bus specification.

4.4 FLOPPY DISK CONTROLLER

The board implements a floppy disk subsystem. It is equivalent to a AT floppy disk controller card. The floppy controller consists of:

- Formatter/controller
- Digital data separation
- Digital write precompensation
- Data rate selection
- Clock generation
- Drive interface drivers and receivers
- Drive motor control
- AT compatible control registers

The floppy controller uses circuitry that is compatible with the industry standard NEC765 floppy controller, and associated external devices, as in a standard AT. It uses data rates of 125, 250, 300, and 500K bits/Sec.

The system supports dual capacity drives, which allow a single drive to read both 360K byte and 1.2 megabyte diskettes. These are standard AT drives in which the data rate shifts between 250K and 500K BPS, but the drive rotational speed stays constant.

The board's 37C65 floppy controller contains a highly reliable 2nd order digital data separator. The device also generates write precompensation (187 nS). Because both critical circuits are fully digital, the floppy drive interface is highly reliable and temperature independent.

The ROM BIOS configures the floppy interface based on the Configuration Memory in the battery-backed CMOS RAM in the clock chip, and backed up in the EEPROM Configuration Memory. You can use any combination of supported drives.

The floppy controller uses interrupt channel IRQ6, and DMA channel 2 (DRQ2 and DACK2). If you do not require floppy disk drives in your application, these and the floppy controller's I/O port addresses are available. To disable the floppy port, disable Floppy Controller in SETUP and remove jumpers W20-1/2, 3/4, and W27-1/2. Refer to Chapter 2 for jumpering information and for the port's connector pinout.

4.5 SERIAL CONTROLLER

The two AT-standard 16C450-compatible RS232C serial ports are full duplex asynchronous communications channels. Many of their features are software selectable. These include: baud rate (up to 57.6K baud); word size (5-8 bits); and stop bits (1, 1.5, or 2). The RS-232 specification limits the drivers usable baud rate to 19.2K baud, as the maximum slew rate is 30 volts/uSec.

Each serial port has three output and 5 input signals with EIA compatible buffers. Onboard DC-to-DC converters generate the ±9 volts required by the RS232C drivers.

You can disable either serial port with SETUP. When you disable a port, its I/O port addresses and system interrupt become available for other devices. Refer to Chapter 2 for the connector pinouts of these two ports.

4.6 PARALLEL CONTROLLER

The parallel printer port is functionally identical to a standard AT port with one exception: in a conventional AT, when you read the eight data lines, you read the data you just wrote to them. On the Little Board/486-II, you can sense the state of the lines as they are influenced by an external device.

Thus, you can use the parallel port as a bi-directional data port with up to 12 output lines and 17 input lines. This can be very valuable in custom applications. For example, you might use it to control an LCD display, scan keyboards, sense switches, or interface with optically isolated I/O modules. All data and interface control signals are TTL compatible.

In SETUP, you can configure the parallel port as either the primary parallel port, or the secondary parallel port. These differ in the I/O port base address (378h or 278h). You can also disable the port. If you disable it, its I/O port addresses and system interrupt become available for other devices. Any changes you make in SETUP take effect the next time you boot the system.

The default mode of the port is AT-compatible. That is, the port is configured as "output only". To use the port as a bi-directional data port you must put it in Extended Mode with a BIOS call. Here is an example of code to do that.

```
;-----------------------------------------------------------
; Code to set the parallel port mode
;-----------------------------------------------------------
MOV   AH,0CDh     ; AMPRO command
MOV   AL,0Ch      ; AMPRO function
MOV   BX,01h      ; Extended mode (00 for output-only mode)
INT   13h
```

