4.1 The Intel i486 Microprocessor (CPU)

The Intel 80486 microprocessor (also referred to as 486) uses 32-bit architecture. It contains on-chip memory management, on-chip floating point, and cache memory units.

Some features of the 486 are:

* 100% binary compatible with the i386 microprocessor; contains several features and enhancements to improve performance.

  - On-chip cache memory allows frequently used data and code to be stored on-chip, reducing external bus access requirements.

  - The 387 math coprocessor has been brought on-chip.

  - RISC design techniques have been used so that frequently used instructions execute in one cycle.

The 486 provides more than twice the performance of the 386 microprocessor.

The processor can address up to 4-Gigabytes of physical memory and 64-Terabytes (1-Terabyte = 1-K Gigabyte) of virtual memory. To implement a virtual memory system, the 486 supports full restartability for all page and segment faults. It contains a memory management unit (MMU) and protection architecture which includes address translation registers, advanced multitasking hardware and protection mechanism to support

operating systems. In addition, it is object code compatible with the 8086 family of microprocessors. The 486 has built-in features to support coprocessors, DMA and interrupts (both maskable and non-maskable). It has two modes of operation: Real address mode and Protected virtual address mode.

In the Real Mode, the 486 operates using the same architecture as the 8086. It allows access to its 32-bit register set. The addressing, memory size, and interrupt handling are identical to the Real Mode of the i286. All 486 instructions are available in Real Mode except some privileged instructions used in protected mode only.

In Protected mode, all capabilities of the 486 are unlocked. This mode is called the Protected Virtual Address Mode. All existing 8086, 80286, and 80386 software can be run.
The differences between Real Mode and Protected Mode are, for the programmer, an increased address space and a different addressing mechanism.

4.2 Processor Clock Speed

The AMI 486 Enterprise can operate at two different speeds, a "hi" speed and a "lo" speed. The "lo" speed is programmable through ROM BIOS set up. The "hi" speed is factory set to one of two speeds: 25 MHz or 33 MHZ (The 33 MHz board will be available when the 33 MHz 486 becomes available). At the time of power on, the board boots up at "hi" clock speed. The clock speed can be switched at any time during operation through a keyboard command (see section 2.9). This is under

most of the operating system environments that use system BIOS control, e.g. PC DOS, MS DOS, etc. An exception is the UNIX operating environment, which does not require onboard BIOS after booting. Under UNIX or similar operating systems, the clock speed cannot be switched through a keyboard command at any time. The clock speed can also be changed through a turbo switch connected to **J27**. This mode of clock switching is independent of the operating system environment. For UNIX operating system environment, the system operating speed can be changed through the turbo switch. The system will always boot up in "hi" speed and will remain there until manually switched to "lo" speed using one of the methods previously mentioned. When you switch to "hi" speed, the turbo LED is turned on. Similarly, when you switch to a "lo" speed the turbo LED is turned off. The keyboard switch and the hardware switch are independent of each other. In other words, either the keyboard or the turbo switch can be used to switch to "lo" speed and either the keyboard or the turbo switch can be used to switch back to "hi" speed.

4.3 Dynamic Device Sizing Capability

The 486 microprocessor supports two types of accesses: Memory, and Input/Output. Each type of access can be 32-, 16- or 8-bit wide. The AMI 486 Enterprise allows any type of access to a device of any width. If necessary, the hardware will split a 486 bus cycle into a number of (up to 32/8=4) cycles to allow access to a 16- or 8-bit device. On-board main memory and cache memory is organized as 32-bit wide. The WEITEK

Coprocessor is also 32-bit wide. All the on-board I/O devices are 8-bit wide. The EISA I/O slots of the AMI 486 Enterprise are capable of supporting 32, 16 or 8 bit memory or I/O devices.

4.4 Turbocache486 Cache Module

With the 486, a zero wait state burst read cycle requires five clock cycles to fetch four 32-bit words. The required number of clock cycles increases by one with every wait state introduced in the access. To speed up the memory read accesses, AMI incorporates an optional 64KB or 128KB of 32 bit wide directly mapped cache module. The Cache is implemented with a write through algorithm, storing data relating to that memory region which is most frequently accessed by the 486. In this scheme, with the cache memory subsystem enabled, XX% of data will be available in the cache for memory read operations (this is a statistical value for normal program execution). This improves system speed, since the cache access requires no wait states, whereas the main memory read access would require a few wait states.

The cache memory subsystem caches the following memory address region: 00000000H-0009FFFFH, 00100000H-01FFFFFFH for memory on the system board and on the memory board on the AMI 32-bit slot. It does not cache memory devices on EISA slots. This avoids compatibility problems with some of the EISA/AT compatible memory cards which have paging, dual porting and like facilities.

If the cache module is disabled, it will be flushed, causing all cache locations to be invalidated. If the

cache is disabled, the data will be read from system memory for any memory read access. If the cache is enabled, data will come from cache memory for a memory read hit access resulting in 0-wait state cycle.

The cache memory subsystem can be enabled by a keyboard key combination <CTRL><ALT><SHIFT><+> and disabled by keyboard key combination <CTRL><ALT><SHIFT><->.

4.5 On Board Memory Subsystem

There are two 32-bit banks of dynamic memory (called RAM, Random Access Memory). Both memory banks are capable of supporting 256K, 1M, or 4M type dynamic memory chips facilitating options for having a total of 1MB, 2MB, 4MB, 8MB 16MB, or 32MB of memory on the motherboard. Both memory banks support Single In-line Memory Modules, which will be referred to as SIMM. The memory subsystem supports 486 burst read mode and EISA burst read/write mode. The memory subsystem supports paging with a page size of 2K if one bank is present or 4K if both banks are present. Both the banks have to use the same type of memory. The memory address ranges for different combinations are shown in TABLE 4.5.1.

64KB block of memory is used for RAM Shadowing System BIOS ROM to improve System BIOS execution speed. Another 32KB block of memory is used for RAM Shadowing Video BIOS. RAM Shadowing of video BIOS is optional.

TABLE 4.5.1

BANK1 type	BANK2 type	Memory Address Range (hexadecimal)
256K	Absent	00000000 - 0009FFFF
256K	256K	00000000 - 0009FFFF
		00100000 - 001FFFFF
1M	Absent	00000000 - 0009FFFF
		00100000 - 003FFFFF
1M	1M	00000000 - 0009FFFF
		00100000 - 007FFFFF
4M	Absent	00000000 - 0009FFFF
		00100000 - 00FFFFFF
4M	4M	00000000 - 0009FFFF
		00100000 - 01FFFFFF

4.6 AMI 32-BIT Memory Card on 32-BIT Slot

The AMI 32-bit proprietary slot has been specially designed for fast access to 32-bit memory card. It works at the same speed as the on-board memory. The AMI 32-bit memory card is configurable so that it can be placed at memory address space above the on-board memory address region. Installation details for the AMI 32-bit memory card can be found in Section 8.

The AMI 486 Enterprise 32-bit memory adapter card supports four 32-bit banks. All the banks use SIMM modules and all of them can take either 256K, 1M, or 4M type DRAM. The SIMM modules need to be filled up in steps of 2 banks with same kind of memory because of page interleave support. With all the banks filled up, it increases the system memory by 64 megabytes for a total of 96 megabytes with 32 megabytes of on board memory.

Table 4.6.1 shows the memory address region the AMI 32-bit memory card can occupy for different configurations of on-board memory.

