1-800-44-MYLEX
1-415-657-7667
683-4600
1-415-683-4610
GEORGE DEL CARPIO
688-4674

MYLEX

Mylex MI386

16MHz and 20MHz

Technical Reference Manual

Revision 1.0

November, 1988

Table of Contents

Chapter One

System Overview

1.1 Product Description

The **Mylex MI386** system board is a high performance Intel 80386 based IBM AT form factor board. It may be used as a mother board component in a new system, or as a replacement mother board for an existing installation. The board is available with two different processor clock speeds as shown in the following table:

Part Number	CPU	On-board SRAM
MI386-16	16MHz, 32-bit	64KB, 35/40ns
MI386-20	20MHz, 32-bit	64KB, 25/35ns

The part number of a Mylex MI series board includes the clock speed at which the board runs, as shown in the table above.

The DRAM installed on MI386-XX boards may be consist of either Single In-line Memory Modules (SIMMs) or DIP SIMM modules. A SIMM has 9 DRAM chips surface mounted on a small board. Each board may have either 256Kbit × 9 or 1Mbit × 9 DRAM chips, providing either 256KB or 1MB of on-board memory per module. DIP SIMMs also are small boards with 9 DIP DRAM chips mounted on them. Although they provide the same amount of memory per module as the SIMMs (that use DRAM chips of the same size), DIP SIMMs have a larger profile.

Mylex part numbers that begin with "SM" refer to SIMM modules and those that begin with "DS" refer to DIP SIMM modules. The operating speed of the DRAM devices is either 100ns or 120ns. For optimum performance MI386-20 boards should use only 100ns DRAMs. Using 100ns DRAMs with the MI386-16 boards does not improve performance. For this reason, it is more economical to use 120ns DRAMs with the MI386-16 board.

Another attribute of DRAMs is their storage capacity. 256Kbit or 1Mbit DRAM chips are used in SIMMs and DIP SIMMs. There are eight sockets on the MI386-XX board for holding SIMMs or DIP SIMMs organized in two banks. Four 256Kbit × 9 modules occupy one bank to provide 1MB of on board RAM. Each bank, if used, should be **completely** filled with the same type of modules (256Kbit or 1Mbit). Also, if the second bank is used, it should be filled with the same type of modules. This implies that the size of the total on-board DRAM can **only** be 1MB, 2MB, 4MB or 8MB.

The following table shows the part numbers of memory modules with which MI386 boards are available:

Part Number	SIMM/DIP SIMM	DRAM (Size, Speed)
SM256-100	SIMM	256K-bit, 100ns
SM256-120	SIMM	256Kbit, 120ns
SM1024-100	SIMM	1024Kbit, 100ns
SM1024-120	SIMM	1024Kbit, 120ns
DS256-100	DIP SIMM	256Kbit, 100ns
DS256-120	DIP SIMM	256Kbit, 120ns
DS1024-100	DIP SIMM	1024Kbit, 100ns
DS1024-120	DIP SIMM	1024Kbit, 20ns

1.2 Features and Specifications

The main features of the MI386 board are listed below:

- IBM AT form-factor: approximately 12" by 13" (or 30.5cm by 33cm)
- Intel 80386 Microprocessor
- Optional 80387 Numeric Coprocessor: 16MHz for MI386-16, or 20MHz for MI386-20
- Battery operated system clock

- User selectable synchronized system clock switching option for reducing the processor clock frequency to 8MHz for the MI386-20 the MI386-16.
- SIMM (Single In-line Memory Module) pack or DIP SIM Modules on mother board (two banks with a maximum of 8MB DRAM)
- 32-bit memory connector (aligned with one 8-bit I/O slot)
- 64KB direct mapped cache memory to cache 16MB memory address space
- Random Access Memory (RAM) subsystem
- Complementary Metal Oxide Semiconductor (CMOS) RAM to maintain system configuration, with battery backup
- Keyboard and speaker attachments
- Seven channel Direct Memory Access (DMA)
- 16 level interrupt
- Three programmable timers
- I/O bus operation at 10MHz or 8MHz for the MI386-20 and 8MHz for the MI386-16
- Six I/O slots with a 62 + 36 pin card-edge socket ("16 bit" IBM AT compatible I/O slot)
- 2 I/O slots with a 62-pin card-edge socket ("8 bit" IBM AT/PC compatible I/O slot)

1.3 The Intel 80386 Microprocessor

The Intel 80386 is a high performance 32-bit microprocessor designed for multitasking operating systems. The processor can address up to 4 gigabytes of physical memory and 64 tetrabytes (1 tetrabyte = 1 K gigabyte) of virtual memory (thereby limiting the physical address space to 16MB). It incorporates integrated memory management and protection in its architecture, in the form of 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 80386 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 Real Address mode it operates as a fast 8086, with 32-bit extension if desired. The Protected mode is the natural environment of the 80386, and software can perform a task switch into tasks designated as virtual 8086 mode tasks. Virtual 8086 tasks can be isolated and protected from one another by the use of paging and the I/O permission bit map.

1.4 Clock Speed

The Mylex MI386-16 operates at a system clock speed of 16 MHz (or clock cycle time of 62.5ns), and the MI386-20 operates at 20 MHz (or clock cycle time of 50ns). Both boards can run at a slower clock rate of 8MHz. The clock speed can be switched at any time when the machine is operating.

The 80387 numeric coprocessor runs at 16MHz (for the MI386-16) and at 20Hz (for the MI386-20).

1.5 Data Access: Bus Width

The 80386 microprocessor supports two types of accesses: Memory, and Input/Output. Each type of access can be 32, 24, 16 or 8 bits wide. Memory and I/O devices are 32, 16 or 8 bit wide. The MI386 allows any type of access to a device of any width. If necessary, the hardware can break up a 80386 bus cycle into the required number of cycles (up to 32 / 8 = 4), to allow access to a 16 or 8 bit device. All the on-board devices are organized into a 32-bit wide memory space. These include the DRAM, the high-speed cache memory and the EPROM which contains the BIOS.

The MI386 can support 16 bit and 8 bit memory and I/O devices in IBM AT compatible I/O slot, in addition to a 32-bit memory devices.