This code leaves the port in input mode. Once the port is in Extended Mode, you can directly access the control register without using the BIOS. The port address is 37Ah for LPT1 and 27Ah for LPT2. You can dynamically change the port between input and output modes by changing bit 5. A 1 in bit five sets the port to input only; a 0 sets it to output only. Here is a sample of code for dynamically changing the port direction after it is in Extended Mode.

```
;-------------------------------------------------------------
; Code to change the parallel port direction to output
;-------------------------------------------------------------
MOV   DX,37Ah          ;(27Ah for LPT2)
IN    AL,DX
OR    AL,0010 0000b    ;set bit 5
OUT   DX,AL
;
;-------------------------------------------------------------
; Code to change the parallel port direction to input
;-------------------------------------------------------------
MOV   DX,37Ah          ;(27Ah for LPT2)
IN    AL,DX
AND   AL,1101 111b     ;clear bit 5
OUT   DX,AL
```

Besides the eight data lines, you can use the four control lines (-STROBE, -AUTOFD, -INIT, and -SEL IN) as general purpose output lines. Similarly, you can use the five status lines (-ERROR, SEL OUT, PAPER EMPTY, -ACK, and BUSY) as general purpose input lines.

You can read the four control lines and use them as input lines. These lines have open collector drivers with 4.7K ohm pullups. To use a control line as an input line, you must first write to its corresponding bit in the control register. Refer to the Active High/Low column in Table 4-7. If the line active high, write a 1. If the line is active low, write a 0. This will cause the line to float (pulled up by the 4.7K ohm resistors). When they float, you can use them as inputs. Table 4-6 is a summary of the uses of the parallel port lines.

Signal Type	Number of Lines	Function
Data	8 lines	Read/Write
Control	4 lines	Read/Write
Status	5 lines	Read Only

Table 4-39 Parallel Port Use

Bit 4 in the control register (Table 4-7) enables the parallel port interrupt. If this bit is high 1, then a rising edge on the -ACK (IRQ) line will produce an interrupt on the interrupt selected by W24, either IRQ5 or IRQ7.

Register	Bit	Signal Name or Function	In/Out	Active High/Low	J3 Pin	DB25F Pin
DATA	0	Data 0	I/O	High	3	2
(378h)	1	Data 1	I/O	High	5	3
(278h)	2	Data 2	I/O	High	7	4
	3	Data 3	I/O	High	9	5
	4	Data 4	I/O	High	11	6
	5	Data 5	I/O	High	13	7
	6	Data 6	I/O	High	15	8
	7	Data 7	I/O	High	17	9
STATUS	0	1	---	---	---	---
(379h)	1	1	---	---	---	---
(279h)	2	1	---	---	---	---
	3	-ERROR	In	Low	4	15
	4	SEL OUT	In	High	25	13
	5	PAPER OUT	In	High	23	12
	6	-ACK (IRQ)	In	High	19	10
	7	BUSY	In	Low	21	11
CONTROL	0	-STROBE	Out*	Low	1	1
(37Ah)	1	-AUTOFD	Out*	Low	2	14
(27Ah)	2	-INIT	Out*	High	6	16
	3	-SEL IN	Out*	Low	8	17
	4	IRQ ENABLE	---	High	---	---
	5	-OUT ENABLE	---	Low	---	---
	6	1	---	---	---	---
	7	1	---	---	---	---
* Can also be used as input (see text).						

Table 4-40 Parallel Port Register Bits

4.7 INTEGRATED DEVICE ELECTRONICS (IDE) INTERFACE

The module's IDE interface is for hard disk drives that contain an embedded hard disk controller. The advantage of using IDE drives is that you need no other hard disk controller because the controller is in the drive itself.

The IDE interface provides bus signal buffering. You can expect reliable operation with ribbon cables up to 18 inches in length, or perhaps longer, depending on system ambient noise level. You can attach up to two IDE drives to the interface. Each drive must be configured according to the manufacturer's instructions as Drive 1 or Drive 2. This may vary from one drive manufacturer to another.

You can use both IDE drives and SCSI devices in the system at the same time. Configure drive usage using the SETUP facility. In addition, using SETUP you can select a floppy drive, an IDE drive, or a SCSI drive as the system default boot device. IDE drive configuration is covered in Chapter 3.