TABLE 4.6.1

BANK: (3-6 on card)						Memory Card	Total
1	2	3	4	5	6	Hex Address Range	RAM
1M	1M	1M	----	----	----	00800000 - 00BFFFFF	12M
1M	1M	1M	1M	----	----	00800000 - 00FFFFFF	16M
1M	1M	1M	1M	1M	----	00800000 - 013FFFFF	20M
1M	1M	1M	1M	1M	1M	00800000 - 017FFFFF	24M

1M1M 4M---------- 00800000 - 017FFFFF 24M
1M1M 4M4M-------- 00800000 - 027FFFFF 40M
1M1M 4M4M4M------ 00800000 - 037FFFFF 56M
1M1M 4M4M4M4M---- 00800000 - 047FFFFF 72M
4M4M 1M---------- 02000000 - 023FFFFF 36M
4M4M 1M1M-------- 02000000 - 027FFFFF 40M
4M4M 1M1M1M------ 02000000 - 02BFFFFF 44M
4M4M 1M1M1M1M---- 02000000 - 02FFFFFF 48M
4M4M 4M---------- 02000000 - 02FFFFFF 48M
4M4M 4M4M-------- 02000000 - 03FFFFFF 64M
4M4M 4M4M4M------ 02000000 - 04FFFFFF 80M
4M4M 4M4M4M4M---- 02000000 - 05FFFFFF 96M

4.7 Read Only Memory (ROM) System BIOS Subsystem

The System BIOS ROM subsystem consists of 1 Read Only Memory (ROM) chip occupying memory address range of 000E0000H-000FFFFFH and FFFE0000H-FFFFFFFFH. The System BIOS ROM is 16-bit wide and it can be of type 27512 for a total of 64K ROM or 27010 for a total of 128K ROM. The ROM area 000E0000H-000EFFFFH is used for system board manufacturing diagnostics and the ROM area 000F0000H-000FFFFFH is used for system ROM BIOS. The system BIOS maps the 000E0000H-000EFFFFH area to the EISA bus if the manufacturing diagnostic mode is not used. The ROM area 000F0000H-000FFFFFH is shadowed into the main memory to improve system performance.

4.8 EGA/VGA RAM SHADOW

The memory space from address 0C0000H thru 0C7FFFH is reserved for Video Display ROM. Most often, only the EGA BIOS (0C0000H-0C3FFFH), which is normally accessed through a 8-bit bus, is located in this area. The slow execution of this device driver makes the video I/O sluggish. The AMI 486 Enterprise provides the option of mapping this space into the 32 bit system board RAM. This improves the video I/O speed, since the EGA BIOS can now be accessed through a 32-bit bus. When this is done, the memory address 0C0000H through 0C7FFFH cannot be accessed on the 8/16-bit I/O slot.

4.9 WEITEK 4167 Numeric Coprocessor Support

The AMI 486 Enterprise motherboard supports WEITEK 4167 floating point Coprocessor. It occupies memory address range of C0000000H-C0FFFFFFH. Installation procedures for the Weitek 4167 are given in Section 7.

4.10 I/O Address Map

With the EISA bus, there are 16 address bits used for I/O decoding as opposed to 10 bits for the ISA (AT) bus. The upper four bits (bit12 - bit15) are used to determine the specific slot of the EISA bus that is being addressed. Slot 0 is reserved for system board I/O. Address bits 8 and 9 must be low to access a specific slot. If either address bit 8 or bit 9 is high, then ISA (AT) compatable I/O is being addressed and the upper six bits (bit10 - bit15) are ignored. Address bits 10 and 11 are used on EISA expansion cards to provide more I/O address space. The I/O address map for an EISA system is shown in Table 4.10.1.

Table 4.10.1

Address Range	Reserved for
0000-00FF	ISA System board peripherials
0100-03FF	ISA Expansion boards
0400-04FF	System board
0500-07FF	ISA Expansion boards (same as 0100-03FF)
0800-08FF	System board
0900-0BFF	ISA Expansion boards (same as 0100-03FF)
0C00-0CFF	System board
0D00-0FFF	ISA Expansion boards (same as 0100-03FF)
1000-10FF	EISA Expansion Slot 1
1100-13FF	ISA Expansion boards (same as 0100-03FF)
1400-14FF	EISA Expansion Slot 1
1500-17FF	ISA Expansion boards (same as 0100-03FF)
1800-18FF	EISA Expansion Slot 1
1900-1BFF	ISA Expansion boards (same as 0100-03FF)
1C00-1CFF	EISA Expansion Slot 1
1D00-1FFF	ISA Expansion boards (same as 0100-03FF)
n000-n0FF	EISA Expansion Slot "n"
n100-n3FF	ISA Expansion boards (same as 0100-03FF)
n400-n4FF	EISA Expansion Slot "n"
n500-n7FF	ISA Expansion boards (same as 0100-03FF)
n800-n8FF	EISA Expansion Slot "n"
n900-nBFF	ISA Expansion boards (same as 0100-03FF)
nC00-nCFF	EISA Expansion Slot "n"

nD00-nFFF

I/O address hex 000 through 0FF is reserved for the ISA system board I/O. I/O address hex 100 through 3FF is available for ISA compatible I/O channels. The ISA compatible system board I/O map is shown below:

Table 4.10.2

Range	Device
000-01F	DMA controller 1, 8237A-5 compatible controller
020-03F	Master Interrupt Controller, 8259A compatible
040-05F	Timer, 8254-2 compatible
060-07F	Real time clock, NMI (non-maskable interrupt) mask and Keyboard Controller (8742)
080-09F	DMA Page Register, 74LS612 compatible
0A0-0BF	Slave Interrupt Controller, 8259A compatible
0C0-0DF	DMA Controller 2, 8237A-5 compatible controller
0E0-0FF	80387 Math Coprocessor use
460-46F	Reserved for System use
1F0-1FF	Fixed Disk
200-207	Game I/O
278-27F	Parallel printer port 2
2F8-2FF	Serial port 2
378-37F	Parallel printer port 1
380-38F	SDLC, bisynchronous 2
3A0-3AF	Bisynchronous 1
3B0-3BF	Monochrome display and printer

adapter
3C0-3CF Enhanced Graphics Display adapter
3D0-3DF Color Graphics Monitor adapter
3F0-3F7 Diskette Controller
3F8-3FF Serial port 1

4.11 Integrated System Peripheral (82357)

The Integrated System Peripheral (ISP) has incorporated several CPU support functions into a single chip. It is part of a chip set designed by Intel to provide the system functions necessary to meet the EISA Specifications.

4.11.1 System Arbiter

The System Arbiter accepts requests from different sources for the control of the EISA bus. It then grants the control of the EISA bus to the requesting device (master) provided that no other master is requesting the bus. If two or more masters request the bus at the same time, then the system arbiter decides which has the highest priority and grants that master control of the bus and holds off the other masters. If there are no masters requesting the bus, the host CPU has control by default.

The priority is determined by a three way rotating scheme with the DMA Controller, Refresh, and all other masters. All three have equal priority, and EISA bus access is determined on a rotating basis. There are lower levels of priority under the DMA Controller and the other masters. Refer to Figures 4-1 and 4-2 for an illustration of the prioritization.

The other masters include the Host CPU and masters plugged into the EISA expansion slots. The Host CPU and the combination of all of the EISA masters have equal priority on a rotating basis. The EISA masters are one level lower and they each have equal priority on a rotating basis.