Chapter Two

Installation

2.1 Unpacking the MI386 System Board

This chapter will guide the user in the hardware installation of the Mylex MI386 mother-board. The procedure will include the mounting and configuring of the system board. The installation of peripherals and other devices on the system are outside the scope of this section. In order to perform this procedure, you will need:

- An IBM AT compatible system enclosure with the appropriate mounting hardware. The mounting hardware should include four nylon standoff pieces and two mounting/grounding screws. These may be removed from the mother-board in an existing system.
- A philips or flat head screwdriver, as appropriate, depending on the type of screws that are used with the enclosure.
- A Mylex MI386 System board

Some experience in handling computer equipment would be useful in performing this procedure. However, this procedure can be accomplished by a person who does not have a technical background, as long as reasonable care is taken. Because no high-voltage components are involved, and there should be no hazard in performing this procedure as long as a UL approved power supply is used.

Every effort has been made to make sure that your Mylex MI386 system board is ready for operation at the time of shipment. Our experience has shown that the most common cause for board failure after shipment and before operation for the first time is static electricity. Static electricity can also cause damage that will not affect the operation of components until well after the unit has been put in service. Standard precautions are listed on the following page.

- Leave the system board in its original packaging for as long as possible.

- Cold days and dry air are likely to cause static electricity problems. In general, avoid environments that are conducive to the generation of static electricity.

- When unpacking and handling the board and other system components all materials should be placed on a grounded anti-static mat. The operator should wear an anti-static wristband connected to the same ground as the anti-static mat. (A less expensive solution is to use a sheet of conductive aluminum foil grounded through a 1 mega-ohm resistor, instead of the anti-static mat. Instead of a wrist-band, a strip of aluminum foil may be wrapped around the operator's wrist, and grounded through a 1 mega-ohm resistor.)

Inside the carton, the system board is packed in an anti-static bag, and sandwiched between sheets of sponge. Extract the system board and place it on the grounded surface described above, with the component side up.

Save the original packing materials for use in the unlikely event that the system board needs to be returned to Mylex or a Mylex authorized distributor.

Inspect the system board for damage. Components mounted on sockets may be pressed down to make sure they are properly seated. If any evidence of damage to the system board is apparent, **do not** apply power or attempt to continue installation without obtaining authorized technical assistance.

The MI386 system board is now ready for installation.

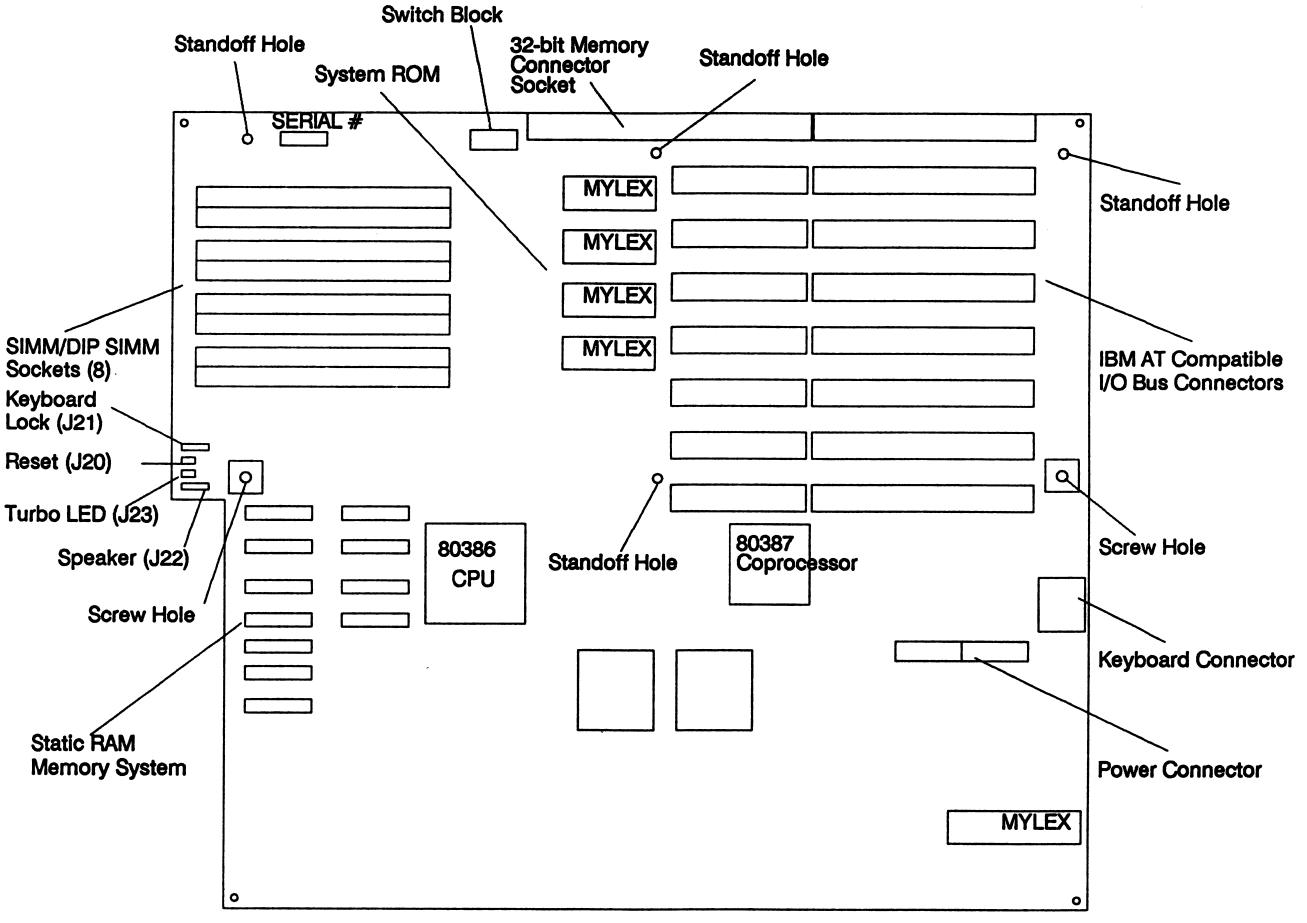


Figure 2-1: MI386 System Board

2.2 Installing the MI386 System Board

Before attempting to mount the board in the chassis, study the system board diagram (Figure 2-1) to get aquainted with the locations of stand-offs and mounting screws. (Stand-offs and mounting screws are supplied with the chassis; not with the system board. If you are upgrading your system, these parts can be removed from the old system board.)