4.8 SMALL COMPUTER SYSTEM INTERFACE (SCSI)

The SCSI interface is implemented using an Adaptec AIC-6360 SCSI-II controller chip. The device is ANSI X3.131 compatible with 48 mA bus drivers. It provides hardware support for bus arbitration, REQ/ACK handshaking, asynchronous SCSI data transfer, and all required SCSI Initiator and Target functions. The SCSI port interfaces to the CPU on an internal 16-bit bus. The SCSI bus to external devices is 8 bits wide.

If you need to access the SCSI chip directly, refer to the AIC-6360 technical manual for additional technical details:

AIC-6360 Technical Manual
Adaptec Corporation
691 South Milpitas blvd.
Milpitas, CA 95035

For more information on the SCSI interface, refer to the American National Standards Institute document, ANSI X3.131-1986, available from:

American National Standards Institute
1430 Broadway
New York, NY 10018

The SCSI interface on the Little Board/486-II uses active terminators for the SCSI bus. This meets the SCSI bus specification while substantially reducing power requirements. Only two terminators should be enabled on the SCSI bus at one time. A jumper (W28) is provided for enabling or disabling the active terminators. See Chapter 2 for details.

The SCSI specification requires that the SCSI interface be able to provide termination power for external devices. You can short W6 to provide power to the SCSI TERMPWR signal. External termination networks may require this external supply. As recommended by the SCSI specification, a Shottky diode on the board limits current flow from the SCSI TERMPWR line to the board. This prevents system damage from occurring in the event other devices on the SCSI bus are also supplying power to the TERMPWR line.

You can eliminate the SCSI function by disabling it in SETUP. Removing the jumper from W1 frees I/O port addresses 340-35Fh.

4.8.1 SCSI Hard Disk Support

You can use SCSI drives instead of a conventional hard disk controller. Depending on the hard disk configuration you select in SETUP, the ROM BIOS automatically maps hard disk functions to the SCSI interface. It does this using the SCSI direct access device Common Command Set for compatibility with a variety of SCSI disk drives. Due to the use of the SCSI Common Command Set, you can use a variety of commercially available hard disks. Other direct access device options include: optical storage devices, certain types of tape drives, and SCSI solid state disk drives.

The ROM BIOS reads the SCSI Initiator ID from the configuration EEPROM. This ID is used in all SCSI transactions. You set the Initiator ID in SETUP (see Chapter 3). Typically, use the default Initiator ID of 7; The ROM BIOS performs a SCSI bus reset on powerup or system reset when you use this value.

4.8.2 Ampro SCSI BIOS

To make it easy to use SCSI devices and functions not supported by the standard Ampro drivers and utilities, the ROM BIOS includes the SCSI BIOS functions that allow use of the SCSI interface without direct hardware programming.

The SCSI BIOS adds an additional layer of standardization to SCSI. It provides a high level interface for software. The Ampro SCSI BIOS is accessed through an extension of the INT13 ROM BIOS support. SCSI drivers and utilities written around the Ampro SCSI BIOS INT13 extensions will run

on any system in which the INT13 extensions are available. (This includes all Ampro's DOS compatible Little Board single board systems).

If you plan to support SCSI devices using the SCSI BIOS, you should acquaint yourself with the SCSI bus and its protocols. You can obtain a copy of the ANSI SCSI specification, from the American National Standards Institute. Additional detailed information on the SCSI BIOS is available in application note AAN-8804, available from Ampro.