The DMA controller can have either fixed or

rotating priority. In the fixed mode (the default mode after reset), the DMA channels have a fixed priority with channel 0 being the highest priority and channel 7 being the lowest priority. Figure 4-1 shows the priority arbitration with the DMA in fixed priority mode. By changing the command register the DMA controller may be placed in rotate priority mode. In this mode channels 5, 6, 7, and all other channels have equal priority on a rotating basis. Channels 0, 1, 2, and 3 are one level lower and all have equal priority on a rotating basis. Figure 4-2 shows the priority arbitration in this mode.

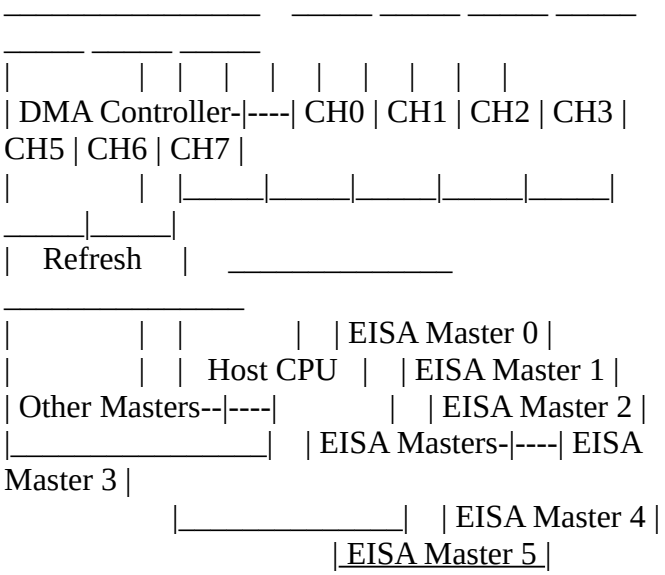

Figure 4-1 Arbitration Priority with DMA in Fixed Priority

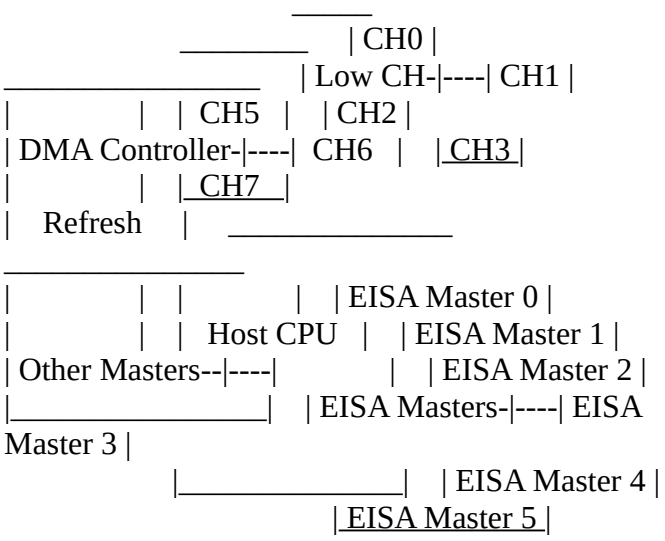

Figure 4-2 Arbitration Priority with DMA in Rotating Priority

If a system with DMA-2, DMA-3, DMA-5, Host CPU, EISA Master 1 and EISA Master 3 had requests from all sources continuously and the DMA in fixed mode the EISA bus accesses would be as follows:

DMA-2 --> CPU --> DMA-2* --> EISA Master 1 --
>
DMA-2* --> CPU --> DMA-2* --> EISA Master 2
--> repeat

* As long as DMA-2 was requesting the bus, any lower priority DMA request (DMA-3 and DMA-5 in this example) could not have access.

With the DMA in rotate mode, the EISA bus accesses would be as follows:

DMA-2 --> CPU --> DMA-5 --> EISA Master 1 -->
DMA-3 --> CPU --> DMA-5 --> EISA Master 3 -->
repeat

Any refresh requests that may come will use the bus in between the DMA and other master accesses.

4.11.2 Direct Memory Access (DMA) Controller

The system supports seven DMA channels, using two Intel 8237A compatible DMA controllers built inside the ISP. DMA controller-1 (I/O address 000H-01FH) handles channels 0 to 3, which are capable of doing 8 bit data transfers between 8-bit I/O adapters and 8-, 16- or 32- bit system memory. Each channel can transfer data throughout the 16-megabyte system address space in 64KB blocks.

DMA controller 2 supports channels 4 through 7.

Channel 4 is used to cascade DMA Controller 1. Channels 5, 6, and 7 support 16 bit data transfer between 16-bit I/O adapters and 16- or 32- bit system memory. These channels can transfer data throughout the 16-megabytes of address space in 128KB blocks. Channels 5, 6 and 7 cannot transfer data to 8- bit system memory.
The following table shows the address generation for the DMA channels. DMA page register supplies upper 8 bits of address for CTLR 1 and 7 bits of address for CTLR 2.

Controller	DMA page register supplied address	DMA controller supplied address
1	A23 through A16	A15 through A0
2	A23 through A17	A16 through A1

For DMA controller 1, 'byte high enable' signal is generated by inverting A0 signal. For DMA controller 2 both 'byte high enable' and A0 are forced low to facilitate 16-bit transfer.

The following table shows the addresses for the page register:

Page register	I/O hex address
DMA Channel 0	087
DMA Channel 1	083
DMA Channel 2	081
DMA Channel 3	082
DMA Channel 5	08B
DMA Channel 6	089
DMA Channel 7	08A
Refresh	08F

For DMA Channels 5 thru 7, the page register data bits D7 through D1 are loaded with address A23 thru A17. Data bit D0 of the page registers for Channels 5 through 7 is not used. Since DMA Channels 5 thru 7 do 16-bit data transfer, the count registers for these channels have to be loaded with half the transfer byte-count. Also, the base address registers have to be loaded with the real address divided by 2.

The following are the register addresses of DMA controllers 1 and 2:

Command Codes	Hex address for CTLR 1	CTLR 2
CH0 base and current address	000	0C0
CH0 base and current word count	001	0C2
CH1 base and current address	002	0C4
CH1 base and current word count	003	0C6
CH2 base and current address	004	0C8
CH2 base and current word count	005	0CA
CH3 base and current address	006	0CC
CH3 base and current word count	007	0CE
Read Status register/Write command register	008	0D0
Write request register	009	0D2
Write single mask register bit	00A	0D4
Write mode register	00B	0D6
Clear byte pointer flip flop	00C	0D8
Read temporary register/Write master clear	00D	0DA
Clear mask register	00E	0DC
Write all mask register bits	00F	0DE

4.11.3 Interrupt Controller

The system processor can be interrupted through NMI (Non-Maskable-Interrupt) as well as through two Intel 8259A compatible interrupt controllers built inside the ISP that provide 16 levels of system interrupt. Any or all interrupt levels, including NMI, can be masked. The following shows the interrupt level assignments in decreasing order of

priority:

TABLE 4.12.1

NMI System board Dynamic Memory parity, 32-bit slot parity or I/O channel Check

Priority	Label	Controller	Function
1	IRQ 0	1	Timer 1, Counter 0 output
2	IRQ 1	1	Keyboard Controller (Output Buffer Full)
3-10	IRQ 2	1	Interrupt from Controller 2
3	IRQ 8	2	Real-Time-Clock Interrupt
4	IRQ 9	2	EISA Bus Pin B04
5	IRQ 10	2	EISA Bus Pin D03
6	IRQ 11	2	EISA Bus Pin D04
7	IRQ 12	2	EISA Bus Pin D05
8	IRQ 13	2	Coprocessor Error
9	IRQ 14	2	EISA Bus Pin D07, Fixed Disk Controller
10	IRQ 15	2	EISA Bus Pin D06
11	IRQ 3	1	EISA Bus Pin B25, Serial Port 2
12	IRQ 4	1	EISA Bus Pin B24, Serial port 1
13	IRQ 5	1	EISA Bus Pin B23, Parallel Port 2
14	IRQ 6	1	EISA Bus Pin B22, Floppy Disk Controller
15	IRQ 7	1	EISA Bus Pin B21, Parallel Port 1