Place the chassis on the anti-static mat and remove the cover. Keep the plastic clips, nylon stand-offs and screws for mounting the system board separately.

The chassis has to be connected to a ground to avoid damage to the board due to static electricity while mounting it. To ground the chassis, connect an alligator clip with a wire lead to any unpainted part of the chassis. Ground the other end of the lead at the same point as the mat and the wristband. Rotate the chassis so that the front is to your right and the rear to your left. The side facing you is where the system board is to be inserted. The power supply will be mounted on the far side.

Take the four nylon stand-offs and push them into the holes provided for them in the system board (See Figure 2-1) from the bottom side. Note that there are other holes in the mother-board that are not intended for the stand-off pieces. The stand-offs will lock in place.

Make sure that the two plastic clips that hold the far end of the board are installed on the chassis. On AT style cases, these clips can be hard to reach, and it may be necessary to dismount the hard disk drive and/or the power supply to install them.

On the chassis, locate the holes where the stand-offs go in. Hold the system board (component side up) with the edge with three stand-offs towards you, and the edge with no stand-offs away from you. (The edge connectors on the system board for the adaptor cards should be on your left.) Carefully slide the board into the chassis, making sure that the stand-offs go into the slots provided for them. If the stand-offs are properly locked, the board should not slide right or left (forwards and backwards with respect to the chassis) and should be level with the

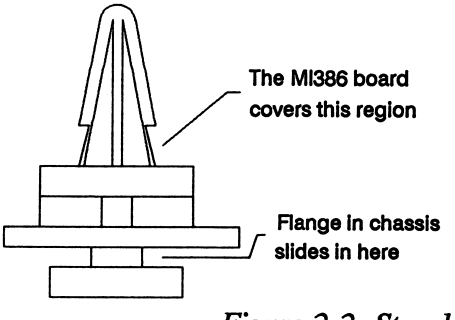


Figure 2-2: Standoff Piece

chassis. The far edge of the board should fit into the slots in the plastic clips. In case the board is not properly in place, **slide it out completely** and try again. On most AT style enclosures, the board will appear to fit loosely. To make sure that the board is mounted properly check: 1) the alignment of the mounting screw holes on the system board with their counterparts in the chassis, and 2) the elevation of the board with respect to the chassis. It is possible for a standoff to be in position laterally, but not vertically. (See Figure 2-2).

After the board is properly positioned, insert and tighten the two mounting screws at the front and the back of the board. Do not use excessive torque on these screws.

2.3 Switch Settings

After checking that the system board is properly installed, the next step is the setting of the DIP Switches.

The MI386 mother board has one 8-switch DIP switch that is to be set. The DIP switch (SW) is located near the 32-bit memory connector marked J11. The switches 1 through 8 should be set as appropriate, based on the following description. Also refer to Figure 2-3.

Switch 1: ON: 512KB of conventional memory

OFF: 640KB of conventional memory

Switches 2 and 3: Refer to Section 2.4

Switch 4: ON: 256K DRAMs are in use

OFF: 1024K DRAMs are in use

Switch 5: ON: Use on-board EGA BIOS

OFF: Do not use on-board EGA BIOS

Switch 6: OFF: Factory default setting

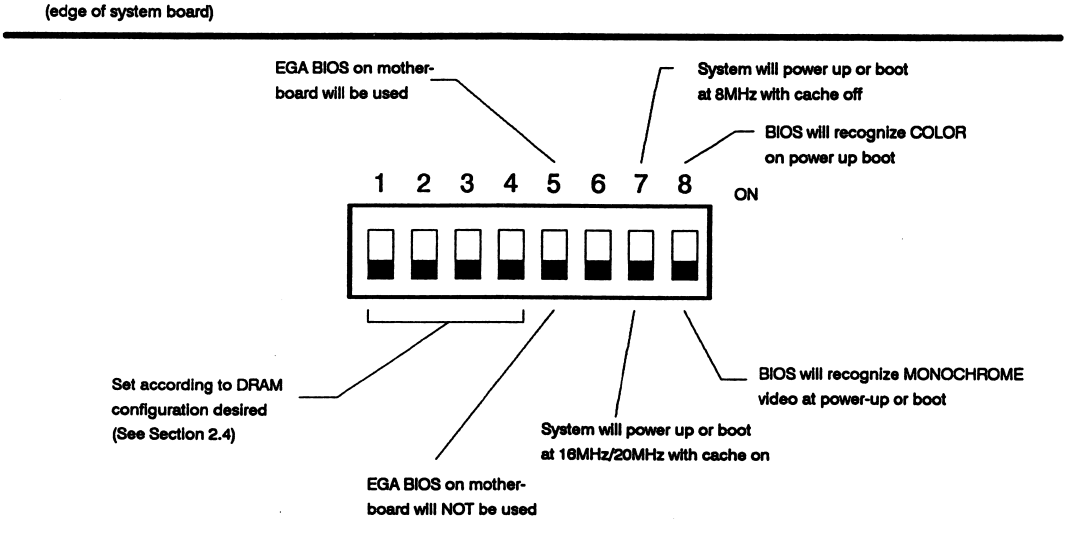
Note: The MI386-XX supports only the 80387 math coprocessor.

Switch 7: ON: System will boot at 16 (or 20)MHz

OFF: System will boot at 8MHz

Switch 8: ON: BIOS will recognize color video on booting

OFF: BIOS will recognize monochrome video on booting



2-3: Switch Settings

2.4 DRAM Switch Settings

The table below indicates how switches 1 through 4 should be set, depending on the amount of conventional memory and extended memory provided by the mother-board. As mentioned in the previous chapter, DRAM can be configured on-board only in 1MB, or 2MB, or 4MB or 8MB sizes. The two numbers shown correspond to the amount of conventional memory (memory below 1MB address) and the amount of extended memory (memory above the 1MB address). Refer to Chapter 4 for more information.