4.9 BYTE-WIDE MEMORY SOCKETS

This section describes the three byte-wide device sockets on the Little Board/486-II

4.9.1 General Description of the Byte-Wide Sockets

The three byte-wide memory sockets at S0, S1, and S2 support a variety of 28- and 32-pin JEDEC pinout memory devices, including EPROM, Flash EPROM, and page-addressed EPROM. One socket, S0, also accepts an SRAM or NOVRAM device. Chapter 2 gives examples of the memory devices each socket will hold. Ampro's solid state disk (SSD) drive support in the ROM BIOS and optional SSD Support Software treat these sockets as up to three DOS disk devices, containing up to 1M byte of EPROM each. The PLCC sockets (S1 and S2) are wired to accept JEDEC EPROMs up to 1M byte although 512K byte devices are the largest available (circa 1995). The DIP socket (S0) will hold currently available 1M byte EPROMs.

The BIOS accesses the byte-wide sockets as 8-bit devices, but only one byte-wide socket is enabled at a time. Using SETUP, select the starting address and size of each byte-wide socket used in your application. If you install smaller devices than specified in SETUP, the devices are "mirrored" (that is, appear at multiple addresses) in the socket's address window.

The byte-wide sockets can be disabled by the ROM BIOS at powerup and boot time via options in the Configuration Memory. Additionally, the sockets can be enabled or disabled under software control, using a ROM BIOS call provided for this purpose. See the code example below. When you disable a byte-wide socket (with SETUP or the BIOS call), its address space is available on the expansion bus. When you enable a socket, its address space is not available on the bus, even if there is no memory device installed in the socket. When you select one of the byte-wide sockets, the logic automatically deselects the other two. Devices used in the byte-wide sockets must have access times of 250 nS or less.

4.9.2 Accessing Large Devices

For devices over 64K bytes, select the 64K byte window size in SETUP. You must then select which segment of the device you want to appear in that window, using code equivalent to that illustrated below. Table 4-8 gives the byte (in Hex) to write to the BH register to select each segment of a large device. Here is a simple assembly language routine that controls the byte-wide memory sockets. The segment byte is given in binary.

```
;-------------------------------------------------------
; Page select code for byte-wide sockets
;-------------------------------------------------------
MOV   AH,0CDH          ; AMPRO function call
MOV   AL,nn            ; Use 03 for S0, 04 for S1, or 0A for S2
MOV   BL,nn            ; Use 01 to turn ON or 00 to turn OFF
MOV   BH,xxxx0000b     ; The upper nibble of BH contains the page
                       ; number for devices larger than 64 K.
INT   13H
```

Device Size	64KB Segments	Segment Address (Upper Nibble of BH)
128K	2	LOWER BH=00h UPPER BH=10h
256K	4	FIRST BH=00h SECOND BH=10h THIRD BH=20h FOURTH BH=30h
512K	8	FIRST BH=00h SECOND BH=10h THIRD BH=20h FOURTH BH=30h FIFTH BH=40h SIXTH BH=50h SEVENTH BH=60h EIGHTH BH=70h
1M	16	FIRST BH=00h SECOND BH=10h THIRD BH=20h FOURTH BH=30h FIFTH BH=40h SIXTH BH=50h SEVENTH BH=60h EIGHTH BH=70h NINTH BH=80h TENTH BH=90h ELEVENTH BH=A0h TWELFTH BH=B0h THIRTEENTH BH=C0h FOURTEENTH BH=D0h FIFTEENTH BH=E0h SIXTEENTH BH=F0h

Table 4-41 Segment Addressing in Large Memory Devices

4.9.3 Byte-Wide Socket Signals

Jumper arrays W2, W8 and W9 (for S0), W13 (for S1) and W15 (for S2) configure the byte-wide sockets. The following tables list the signals that appear on the pins of the arrays.