The NMI can be disabled by writing into I/O port 70H with data bit 7 set and enabled by writing to

I/O port 70H with data bit 7 reset. There are three sources through which NMI can be generated: (1) On-board dynamic RAM parity failure; (2) 32-bit adapter card memory parity failure; (3) error reported by I/O channel adapter card through `-I/O channel check (-IOCHCK) signal'. At power-on, the NMI and I/O-checks are disabled. Before NMI is enabled, following steps should be taken:

1. Write data in all system board and I/O adapter memory locations; this will establish good parity at all locations.

2. Enable On-board and 32-bit slot memory parity check (write into port 61H with data bit 2 set to zero).

3. Enable I/O channel check signal (write into port 61H with data bit 3 set to zero).

Note: All these functions are performed automatically by POST (Power On Self Test).

The status bits (I/O port 61H) indicate whether NMI is due to On-board or 32-bit slot memory parity check or I/O-check.

4.11.4 Interval Timers

The system board has five programmable timer/counters controlled by the Intel 8254-2 compatible timer/counter built inside the ISP chip. The counter/timers are addressed as two separate 8254 timers. Timer 1 contains three counters and timer 2 contains two counters. Interval Timer 1, Counter 0 is connected to IRQ0 of the interrupt controller to provide a system timer interrupt that is

used for the time-of-day, diskette time-out, and other system timing functions. Counter 1 is used for refresh requests and counter 2 is used to generate the tones for the speaker.

Interval Timer 2, Counter 0 can be used to generate NMI interrupts at regular intervals to prevent the system from locking up. Counter 1 is not used. Counter 2 is connected to the SLOWH# output and is used as a means of slowing down the host CPU in order to emulate slower systems. This may be necessary for some software that was designed to run at slow speeds. In order to use the SLOWH# feature, Counter 2 must be programmed for one-shot mode and is triggered by the refresh conter (Timer 1, Counter 1). Every time the refresh counter times out, the SLOWH# will go active for the programmed period and is used to generate a HHOLD to the host CPU in order to prevent the CPU from operating during that period and thus slowing the overall operation of the CPU by means of pulse width modulation. Counter 2 is programmed during boot-up and the HWDTRBO (hardware turbo switch) and SOFTBEN (keyboard turbo switch) are used to enable or disable this feature.

The Interval Timer functions are defined as follows:

Timer 1, Counter 0: System timer

Gate	Always enabled
Clkin	1.193 MHz clock (OSC/12)
Clkout	Interrupt controller IRQ0

Timer 1, Counter 1: Refresh Request Generator

Gate	Always enabled
Clkin	1.193 MHz clock (OSC/12)
Clkout	Dynamic memory refresh request

Timer 1, Counter 2: Tone generation for speaker

Gate	Controlled by bit 0 of I/O port address 61H
Clkin	1.193 MHz clock
Clkout	Audio frequency output to speaker

Timer 2, Counter 0: Fail-Save Timer

Gate	Always enabled
Clkin	0.298 MHz clock (OSC/48)
Clkout	NMI interrrupt

Timer 2, Counter 1: Not implemented

Timer 2, Counter 2: CPU speed control

Gate	Refresh Request (Timer 1, Counter 1)
Clkin	8 MHz (BCLK)
Clkout	SLOWH#

Note: Timer 1, Counter 1 is programmed as a rate generator of 15 microsecond period, to refresh all the Dynamic RAMs in the system.

The output of Timer Counter 2 is logically ANDed with bit 1 of the I/O port at address 61H to further modulate the output. The Timer/Counters are programmed by the system through port 40-43H for

Timer 1 and 48-4BH for Timer 2. The address map is as shown:

TABLE 4.11.1

Address	Register Select
040 (hex)	Timer 1, Counter 0
041 (hex)	Timer 1, Counter 1
042 (hex)	Timer 1, Counter 2
043 (hex)	Command Mode Register
048 (hex)	Timer 2, Counter 0
049 (hex)	(reserved)
04A (hex)	Timer 2, Counter 2
04B (hex)	Command Mode Register

4.13 Description of I/O read/write port 061H

Write port 61H bit definitions

Bit 0 Timer channel 2 GATE input control (1=GATE is enabled)
Bit 1 Timer channel 2 OUTPUT control (1=OUTPUT
will go to audio speaker)
Bit 2 Enable On-board and 32-bit slot parity check (0=Enable, 1=Disable)
Bit 3 Enable I/O channel check (0=Enable, 1=Disable)

Bit 4-7 Not Used

Read port 61H bit definitions

Bit 0 'Timer channel 2 GATE input control' bit
Bit 1 'Timer Channel 2 OUTPUT control' bit
Bit 2 'enable On-board and 32-bit slot parity check' bit
Bit 3 'enable I/O channel check' bit
Bit 4 System memory refresh determine signal (should toggle at a time period of 30 microseconds if refresh happens properly)
Bit 5 Timer channel 2 OUTPUT
Bit 6 I/O check signal (This bit will be set if there is any I/O channel check error)
Bit 7 System board RAM parity check signal (This bit will be set if there is any system board or 32-bit slot dynamic memory parity failure)

4.15 Real-Time-Clock/CMOS RAM

The RTC/CMOS RAM chip is a MOTOROLA MC146818 RTC/CMOS RAM IC. It contains the real time clock and 128 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of RAM. The rest is used to keep system configuration information.

The following table shows the CMOS RAM addresses:

Addresses	Description
00-0D	Real time clock information
0E	Diagnostic status byte
0F	Shutdown status byte
10	Diskette drive type byte - drives A and B
11	Reserved
12	Fixed disk type byte - drives C and D
13	Reserved
14	Equipment byte
15	Low base memory byte
16	High base memory byte
17	Low expansion memory byte
18	High expansion memory byte
19-2D	Reserved
2E-2F	Two-byte CMOS checksum
30	Low expansion memory byte
31	High expansion memory byte
32	Date century byte
33	Information flags (set during power on)
34-3F	Reserved

Note: The 2-byte CMOS checksum is on address 10H-20H.

These bytes can be read/written by first outputting the address to I/O port 70H and then by reading/writing data from/to I/O port 71H.

REAL TIME CLOCK (RTC) Information

The following table describes real-time clock bytes with their addresses:

Byte	Function	Address	
0	Seconds	00	
1	Second alarm	01	
2	Minutes		02
3	Minute alarm		03
4	Hours	04	
5	Hour alarm	05	
6	Day of week		06
7	Date of month	07	
8	Month	08	
9	Year	09	
10	Status Register A	0A	
11	Status Register B	0B	
12	Status Register C	0C	
13	Status Register D	0D	

Note: The built-in setup program of the system BIOS initializes registers A, B, C and D when the time and date are set. Also, software interrupt 1A is the BIOS interface to read/set the time and date. The function of the status register bits is as follows:

Status Register A

Bit 7 Update in Progress (UIP) - A 1 indicates that an update cycle is in progress or will soon begin. A 0 indicates that the time, calendar and alarm information in the RAM is fully available to read/write. Writing a 1 to the SET bit in Register B inhibits any update cycle and then clears the UIP bit.