Switches				Mother Board Memory			
4	3	2	1	1MB	2MB	4MB	8MB
OFF	OFF	OFF	OFF	N/A	N/A	640/3072	640/4096
OFF	OFF	OFF	ON	N/A	N/A	512/3072	512/4096
OFF	OFF	ON	OFF	N/A	N/A	640/3072	640/7168
OFF	OFF	ON	ON	N/A	N/A	512/3072	512/7168
OFF	ON	OFF	OFF	N/A	N/A	640/3456	640/3456
OFF	ON	OFF	ON	N/A	N/A	512/3584	512/3584
OFF	ON	ON	OFF	N/A	N/A	640/3072	640/3072
OFF	ON	ON	ON	N/A	N/A	512/3072	512/3072
ON	OFF	OFF	OFF	640/0	640/1408	640/0	640/1408
ON	OFF	OFF	ON	512/0	512/1356	512/0	512/1356
ON	OFF	ON	OFF	640/0	640/1024	640/0	640/1024
ON	OFF	ON	ON	512/0	512/1024	512/0	512/1024
ON	ON	OFF	OFF	640/384	640/384	640/384	640/384
ON	ON	OFF	ON	512/512	512/512	512/512	512/512
ON	ON	ON	OFF	640/0	640/0	640/0	640/0
ON	ON	ON	ON	512/0	512/0	512/0	512/0

SIMM/DIP SIMM pack DRAM should first be loaded into the first memory bank (U5-U8) which corresponds to the lower addresses and then in the second memory bank (U1-U4) which corresponds to the higher addresses.

2.5 Jumper Options and Connectors

After the DIP switches are set correctly, other attachments and jumper option adjustments on the mother-board have to be made. See Figure 2-1 for the location of each of these jumpers and connectors. Most of the jumpers are set appropriately at the factory.

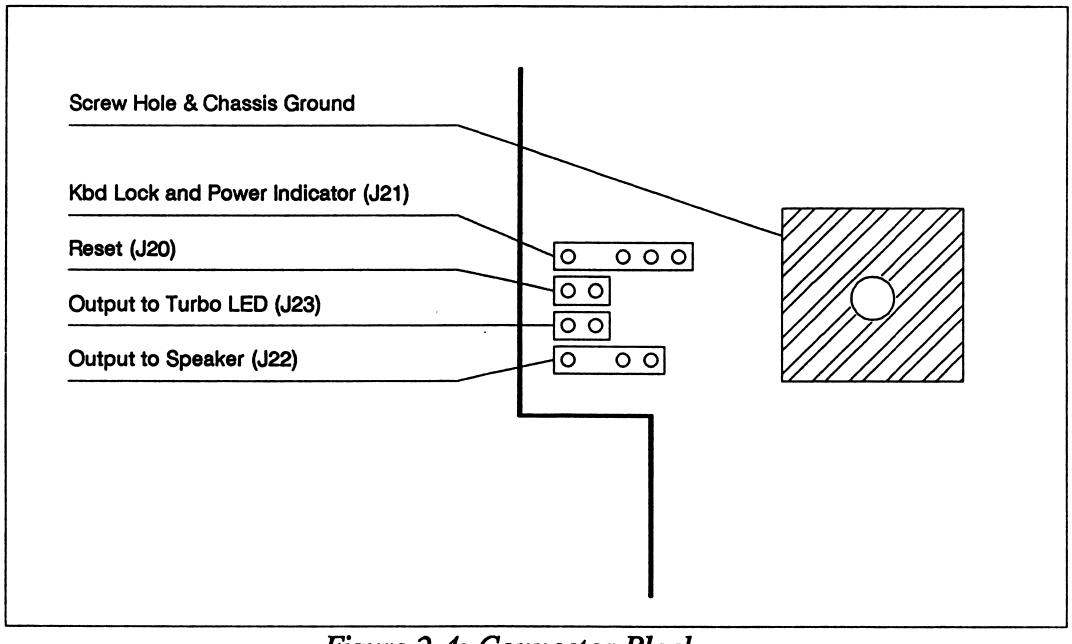


Figure 2-4: Connector Block

J3-J10 (62 pin I/O Connectors):

These are the I/O bus connectors associated with "8-bit" devices. The pin assignments are documented in Chapter 5.

J11 (32 bit Memory Connector):

This socket may be used to increase system address space by installing additional memory.

J12-18 (36 pin I/O Connectors):

These are the I/O bus connectors associated with "16-bit" devices. The pin assignments are documented in Chapter 5.

J19 (3 pin Berg strip):

This is used to select either 256K EPROMs or 128K ROMs. By Default, pins 1 and 2 are jumpered, indicating that 256K EPROMs are installed.

J20 (2 pin Berg strip):

When shorted, a "hard" reset signal is given to the 80386 CPU. This connector will be connected to any reset button provided on the enclosure being used. (See Figure 2-4) Note: IBM AT enclosures do not have a reset button.

J21 (5 pin single-in-line Berg):

Keyboard Lock/Power LED Connector (See Figure 2-4). This connector connects the keyswitch on your enclosure, and also supplies the signal for the power indicator LED.

J22 (4 pin single-in-line Berg):

Speaker Connector (See Figure 2-4). This connector uses only two conductors to the speaker in the enclosure.

J23 (2 pin BERG strip):

Turbo LED Connector (See Figure 2-4). This connector generates the output signal that drives a "Turbo" LED on the front of the enclosure. Note that IBM AT enclosures do not have this indicator.

PS1-2 (Two 6 pin Power Supply connectors):

These connectors are provided for the power supply. It is important that these connectors are correctly connected (See Figure 2-5); if not, the MI386 board may be damaged. In general, most power supplies are wired so as to provide GROUND con-

nections on the **black** wires. These are mounted on the center four pins of the PS1 - PS2 array.

C186 (Variable Capacitor):

This variable capacitor can be used to adjust the 14.31818 MHz oscillator (OSC) signal that is used to obtain the color burst signal required for color television.