W2 Pin	Signal	Signal Function
1	CEO	Battery backup chip select (for SRAMs)
2	-CS	Socket S0, pin 22, chip select
3	-BWCS0	Chip select for EPROM and Flash devices

Table 4-42 Byte-Wide S0 Jumper Pin Signals, W2

W8 Pin	Signal	Signal Function
1	BA19	Buffered Address 19; BH register, bit 7
2	S0-1	Socket S0, pin 1; W9-1

Table 4-43 Byte-Wide S0 Jumper Pin Signals, W8

W9 Pin	Signal	Signal Function
1	SA15	System Address 15
2	S0-31	Socket S0, pin 31
3	BA18	Buffered Address 18; BH register, bit 6
4	S0-3	Socket S0, pin 3
5	-PMEMW	Memory Write; Write protect function controlled
6	S0-1	Socket S0, pin 1
7	SA14	System Address 14
8	S0-29	Socket S0, pin 29
9	Vpp	Flash EPROM programming voltage (through W16)
10	Vcc*	Power
11	S0-30	Socket S0, pin 30
12	BA17	Buffered Address 17; BH register, bit 5
*W25-1/2 = Vcc W25-2/3 = Backup battery power. Do not use for EPROMs		

Table 4-44 Byte-Wide S0 Jumper Pin Signals, W9

W11 Pin	Signal	Signal Function
1	BWCS0	S0 chip select (to W2)
2	BW0	Byte-wide 0 chip select source
3	BIOS	BIOS chip select source
4	U19CS	U19 chip select to BIOS Flash Memory

Table 4-45 Byte-Wide S0 Jumper Pin Signals, W11

W14 Pin	Signal	Signal Function
1	N/C	No connection
2	BA19	Buffered Address 19, BH register, bit 7
3	S1-1	Socket S1, pin 1

4	Vpp	Flash EPROM programming voltage (through W16)
5	-PMEMW	Memory Write; Write protect function controlled
6	S1-31	Socket S1, pin 31
7	BA18	Buffered Address 18; BH register, bit 6
8	NC	No connection

Table 4-46 Byte-Wide S1 Jumper Pin Signals, W14

W15 Pin	Signal	Signal Function
1	N/C	No connection
2	BA19	Buffered Address 19, BH register, bit 7
3	S2-1	Socket S2, pin 1
4	Vpp	Flash EPROM programming voltage (through W16)
5	-PMEMW	Memory Write; Write protect function controlled
6	S2-31	Socket S2, pin 31
7	BA18	Buffered Address 18; BH register, bit 6
8	NC	No connection

Table 4-47 Byte-Wide S2 Jumper Pin Signals, W15

W16 Pin	Signal	Signal Function
1	S1, S2	Software controlled Vpp
2	+12V	Flash EPROM programming power

Table 4-48 Jumper Pin Signals, W16

W25 Pin	Signal	Signal Function
1	Vcc	+5 volt supply, Vcc
2	S0	Socket S0, power
3	Vcco	Backup-battery power to S0

Table 4-49 Jumper Pin Signals, W25

W26 Pin	Signal	Signal Function
1	U19	Flash BIOS programming input (factory use only)
2	Vpp	Programming voltage input

Table 4-50 Jumper Pin Signals, W26

4.9.4 Write Protection

If you use an SRAM or Flash EPROM in byte-wide socket S0, you can write protect it to prevent unwanted data from inadvertently being written to it. You can do this dynamically with the following code.

```
;-------------------------------------------------------------
; Code to write protect/un-write protect socket S0
;-------------------------------------------------------------
MOV   AH,0CDh     ; AMPRO function call
MOV   AL,0Eh      ; Subfunction
MOV   BL,0        ; Use 0 to disable write protect (enable writing)
MOV   BL,1        ; Use 1 to enable write protect (disable writing)
INT   13H
```

4.9.5 Flash EPROM Programming

To program Flash EPROMs you need to supply +12 volts to the board, and jumper W7-1/2 and W16-1/2 to supply the power to the software controlled Vpp (programming voltage) switch, and turn the switch on at the appropriate time. The following code turns the Vpp switch on and off.

```
;-------------------------------------------------------------
; Code to turn Vpp power on or off
;-------------------------------------------------------------
MOV   AH,0CDh     ; AMPRO function call
MOV   AL,09h      ; Subfunction
MOV   BL,0        ; Use 0 to turn off +12 Flash EPROM
                  ; programming voltage
MOV   BL,1        ; Use 1 to turn on +12 Flash EPROM
                  ; programming voltage
INT   13H
```