Bits 6-422-Stage Divider (DV2-DV0) - The divider selection bits identify which of the three time base frequencies (4.194304 MHz, 1.048576 MHz and 32.768 KHz) is in use. The system initializes the divider selection bits by 010, which selects a 32.768 KHz time base.

Bits 3-0Rate Selection Bits (RS3-RS0) - These bits select the divider output frequency. The system initializes the rate selection bits by 0110, which selects a 1.024 KHz square wave output frequency and a 976.562 microsecond periodic interrupt rate if SQWE and PIE bits of register B are enabled.

Status Register B

Bit 7 SET - A 0 updates clock functions normally by advancing the counts once-per-second. A 1 aborts any update cycle in progress and the program can initialize the 14 time and calendar bytes without any update occurring in the midst until a 0 is written to this bit.

Bit 6 Periodic Interrupt Enable (PIE) - This is read/write bit which allows an interrupt to occur at a rate specified by the RS3-RS0 bits in Register A. A 1 enables the interrupt and a 0 disables it. The system initializes this bit to 0.

Bit 5 Alarm Interrupt Enable (AIE) - This read/write bit when set to 1 allows an alarm interrupt to occur. A 0 disables the interrupt. The system initializes this bit to 0.

Bit 4 Update Ended Interrupt Enable (UIE) - This read/write bit when set enables the update ended interrupt and a 0 disables it. The system initializes this bit to 0.

Bit 3 Square Wave Enabled (SQWE) - This bit when set to 1, enables a square wave signal at the frequency specified in the rate selection bits RS3-RS0 to appear on the SQW pin. The SQW pin is held low when SQWE bit is set to 0. The system initializes this bit to 0.

Bit 2 Date Mode (DM) - This bit indicates whether time and calendar updates are to use binary or BCD formats. This read/write bit, when set to 0 indicates binary and a 1 indicates BCD. The system initializes this bit to 0.

Bit 1 24/12 - This read/write bit establishes the format of the hour bytes. A 1 indicates the 24-hour mode and a 0 indicates the 12-hour mode. The system initializes this bit to 1.

Bit 0 Daylight Savings Enabled (DSE) - This read/write it when set to 1, allows the program to enable two special updates. On the last Sunday in April the time increments from 1:59:59AM to 3:00:00AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur if DSE bit is set to 0. The system initializes this bit to 0.

Status Register C

Bit 7-Bit 4 are read-only flags which are used by the program to determine the source of interrupts when the AIE, PIE and UIE interrupts are enabled in Register B.

Bit 3-Bit 0 are unused bits are read as 0 and they cannot be written.

Bit 7 Interrupt Request Flag (IRQF)

Bit 6 Periodic Interrupt Flag (PF)

Bit 5 Alarm Interrupt Flag (AF)

Bit 4 Update Ended Interrupt Flag (UF)

Status Register D

Bit 7 Valid RAM and Time bit (VRT) - It indicates the condition of the contents of the RAM through the power sense (PS) pin. A 0 appears on the VRT bit if the power sense pin is low, indicating that the real time clock has lost its power (battery dead). The processor program sets the VRT bit to 1 when the time and calendar are initialized to indicate that the RAM and time are valid.

Bits 6-0These unused bits are read as 0 and cannot be written into.

CMOS RAM Configuration Information

The following are the bit definitions for the CMOS configuration bytes (addresses hex 0E-3F):

Diagnostic Status Byte (address hex 0E)

Bit 7 Real time clock chip power status - a 0 indicates that the chip has not lost power. A 1 indicates that the chip has lost power.

Bit 6 Configuration Record Checksum Status Indicator - a 0 indicates that the Checksum is good. A 1 indicates that it is bad.

Bit 5 Incorrect Configuration Information - This bit is used to check the validity of the content of the equipment byte of the configuration record. A 0 indicates that the configuration information is valid, and a 1 indicates it is invalid.

Bit 4 Memory Size Miscompare - A 0 indicates that the power on diagnostic program check of the system memory size matches with that in the configuration record. A 1 indicates that the memory size is different.

Bit 3 Fixed Disk Adapter/Drive C Initialization Status - A 0 indicates that the fixed disk adapter and drive are functioning properly and the system can attempt to 'boot up'. A 1 indicates that the adapter and/or drive C failed initialization, which prevents the system from 'boot up'.

Bit 2 Time Status Indicator - A 0 indicates that the time is valid and a 1 indicates that the time is invalid.

Bits 1-0 Reserved.

Shutdown Status Byte (address hex 0F)

This byte defines the cause of a processor shutdown when the SYSTEM FLAG (See in Keyboard Controller description) is set. The following table defines the shutdown byte content:

Content	Cause
0	Soft Reset or Unexpected shutdown
1	Shut down after memory size

2 Shut down after memory test
3 Shut down with memory error
4 Shut down with boot loader request
5 JMP DWORD request (with interrupt controller INIT)
6 Protected mode test passed
7 Protected mode test failed
8 Protected mode test failed
9 Block move shut down request
A JMP DWORD request (w/o interrupt controller INIT)

Diskette Drive Type Byte (address hex 10)

Bit 7-Bit 4	Drive A type:

0000	No drive present
0001	Double Sided Diskette Drive (48 TPI, 360KB)
0010	High Capacity Diskette Drive (96 TPI, 1.2MB)
0011	3 1/2" Diskette Drive (96 TPI, 720KB)
0100-1111	Reserved

Bit 3-Bit 0	Drive B type:

Same as Drive A

Fixed Disk Type Byte (address hex 12)

Bit 7-Bit 4	Drive C type:

0000	No fixed disk drive is present

0001 through 1110 define type 1 through type 14. The type list is given in Appendix B.

1111	Hard Disk type 16 through 255 is indicated in the extended byte (address hex 19)

Bit 3-Bit 0	Drive D type:

The bit definitions are same as above, except for 1111 which implies that the Hard Disk type between 16 and 255 is indicated in the extended byte (address hex 1A).

Equipment Byte (address hex 14)

Bit 7-Bit 6 Total number of diskette drives installed:

00 1 drive
01 2 drives
10 Reserved
11 Reserved

Bit 5-Bit 4 Primary Display

00 Primary display is other than Color Graphics Adapter or Monochrome Displ
ay adapter.
01 Primary display is Color Graphics Adapter in 40 column mode.
10 Primary display is Color Graphics Adapter in 80 column mode.
11 Primary display is Monochrome Display adapter.

Bit 3-Bit 2 Not used

Bit 1 Math Coprocessor 80387 presence bit:

0 Math Coprocessor 80387 not installed
1 Math Coprocessor 80387 installed

Bit 0

1	Diskette drives are installed
0	Diskette drives are not installed

High Base Memory Bytes (address hex 15 and 16)

The size of memory below 1MB is indicated by the 16-bit word formed by these two bytes (address 16H is the higher byte). The content of this word increments by 40H for every 64 KB increment of base memory size. For example, if the system contains 256 KB, the content of the base memory bytes (15H and 16H together) will be 0100H (content of 16H is 01H and content of 15H is 00H). The maximum possible size of memory below 1 MB is 640KB. The maximum possible value of base memory bytes is 0280H.