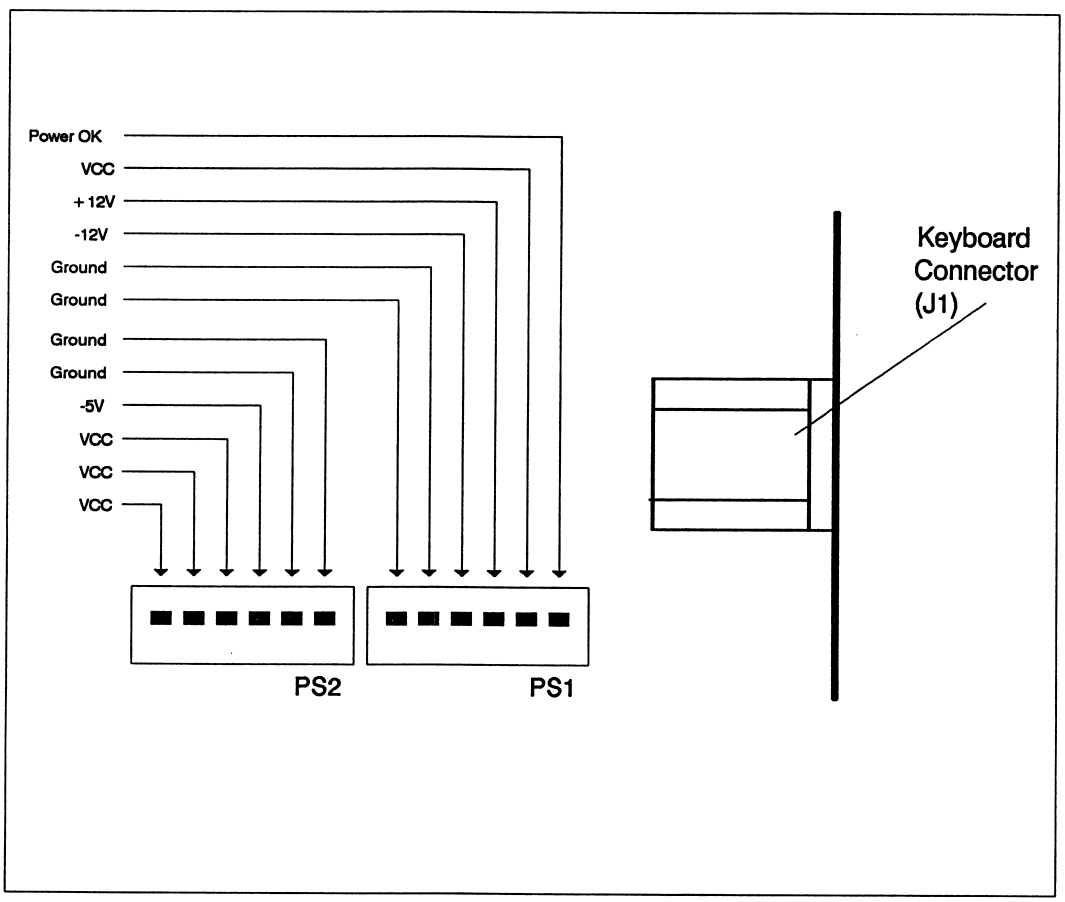


Figure 2-5: Power Connector

2.6 Installing the Intel 80387 Numeric Coprocessor

The 80387 coprocessor should be installed after the system board has been installed. If the MI386-XX system board has not yet been installed, refer to section 2.2. The position of the coprocessor socket is shown in figure 2-1.

All the precautions described in section 2.2 should be followed to avoid damage to the coprocessor due to static electricity.

Also, note that the coprocessor package should be inserted in its socket with the corner closest to pin 1 pointing towards the corner of the board closest to the word "MYLEX" which appears on border of the MI386 board.

During setup, the BIOS will recognize the presence of a math coprocessor on-board, and ask the user to enter the type of coprocessor installed (80287 or 80387). For the MI386 board, only 80387 should be entered, if present. The 80287 math coprocessor cannot be used on the MI386 board. Refer to chapter 3 for details about the SETUP program.

Chapter Three

BIOS: Operation and Setup

This chapter explains the user interface of AMI BIOS, as provided to Mylex by American Megatrends, Inc. Programming AMI BIOS is similar to programming IBM AT BIOS. The program interface to the BIOS is beyond the scope of this manual.

3.1 Introduction to BIOS

The acronym BIOS stands for basic I/O System, and was originally developed for the IBM PC to provide a standard software interface to IBM compatible peripheral devices. The original IBM PC BIOS underwent a number of changes and improvements as it was moved from the PC to the XT, and subsequently to the IBM AT. The AMI BIOS provided with the Mylex MI386 board is closely compatible with the IBM AT BIOS.

In addition to basic I/O services, the BIOS includes the BOOT program, which gains control of the system each time it is powered on or reset. On the IBM AT, the BOOT program will only perform an operational test of the system, then attempt to load a program from either a floppy disk drive or the hard disk drive, if available. On power up or reset, the AMI BIOS performs the functions shown in the flowchart (Figure 3-1).

After the system is powered on, the BIOS conducts a series of functional tests. These are listed below:

- **Video Test:** The presence or absence of a functional video board of the type indicated by Switch 8 on the mother-board is tested (See previous chapter). If absent, the BIOS will attempt to use the alternate type of video board if it detects the presence of one, although the results may be unpredictable. If a video board cannot be found, an error condition results.