Jumpers W7 and W16 must be installed to program Flash EPROMs

4.10 CONFIGURATION EEPROM

System configuration is simpler and more flexible because of the Configuration Memory. The Configuration Memory is a 2K-bit Electrically Erasable PROM (EEPROM). The Configuration Memory serves two purposes: part of it is for SETUP; the rest is for OEM data storage. The ROM BIOS configures the system from the data stored in the SETUP portion of the memory following powerup and reset.

You can store up to 512 bits of data in the OEM portion of the Configuration Memory. To use the OEM memory directly from programs, the Ampro ROM BIOS includes functions for reading and writing the configuration EEPROM. For details, refer to application note AAN-8805, available from Ampro.

You can also use SETUP.COM to read from and write to the configuration EEPROM. To do this, first, save the current contents of the EEPROM to a file. Then with a text editor, change any values you want in the lines designated EEPROM40 through EEPROM70. DO NOT alter any other values. They are codes for the ROM BIOS setup parameters. After you have entered the data you wish, save the file and use SETUP.COM to load the altered file back to the configuration EEPROM.

Do not use the OEM section of the configuration EEPROM as DRAM. It is limited to 10,000 write cycles. Use the ROM BIOS routine for configuration EEPROM access. This routine automatically

maintains a count of the number of times the EEPROM is written. It also maintains a checksum of the contents to insure data integrity.

4.11 SERIAL LOADER OPTION

The Ampro SERLOAD utility is included on the Common Utilities diskette furnished with the Little Board/486-II Development Kit. You can use it to serially download and execute a block of executable code prior to system boot. Data is transferred to the board on a three-wire RS232C cable to COM1 as the target. The host is a remote system sending the data. To use this function, you must enable it on the target board (the Little Board/486-II) using SETUP.

The host may be any PC or AT compatible running DOS that has a standard RS232C serial port. The maximum file size is 64K bytes. The file must be a binary, executable file with its origin at 0000h. Instructions for the use of SERLOAD are provided in the Ampro Common Utilities manual.

4.12 WATCHDOG TIMER

The purpose of a watchdog timer function is to restart the system should some mishap occur. Possible problems include: a failure to boot properly; the application software losing control; temporary power supply problems including spikes, surges, or interference; the failure of an interface device; unexpected conditions on the bus; or other hardware or software malfunctions. The watchdog timer helps assure proper start-up after an interruption.
The Little Board/486-II ROM-BIOS supports the board's watchdog timer function in two ways:

- There is an initial watchdog timer setting, specified using SETUP, which determines whether the watchdog timer will be used to monitor the system boot, and if so, how long the time-out is (30, 60, or 90 seconds).
- There is a special ROM-BIOS function which may be used by application software to start, stop, and retrigger the watchdog timer function.

The initial time-out should be set (using SETUP) to be long enough to guarantee that the system can boot and pass control to the application. Then, the application must periodically retrigger the timer so that the time-out does not occur. If the time-out does occur, the system will respond in a manner determined by how the watchdog timer jumper is set (see Chapter 2).

The following simple assembly language routine illustrates how to control the watchdog timer using the Ampro ROM-BIOS function that has been provided for this purpose:

```
;-----------------------------------------------------------
; Watchdog timer control program
;-----------------------------------------------------------
MOV   AH,0C3h           ; Watchdog Timer BIOS function
MOV   AL,nn             ; Use "00" to disable, "01" to enable
                        ; timer.
MOV   BX,mm             ; Selects time, in seconds
                        ;(00-FFh; 1-255 seconds)
INT   15h
```

The WATCHDOG program is also described in the Ampro Common Utilities manual.

INDEX