Low and High Memory Expansion Bytes (address hex 17 and 18)

The size of memory above 1MB is indicated by the 16-bit word formed by these two bytes (address 18H is the higher byte). In this case also, the content of this word increments by 40H for every 64 KB increment of expansion memory size. For example, if the content of memory expansion bytes is 0200H (content of 18H is 02H and content of 17H is 00H), then the amount of memory above 1 MB is 512 Kilobytes. The maximum possible value is 3C00H for a maximum of 15 MB of memory above 1MB.

Drive C Extended Byte (address hex 19)

Bit 7 - 0 Defines the type of Fixed Disk Drive installed

(Drive C). 00000000 thru 00001111 are reserved. 00010000 thru 11111111 define the types 16 thru 255.

Drive D Extended Byte (Hex 1A)

Bit 7 - 0 Definition is the same as Drive C extended byte.

Checksum (address hex 2E and 2F)

Address hex 2E - Low byte of checksum
Address hex 2F - High byte of checksum

The checksum is calculated by adding the contents of addresses 10H through 20H. The low byte of the checksum is stored in hex 2E and the high byte in hex 2F. **Low and High Expansion Memory Bytes (address hex 30 and 31)**

The size of memory above 1MB is indicated by the 16-bit word formed by these two bytes (address 31H is the higher byte), as determined at power-on time. The content of this word increments by 40H for every 64 KB increment of expansion memory size. Maximum possible value is 3C00H for a maximum of 15 MB of memory above 1MB.

Date Century Byte (address hex 32)

Bit 7-Bit 0 BCD value for the century (BIOS interface to read and set).

Information Flag (address hex 33)

Bit 7 Set if memory is present in address space hex 080000-09FFFF

Bit 6 Used by the Setup utility to put out a first user message after initial setup Bit 5-Bit 0 Reserved

Note: Hex addresses 11, 13, 1B-2D, 34-3F are reserved.

4.16 Keyboard Controller

The keyboard controller is a 8742 single chip microcomputer and is used to support a 486 PC keyboard interface. The controller has the following functions:

-Receive serial data from the keyboard, checking parity of the data and translating it to system scan code, if necessary. Put the received and processed data into the data buffer and interrupt processor.

-Execute system commands through the controller command buffer and place the result, if necessary, in the data buffer and interrupt the processor.

-Transmit system data, placed in the data buffer, to the keyboard in a serial format with the parity bit inserted. Get the response from the keyboard and report to the system.

-Report any error to the system through status register at the time of data communication with the keyboard.

Receiving data from the keyboard

The keyboard sends data in a 11-bit serial format. The first bit is a start bit (low level) followed by 8 data bits (least significant data bit first), an odd parity bit and a stop bit (high level). Data sent is synchronized with the keyboard clock. Upon receiving a byte of data from the keyboard, the keyboard controller places the data in its one byte receive-data buffer and disables the keyboard interface until that data is picked up by the system processor. This avoids data overrun. On parity errors, the controller requests the keyboard to resend the data. If the error is repeated, the controller sets the parity error bit in its status register. If all the 11 bits are not received within 2 milliseconds from start of transmission, a time-out error is indicated by setting the time-out bit in the status register. In case of either of these errors, hex FF is placed in the receive-data buffer.

Sending data to the keyboard

Data is sent to the keyboard in the same serial format as data received from the keyboard. If the time between request to send and start of transmission is greater than 15 milliseconds or the duration of transmission is greater than 2 milliseconds, the transmit time-out error bit is set in the status register. The keyboard is required to acknowledge every transmission from the controller. If the acknowledgement has a parity error, then the controller sets both the parity and transmit time-out error status bits. Also, if the acknowledgement does not arrive within 25 milliseconds, both the receive and transmit time-out

error bits are set. In case of all these errors, hex FE is placed in the data buffer. No retries are made for the error at the time of transmitting to the keyboard.

Keyboard Inhibit

The keyboard can be inhibited through keylock jumper J9. When the keyboard is inhibited, although all transmissions from the system to the keyboard will be allowed, the keyboard controller tests all data received from the keyboard. If it is a response to a command sent to the keyboard, then it is placed in the data buffer. Otherwise, it is ignored.

Keyboard Controller - System Interface

The system communicates with the keyboard controller through an input buffer, an output buffer and a status register. The status register can be read through I/O port 64H. The output buffer can be read through I/O port 60H. The input buffer can be written through both I/O port 64H and 60H. When the input buffer is written through I/O port 64H, the controller interprets it as a command and if it is written through I/O port 60H, then the data is interpreted either as a parameter to a command to the controller or data to be transmitted to the keyboard.

Keyboard Controller Status Register Bit definitions

Bit 0 Output Buffer Full - A 0 indicates that the keyboard controller's output buffer has no data. When the keyboard controller writes to the output buffer, this bit is set to 1. It returns to 0 when the

system reads the output buffer (60H).
Bit 1 Input Buffer Full - A 0 indicates that the keyboard controller's input buffer (60H or 64H) is empty. When the system writes to the input buffer, this bit is set to 1. It gets reset to 0 when the controller reads the input buffer.

Bit 2 System Flag - The keyboard controller can set this bit to 0 or 1 depending on the command from the system. It is set to 0 after power on reset.

Bit 3 Command/Data - This bit is used by the keyboard con-troller to determine whether the input buffer contains the command or data. When the system writes to the input buffer through I/O port 64H, this bit is set to 1. When the system writes to the input buffer through I/O port 60H, this bit is set to 0.

Bit 4 Inhibit Switch - This bit reflects the state of the keyboard inhibit switch. This bit is updated whenever the controller writes to the output buffer. A 0 indicates that the keyboard is inhibited.

Bit 5 Transmit Time-Out - A 1 indicates that a data transmission from the keyboard controller to the keyboard was not properly completed within the predefined time limit.

Bit 6 Receive Time-Out - A 1 indicates that a data transmission from the keyboard to the keyboard controller was not properly completed within the predefined time limit.

Bit 7 Parity Error - A 1 indicates that the last byte received from the keyboard had a parity error. The keyboard sends data with odd parity.

Keyboard Controller I/O ports

The keyboard controller has two 8-bit I/O ports. One is used as an input port and the other as an output port. The following show bit definitions for the I/O ports.

Input Port Definitions

Bit 0 Diagnostic LED.
This is used to indicate that the board is performing alright when the manufacturing diagnostic mode is enabled. LED will blink in manufacturing diagnostic mode.

Bit 1-3 Undefined

Bit 4 RAM on the system board
0=Total 256 KB of on-board RAM
1=512 KB or greater on-board RAM

Bit 5 Manufacturing diagnostics. When this bit is 0, the system BIOS performs diagnostics on the system board in an infinite loop.

Bit 6 Display type switch
0=Primary display is color graphics adapter
1=Primary display is monochrome display adapter

Bit 7 Keyboard inhibit switch
0=Keyboard inhibited
1=Keyboard not inhibited

Output Port Definitions

Bit 0 Reset to the system processor (software should keep it 1 for the system Processor to work)

Bit 1 Gate address 20 of system processor

0: The system processor address 20 is inhibited on the system bus. Address 20 will remain zero for any system processor bus cycle.

1: The system processor address 20 is allowed on the system bus.

Bit 2 Undefined

Bit 3 Undefined

Bit 4 Output buffer full interrupt to the system

Bit 5 Input buffer full

Bit 6 Keyboard clock (output)

Bit 7 Keyboard data (output)

System Commands to Keyboard Controller (I/O address 64H)

20 Read keyboard controller's command byte - The controller sends its current command byte to its output buffer.

60 Write keyboard controller's command byte - The next byte of data supplied by the system is the

controller's command byte, which is written to I/O address Hex 60.