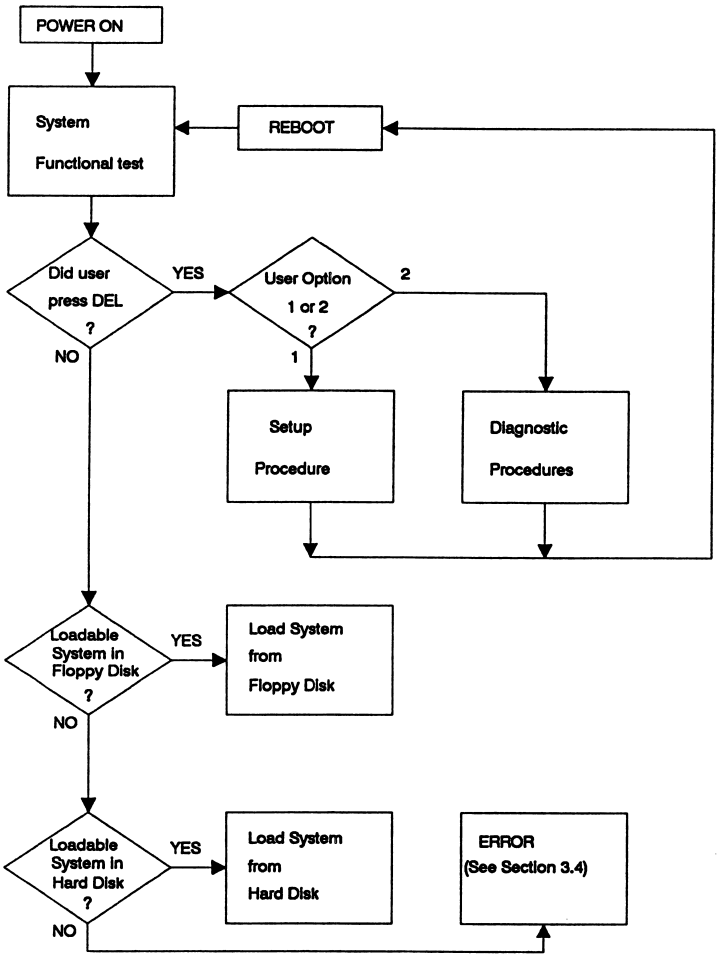


Figure 3-1: BIOS BOOT Flowchart

- **RAM Memory Test:** The amount and functionality of RAM memory is tested. On the video display the amount of RAM memory currently found is displayed as it is tested in 64KB increments. All properly configured memory, including on-board memory and any extended memory cards are included in the count. To bypass this test (in case there is a large amount of memory to test) press the ESC key.

- **Cache Memory Test:** The cache memory is tested and activated if it is functional. The amount of cache memory available is displayed.

- **Disk Drive Test:** Each floppy disk drive and hard disk drive is issued a reset signal and allowed a fixed time to respond.

Because the BIOS is in ROM, the BIOS and BOOT programs are preserved when the computer is powered off. However, this means that in order to change the BIOS the ROM chips will have to be replaced.

3.2 Booting Up

When the system is first powered on or reset, the BIOS will display its title and copyright notice on the top line of the screen, and its current revision number at the bottom of the screen. If the on-board EGA BIOS has been activated with Switch #5 on the system board (see previous chapter), the EGA BIOS copyright and version number will also be displayed. When diagnosing BIOS related problems, it is important to note the BIOS and keyboard BIOS revision number.

An example of the version signature at the bottom of the screen is:

```
0398-110387-K8
```

"110387" is date (11/03/87) of the BIOS release, and "K8" signifies that the Keyboard BIOS revision 8 is being used.

Immediately after this display, the BIOS will perform the functional test of the system described in section 3.1 to make sure that the CPU board, a minimum amount of memory, and a minimum number of peripherals are working. If not, an error condition results (not shown in the flow-chart). This is described in Section 3.4 of this chapter.

If the functional test is passed, the message

```
Press <DEL> key to run SETUP or DIAGS
```

is displayed. The user has a short time interval during which the DEL key can be pressed to obtain access to the in-ROM setup and diagnostic procedures.

If the user presses the DEL key, the screen is cleared and the following message is seen:

```
Entering SYSTEM SETUP due to user request

Want to run SETUP or DIAGS(Y/N)?
```

The user may answer "Y" for yes, or "N" for no. If the answer is "Y", the message:

```
SETUP or DIAGS (1/2)?
```

is displayed. The user may press the "1" key to obtain the SETUP procedure that is documented in the following section, or the "2" key to run the DIAGNOSTICS procedure, described in Section 3.6 of this chapter.

If the DEL key is not pressed within the time limit, the remainder of the flowchart in Figure 3-1 is followed.

3.3 Setup Procedure

In the IBM PC, the predecessor of the IBM AT, the BIOS took the configuration of memory and certain peripherals from option switches on the mother-board and on perhipheral cards that were installed on the system (most notably the hard disk controller card). On the IBM AT, instead of using switches, the same data is stored in a low power RAM device called the CMOS RAM. When the system is powered up or reset, the CMOS RAM is accessed to determine how certain devices are to be operated.

For the system to work properly, the data in the CMOS data base must be set to the correct values. The SETUP procedure is provided for this purpose. If the BOOT program senses that the CMOS data base is not correct, either because it has never been set or because it was compromised, a warning message is displayed.

When the SETUP procedure is started, the operator is prompted for a series of data items. Each response will be entered into the CMOS data base, at the user's option, at the end of the SETUP procedure. The following is a description of the individual questions that the user is asked:

The first question is about the date. The BIOS displays:

```
Current date is ................02-08-1988

Enter new date (MM-DD-YYYY)? ..
```

The current date should be typed in the format shown, with a four digit year. If no entry is made, the date is not changed. SETUP displays next:

```
Current time is.................16:24:29

Enter new time(HH:MM:SS)? .....
```

The current time should be entered in the format shown. If no entry is made, the time is unchanged. SETUP next displays the current video configuration:

```
Primary Display is .........Color display

Current screen width is ....80 columns
```

which is not changable by the user. These values are set according to the position of Switch #8 on the mother board, and the specific video board that is actually installed. SETUP will next ask:

```
Fixed disk drive C type .........
```

and after that:

```
Fixed disk drive D type .........
```

These questions refer to the type of hard disk drives installed. You must select the appropriate drive type according to Table 3-1, based on the number of cylinders, heads, and proper write-precompensation values for both disk drives. You may examine the possible drive types

that are supported by pressing the ESC key. Pressing the RETURN key without making an entry tells the BIOS that a hard disk drive is not installed.

NOTE: If the CMOS data base has already been initialized with hard disk drive type(s) via a prior run of the SETUP program, these questions will be skipped. If you need to change the fixed disk drive types, it will be necessary to complete the following questions until the question "Are these options correct (Y/N)?" (see below) is asked. Answering no to that question will cause SETUP to ask these questions on a second pass.

Disk Drive Type Table

Type	Cyln	Heads	Write Precomp	Landing Zone	Total Capacity
1	306	4	128	305	10 MB
2	615	4	300	615	21 MB
3	615	6	300	615	31 MB
4	940	8	512	940	64 MB
5	940	6	512	940	48 MB
6	615	4	NONE	615	21 MB
7	462	8	256	511	31 MB
8	733	5	NONE	615	31 MB
9	900	15	NONE	901	115 MB
10	820	3	NONE	820	21 MB
11	855	5	NONE	855	36 MB
12	855	7	NONE	855	51 MB
13	306	8	128	319	21 MB
14	733	7	NONE	733	44 MB
15	000	0	000	000	00 MB
16	612	4	ALL	663	21 MB
17	977	5	300	977	42 MB
18	977	7	NONE	977	58 MB
19	1024	7	512	1023	61 MB
20	733	5	300	732	31 MB
21	733	7	300	732	42MB
22	733	5	300	733	31MB

Table 3-1: Disk Drive Type Table

If a low density (360KB) floppy disk is being used, SETUP will automatically initialize the CMOS to the proper values. However, if SETUP senses that a high capacity floppy disk is used, it will then ask:

```
Enter diskette drive A type (1-3)? ...
```

This is because it is not possible for software to determine which of several possibilities is actually used for a high capacity disk drive. The user may enter one of the following answers, which are displayed on the screen:

```
1 - 1.2 MB 5 1/4" diskette

2 - 720KB 3 1/2" diskette
```

Type	Cyln	Heads	Write Precomp	Landing Zone	Total Capacity
23	306	4	ALL	336	10 MB
24	925	7	ALL	925	56 MB
25	925	9	NONE	925	72 MB
26	754	7	754	754	46 MB
27	754	11	NONE	754	72 MB
28	699	7	256	699	42 MB
29	823	10	NONE	823	71 MB
30	918	7	918	918	55 MB
31	1024	11	NONE	1024	98 MB
32	1024	15	NONE	1024	133 MB
33	1024	5	1024	1024	44 MB
34	612	2	128	612	10 MB
35	1024	9	NONE	1024	80 MB
36	1024	8	512	1024	71 MB
37	615	8	128	615	42 MB
38	987	3	987	987	25 MB
39	987	7	987	987	60 MB
40	820	6	820	820	42 MB
41	977	5	977	977	42 MB
42	981	5	981	981	42 MB
43	830	7	512	830	50 MB
44	830	10	NONE	830	72 MB
45	917	15	NONE	918	115 MB
46	000	0	000	000	00 MB

Table 3-1: Disk Drive Type Table (continued)

```
3 - 1.44 MB 3 1/2" diskette
```

Enter the value associated with your drive. Entering the incorrect value will result in improper operation of the drive. This process is repeated for the second floppy disk drive, if one is installed on the system.

After the floppy disk drive configuration has been entered for all drives, SETUP displays the current memory configuration. For example:

```
Base memory size is ........640KB

Expansion memory size is ...3456KB
```

These values cannot be changed by the user. If they appear incorrect for your system (the above example is the default configuration for a 4MB MI386 board), there may be a memory configuration problem with your system.

SETUP next checks for the presence of a math co-processor. If no co-processor is present, the message:

```
Numeric Processor.......Absent

Emulate numeric processor (Y/N)?
```

If desired, "Y" can be entered to invoke the AMI numeric processor emulation code.

If a math co-processor is sensed, SETUP will inqure whether the chip being used is a 80287 or a 80387 as shown below, after the base memory size and expansion memory size (if present) are displayed:

```
Numeric processor..................Present

80287 or 80387 (1/2)?..............
```

In response to the above question the user should enter "2" if a 80387 is installed. Note that the selection "1" cannot be made because the MI386-XX board does not support the 80287 coprocessor.

Beep Count	Meaning
1	DRAM refresh failure
2	Parity Circuit failure
3	Base 64KB RAM failure
4	System Timer failure.
5	Processor Failure
6	Keyboard Controller - Gate A20 error
7	Virtual Mode Exception Error
8	Display Memory R/W Test Failure. (*)
9	ROM-BIOS CheckSum Failure.
(*) Non-Fatal Error.	

Table 3-2: Beep Error Messages

When the SETUP series of questions are complete, the final question:

```
Are these options correct (Y/N)?
```

is displayed. If "N" is entered, the entire series of questions is repeated. If "Y" is entered, the CMOS data base is updated with the entered values and the system automatically resets.

Note that at any time during the execution of the SETUP program, the system can be reset (using the power switch, the reset button, or by pressing CTRL-ALT-DEL on the keyboard) to abort the process without changing the CMOS from its original state.

3.4 Error Messages

One of the functions of the BIOS is to detect certain error conditions during the power-up procedure and while the system is in operation. During power up, if an error occurs before the BIOS has sensed that a video system is available, it will attempt to signal an error with a sequence of tones on the speaker. The number of beeps indicate what kind of problem that was detected by the ROM software. Table 3-2 lists the possible signals.

If an error condition occurs and a video system is available, the BIOS will display the appropriate error message. In such cases, the system will not continue until the operator takes some action. Any error message that occurs during system operation is necessarily fatal to the operation of the program that was running at that time. The possible error messages that may be displayed are:

```
Stand by while system is rebooting.
```

The ROM program is resetting the system. The memory test will be the next display.

```
CLOCK NOT WORKING
```

The BOOT program has detected that the clock is not running. This indicates a problem that needs service by a qualified technician.

```
CMOS DATE/TIME NOT SET
```

The date and time within the CMOS has not been set. Run the SETUP program.

```
Channel-2 timer not functional
```

A hardware problem has been detected. The board should be examined by a qualified technician.

```
Keyboard error
```

A scan code that was unexpected was detected on the keyboard. This is most often generated when one or more keys are pressed when the BOOT program is testing the keyboard. Make sure that keys are not stuck or depressed on the keyboard during the boot-up phase. If no problem is detected, run the DIAGNOSTICS routine for the keyboard (see next section).

```
Keyboard/Interface error
```

The keyboard controller is having difficulty communicating with the keyboard. Check for improper connection. If the problem persists, run the DIAGNOSTICS routine (see next section) for the keyboard.

```
Stray interrupt sensed in controller #1

Interrupt controller #2 not functional
```

These messages indicate that some hardware problem is being sensed on the I/O interrupt lines. This problem may be caused by one of the controller cards on the I/O bus.

```
CMOS battery state low
```

The CMOS battery voltage has dropped below the point where the CMOS memory will be able to reliably store data. The "Dallas" chip may have to be replaced.

```
CMOS system options not set
```

The BOOT program has sensed that certain CMOS options have not been set. Run the SETUP procedure.

```
CMOS checksum failure
```

The BOOT program has sensed that the CMOS setup data has been changed in an improper fashion. This message will always appear when the board is first installed, or when the board is powered up after it has been disconnected from the battery.

```
CMOS memory size mismatch
```

The amount of memory (DRAM) that has been sensed in the initial memory test is different from the amount of memory SETUP has recorded in the CMOS data. This message will appear each time a new memory board is installed or removed from the system. The SETUP procedure must be run, even though SETUP does not ask the user any questions regarding the memory setup.

```
CMOS display configuration mismatch

Display switch setting not proper
```