The bit definitions are as follows:

Bit 7 not used

Bit 6 IBM PC compatibility mode - When this bit is set, the controller converts the scan code received to the IBM PC compatible scan code. This includes converting the two byte break sequence received from the AT compatible keyboard to the one byte PC break code format.

Bit 5 Reserved - Should be 0 for proper operation

Bit 4 Disable keyboard - When set to 1, this disables the keyboard interface by driving the 'clock' line low. Data is not received.

Bit 3 Inhibit override - Writing 1 to this bit disables the keyboard inhibit function through keyboard clock.

Bit 2 System Flag - The keyboard controller writes the value written in this bit to bit 2 of the status register.

Bit 1 Not used

Bit 0 Enable output buffer full interrupt - When this bit is set the controller generates an interrupt to the system when it places data into its output buffer.

AA Self Test - This commands the controller to perform internal diagnostics. The controller returns 55 to the output buffer if no error is detected.

AB Interface test - This commands the controller to test the keyboard clock and data lines. The test result placed in the output buffer is as follows.

00	No error detected
01	Keyboard clock line stuck low
02	Keyboard clock line stuck high
03	Keyboard data line stuck low
04	Keyboard data line stuck high

AD Disable Keyboard Interface - This command disables the keyboard interface by driving the clock line low. It also sets bit 4 of the controllers command byte.

AE Enable Keyboard Interface - This command enables the keyboard interface and also clears bit 4 of the controller's command byte.

C0 Read Input Port - This commands the keyboard controller to read the input port and place the data in the output buffer.

D0 Read Output Port - This commands the keyboard controller to read the output port and place the data in the output buffer.

D1 Write Output Port - This command is used to write the data given through I/O address 60H to the output port. Ensure that the output port bit 0 is not written as 0 since it is connected to the reset of the system processor.

E0 Read Test Inputs - Upon receiving this command the controller reads its T0 and T1 inputs

and places the data in the output buffer. Data bit 0 represents T0 and bit 1 represents T1.

F0-FF Pulse output port - Bit 0 through 3 of output port can be pulsed for approximately 6 microseconds by this command. Bit 0 through 3 of this command indicates which bits of the output port are to be pulsed. A 0 indicates that the bit is to be pulsed and a 1 indicates that the bit is to be kept unmodified. Note that bit 0 of the output port is connected to the reset of the system microprocessor so the processor can be reset by pulsing this bit.

Keyboard controller - Keyboard Interface

The keyboard controller communicates with the keyboard over a clock line (bit 6 of the output port 5) and a data line (bit 7 of the output port). The keyboard controller reads the data line through a test input T1 and the clock line through a test input T0. For any type of data transmission with the keyboard, the keyboard supplies the clock. Data is made available after the rising edge of the clock and is sampled on the falling edge. The hardware protocol for communication with keyboard is given below.

When the keyboard wants to send data, it first checks the clock line for a high level (the keyboard controller can prevent the keyboard from sending data by driving the clock line low through bit 6 of the output port). If the clock and data lines are high (i.e enabled), the keyboard sends the data. Otherwise it stores data in its own buffer.

The keyboard checks the state of the clock line at an

interval of 60 microseconds, in order to sense whether the keyboard controller intends to send data. When the keyboard controller wants to send data, it forces the clock line low for more than 60 microseconds and then releases it with the data line low. This low data is accepted by the keyboard as a start bit (request to send) and it starts clocking the data in. After the tenth bit, the keyboard forces the data line low for one clock period (the stop bit). This action informs the keyboard controller that the keyboard has received its data.

4.17 EISA signal description

The Extended Industry Standard Architecture (EISA) is an extension of the ISA (AT-bus) 8- and 16-bit architecture. EISA systems are fully compatable with ISA system boards while allowing for the several major advances. The following is a description of the EISA signals. For more information about EISA refer to EISA Specification Version 3.10 from BCPR Services, Inc.

BE0#-BE3# (EISA)

Byte Enable 0 thru 3 are used to indicate the portion of the 32 bit (4 byte) data bus that is being addressed for the present bus cycle. BE0# is the least significant byte and BE3# is the most significant byte.

D0-D7 (ISA)

Data bits 0 thru 7 are the least significant byte of the 16 bit or 32 bit data bus. These bits are used by any 8 bit device, by 16 bit devices when SA0, BE0#, or BE1# is active, or by 32 bit devices when BE0# is

active.

D8-D15 (ISA)

Data bits 8 thru 15 are the high byte of the 16 bit data bus and the second least significant byte of the 32 bit data bus. These bits are used by 16 bit devices when SBHE#, BE1#, or BE3# is active, or by 32 bit devices when BE1# is active.

D16-D23 (EISA)

Data bits 16 thru 23 are the second most significant byte of the 32 bit data bus. These bits are used by 32 bit devices when BE2# is active.

D24-D31 (EISA)

Data bits 24 thru 31 are the most significant byte of the 32 bit dat bus. These bits are used by 32 bit devices when BE3# is active.

LA2-LA16 (EISA)

Latchable Address bits 2 thru 16 are part of the 32 bit address bus. These bits along with LA17-LA31 and BE0#-BE3# are used to form the full address for the EISA bus.

LA17-LA23 (ISA)

Latchable Address bits 17 thru 23 are second most significant byte of the 32 bit address bus.

LA24#-LA31# (EISA)

Latchable Address bits 34 thru 31 are the most significant byte of the 32 bit address bus. These bits are active low whitch means they are inverted from the rest of the address bus. A low signifies that the address bit is being accessed.

SA0-SA19 (ISA)

These address bits are used to address system memory and I/O devices. These lines, along with LA17 through LA23, allow up to 16MB of memory access. SA0 through SA19 are gated on the system bus when `BALE' is high and latched on the falling edge of `BALE'. These signals are driven by system board microprocessor or DMA controller. They can also be driven by other bus masters residing on the EISA Bus by activating the `-MASTER' signal.

LA17-LA23 (I/O)

These signals are used to address memory and I/O devices within the system. They are not latched, and they are valid only when `BALE' is high. These signals may not remain valid throughout the whole processor cycle. Their purpose is to generate memory decodes for 1 wait state memory cycles. These decodes should be latched by I/O adapters on the falling edge of `BALE'. These signals may be driven by other microprocessors or DMA controllers, residing on the EISA bus, by activating the `MREQx' signal.

SBHE (I/O)

`Byte high enable', when low, indicates a transfer of data on the upper byte of the 16-bit data bus (SD8 through SD15). Sixteen bit devices use this signal to condition the data bus buffers connected to SD8 through SD15.

SD0-SD15 (I/O)

These are data bus signals 0 through 15 for memory and I/O devices on the I/O adapter cards. SD0 is the least significant and SD15 is the most significant bit. All communications to 8-bit devices on the I/O channel should be through SD0 to SD7. For 16-bit devices on I/O channel SD0 through SD15 is used. 32- or 24-bit transfers, as well as misaligned 16-bit ones, from the 486 are split up into two cycles of 16 bit or less. The data is gated to the appropriate part of the 32-bit processor data bus. Similarly, 16 bit transfers to 8-bit devices will be again converted into two 8-bit transfers with SD8 to SD15 gated to SD0 to SD7 whenever necessary.

BALE (Buffered ALE) (O)

`Address latch enable' is used to latch valid addresses and memory decodes on the system board. To the I/O channel, it indicates a valid CPU address. When used with AEN, it indicates a valid DMA address. Microprocessor addresses SA0 to SA19 are gated when BALE is high and are latched with the falling edge of BALE. BALE is forced high when the system processor goes to hold state so that the address and memory decode latches become flow-through.

RESETDRV (O)

This active high signal is used to reset or initialize system logic at power-up, during hard reset or a low line-voltage.

BCLK (O)

For system boards running at 25 or 33 MHz, the time period of this clock is determined by the SYSCLK oscillator at position **Y2** on the board. If an 8 MHz oscillator is put then the time period is 125 nanoseconds. This clock has a 50% duty cycle and is asynchronous with the microprocessor clock. This signal should only be used for synchronization. It is not intended for uses requiring a fixed frequency.

IOCHCK (I)

This signal is used by the I/O adapter channel cards to report any fatal errors (e.g, parity error information on I/O channel memory cards or error on I/O devices).

IOCHRDY (I)

`I/O channel ready' is used by memory or I/O devices to extend memory or I/O cycles. A slow device, requiring more than the bus cycle time provided by the system board, should pull IOCHRDY low (not active) immediately after detecting valid address together with Read or Write command. Machine cycles are extended by an integral number of SYSCLK cycles (e.g, 125 ns at 8 MHz). This signal should not be held low for more than 2.5 microseconds.

IRQ3-IRQ7, IRQ9-IRQ12 and IRQ14 to IRQ15

(I)

These positive edge triggered Interrupt Requests are used by I/O channel adapter cards to indicate to the microprocessor that it requires attention. The interrupt requests are prioritized. The priority is as shown in decreasing order: IRQ9 through IRQ12, IRQ14, IRQ15 and IRQ3 through IRQ7. An interrupt request is generated by raising an IRQ line from low to high. The line must be held high until the processor acknowledges the interrupt request (Interrupt service routine). IRQ0-IRQ2, IRQ8 and IRQ13 are used by the system board.

IORD# (I/O)

`I/O read', an active low signal, instructs the selected I/O device to drive its data onto the data bus. This signal may be driven by the system microprocessor, system board DMA controllers, or by a microprocessor or DMA controller resident on the I/O channel.

IOWR# (I/O)

`I/O write', an active low signal, instructs the selected I/O device to read data from the data bus. It may be driven by the system microprocessor, system board DMA controllers, or by a microprocessor or DMA controller resident on the I/O channel.

SMEMR# (O), MEMR# (I/O)

These signals instruct the memory devices to drive data onto the data bus. Both of these signals are active low. SMEMR# signal is generated for any read in low 1MB of memory space, and can be driven by the system board microprocessor and DMA controllers only. MEMR# is active in all I/O channel memory read cycles and can be driven by any microprocessor or DMA controller in the system. SMEMR# is generated from MEMR# qualified with the decode of the low 1MB of memory. SMEMR# or MEMR# are not generated if the processor access is on the system board memories. When a microprocessor or a DMA controller on the I/O channel wants to drive the MEMR# signal, it must drive the address lines valid at least one SYSCLK period before driving MEMR# active.

SMEMW# (O), MEMW# (I/O)

These signals, both active low, instruct the memory devices to store the data present on the data bus. MEMW# signal is active in all I/O channel memory write cycles and can be driven by any microprocessor or DMA controller in the system. SMEMW# signal is active in all memory write cycles in low 1MB of memory space. It is generated from MEMW# signal, qualified with the decode of the low 1MB of memory. SMEMW# or MEMW# are not generated if the processor access is on the system board memories. SMEMW# can only be driven by the system board microprocessor and DMA controllers. When a microprocessor or a DMA controller, on the I/O channel, wants to drive the MEMW# signal, it must drive the address lines valid at least one SYSCLK period before driving MEMW# active.

DRQ0-DRQ3 and DRQ5-DRQ7 (I)

DMA requests 0 through 3 and 5 through 7 are asynchronous, active high, channel requests. They are used by peripheral devices and I/O channel microprocessors, to gain DMA service or control of the system. An I/O channel microprocessor can gain control of the system by activating a DMA request line, and then activating -MASTER signal after getting DMA acknowledge (DACK) signal. The DMA request lines are prioritized with DRQ0 having the highest priority and DRQ7 having the lowest. A DMA request is generated by bringing the DRQ line to an active (high) level and keeping it active until the corresponding DACK signal goes active. DRQ0-DRQ3 are used for 8-bit data transfers between 8-bit I/O device and 8/16-bit memory device. DRQ5-DRQ7 are used for 16-bit data transfer between 16-bit I/O device and 16-bit memory device. DRQ4 is used in the system board for cascading the two DMA controllers, and is not available on the I/O channel.

-DACK0 to-DACK3 and -DACK5 to -DACK7 (O)

DMA acknowledge signals 0 through 3 and 5 through 7 are used to acknowledge requests on the corresponding DMA request lines. They are active low.

REFRESH# (I/O)

This active low signal is used to indicate a dynamic memory refresh cycle and can be driven by any microprocessor on the I/O channel. This signal is generated after every 15 microseconds.

AEN (O)

`Address enable', high active, signal is used to degate the microprocessor and other devices from the I/O channel, to enable DMA transfers to take place. When this signal is active, the DMA controller has the control of address bus, data bus, memory control (SMEMR#, SMEMW#, MEMR#, MEMW#) and I/O control (IORD#, IOWR#) signals.

T/C (O)

`Terminal count', normally low, provides a pulse when any DMA channel reaches the terminal count.

-MASTER (I)

This signal is used by a microprocessor or DMA controller residing on the I/O channel to gain control of the system bus. The procedure to gain control of the system bus is as follows:-

* Issue a DRQ to a DMA channel in cascade mode.

* Upon receiving a -DACK, pull the -MASTER signal low to gain control of the system address, data and control lines.

* After -MASTER is low, the I/O microprocessor should wait at least one SYSCLK period before driving the address and data lines and two SYSCLK period before driving the control lines (MEMR#, MEMW#, IORD#, IOWR#).

If -MASTER signal is held low for more than 15 microseconds, system memory data may be lost due

to lack of refresh. The I/O microprocessor (in control) can take care of this by activating the REFRESH# signal.

MCS16# (I)

`-MEM 16 chip select' indicates to the system board whether the present data transfer is a 16-bit 1 wait-state memory cycle (true at 8 MHz system board operation). It must be derived from decoding address LA17 through LA23. `MCS16#' is an active low signal and should be driven with an open collector or tri-state driver capable of sinking 20mA.

IOCS16# (I)

`-I/O 16 bit chip select' signals to the system board that the present I/O data transfer is a 16-bit I/O cycle, requiring at least 375ns. This active low signal is derived from an address decode, and should be driven with an open collector or tri-state driver capable of sinking 20 mA.

OSC (O)

`Oscillator' is a 14.31818 MHz clock with a duty cycle of 50%. This signal is not synchronous with SYSCLK. This signal is used in Color Graphics Adapter card.

0WS# (I)

`0-wait state' signal is used to tell the system microprocessor that the present bus cycle can be completed without inserting any additional wait

states at 8 MHz. In order to run a memory cycle on a 16-bit memory device, without wait states, 0WS# signal should be derived from an address decode gated with a read or write command (also MCS16# signal has to be activated). In order to run a memory cycle to an 8-bit device with a minimum of 2-wait states, 0WS# should be driven active one system clock after the read or write command is active, gated with the address decode for the device. 0WS# is an active low signal and should be driven with an open collector or tri-state driver capable of sinking 20mA.

THIS PAGE INTENTIONALLY

LEFT BLANK