The switch setting on the MI386 board does not match the recorded value for the type of video system being used. Switch #8 (see Chapter 2) must be set appropriately. SETUP may have to be run again to change the CMOS data.

```
Keyboard is locked ..... Unlock it
```

The lock switch is set to the lock position. If the system is not installed with a lock switch, the pins on the connector may be shorted.

```
CACHE MEMORY BAD, DO NOT ENABLE CACHE !
```

The BOOT program has sensed errors in the cache memory. This problem will have to be serviced by a qualified technician.

```
Floppy disk controller failure
```

The BOOT program was unable to communicate with the flopply disk drive controller board.

```
CMOS floppy drives/type mismatch
```

The BOOT program has sensed that the types of drives actually installed on the system are different from those drive types recorded in the CMOS data base. Run the SETUP program to load the CMOS with the correct values.

```
Hard disk controller failure
```

The hard disk controller did not respond to commands.

```
Hard disk unit 0 error

Hard disk unit 1 error
```

The hard disk controller has reported that the data or activity of the hard disk drive is unreliable. This may be caused by setting the wrong drive type in the CMOS via the SETUP utility.

```
Hard disk unit 0 failure

Hard disk unit 1 failure
```

The hard disk drive indicated failed to respond. This message will occur when the CMOS data base indicates the presence of a hard disk drive that is not actually installed or is installed incorrectly.

```
Hard disk unit 0 TYPE undefined in CMOS

Hard disk unit 1 TYPE undefined in CMOS
```

The CMOS was loaded with a drive type for the indicated drive that is not defined. (These are usually types 15 and 47). Run SETUP to load the correct drive type.

```
*** RUN SETUP UTILITY ***
```

The BOOT program is advising the user that the SETUP program needs to be run.

```
Press <F1> key to RESUME
```

A message has been put on the screen that requires acknowledgement from the user. Press the [F1] key to continue operations. If the message has indicated an error that the system could not proceed without running SETUP, the SETUP/DIAG message will be displayed.

```
CMOS INOPERATIONAL
```

The system is unable to alter the state of the CMOS data base. This condition requires service from a qualified technician.

```
8042 GATE A20 ERROR
```

A problem with the keyboard controller has been detected. This condition requires service from a qualified technician.

```
INVALID SWITCH SETTING/MEMORY FAILURE
```

The switch settings for Switches #1-4 on the MI386 board have been set in such a way that the memory system is inoperable. This message can also be generated from an incorrectly configured extended memory board. Review the switch setting in the installation guide and reset as appropriate.

```
DMA UNIT #1 ERROR

DMA UNIT #2 ERROR

DMA CONTROLLER ERROR

INTERRUPT CONTROLLER #1 ERROR
```

An error condition has been sensed in the DMA and Interrupt Controller logic. This condition requires service from a qualified technician.

```
SYSTEM HALTED
```

The system is unable to continue. The BIOS has locked the system from further operation. If further operation is desired, the system may be restarted by a RESET signal or by a power down/up cycle only.

```
SYSTEM DOES NOT SUPPORT RESIDENT BASIC
```

The BOOT program did not find a bootable floppy disk or hard disk drive that was installed. On IBM AT computers, the system would normally invoke ROM-BASIC at this point. However, AMI BIOS does not include ROM (Resident) BASIC. A bootable medium must be provided to the system.

```
DISKETTE BOOT FAILURE

INVALID BOOT DISKETTE
```

The BOOT program was unable to boot the floppy disk drive, or the diskette that was in the drive was not bootable. Correct the problem and reset the system.

```
DRIVE NOT READY ERROR
```

The floppy or hard disk drive did not indicate a ready status within the expected time frame. This may indicate a problem with the disk drive.

3.5 Keyboard Command Codes

One of the services that the BIOS provides to software is the keyboard interrupt and handling routines. As long as the interrupt vectors that the keyboard controller uses are not changed or altered by software, certain key combinations on the keyboard will invoke special functions. The key combinations are acutated by pressing the **[CTRL]** and **[ALT]** keys simultaneously, holding them, and then pressing the third key in the combination. After all keys have been pressed, the keys can be released.

The valid key combinations are:

[CTRL] [ALT] [DEL]

Reboot the computer. Any activity currently happening will be halted, and the system will re-initialize. The memory count routine will be skipped.

[CTRL][ALT] [-]

Low speed mode. Note that the [-] (minus key) must be the key on the numeric keypad; not the one in the letters section. On receiving this command, the MI386-XX will change its operation to 8MHz. Note that when operating in low-speed mode, the cache is automatically disabled.

[CTRL] [ALT] [L-SHIFT] [-]

Turn off cache. Note that the **[L-SHIFT]** refers to the left hand shift key, and the [-] (minus) key must be the key on the numeric keypad; not the one near the letters on the keyboard. This combination takes effect only while in high speed mode (16MHz or 20MHz).

[CTRL] [ALT] [+]

Change to high-speed mode with cache enabled. Note that the [+] (plus sign) key must be the key on the numeric keypad; not the one near the letters on the keyboard. This combination will initiate high speed (16MHz or 20MHz, depending on which board is used) operation.

[ALT] [digits 0-9]

Enter special input code. Each ASCII character is assigned a special decimal number which can be found in an ASCII table. For example, the letter **C** (captial C) corresponds to decimal code 67. This letter can be entered by pressing and holding the **[ALT]** key, entering the digits **67** on the numeric keypad, and then releasing the **[ALT]** key. This feature is useful for entering graphic symbols and special codes into programs.

3.6 Diagnostic Routines

The AMI BIOS includes a full set of diagnostic routines which can be used to examine problems that may occur with any peripheral device on the system. They are activated by pressing the **[DEL]** key during the boot-up phase as described at the beginning of this chapter.

When started, the diagnostic routines present the user with a "bar type" menu, where options can be selected by using the arrow keys (left and right), and activated by pressing the **[ENTER]** key. Each option will activate a sub-menu which will then appear.

For the most part, the diagnostics are self-explanatory. However, there are a number of operational notes to consider:

- The disk diagnostic routines will accept an arbitrary set of numbers for the dimensions of the drive being tested. Note that this does not mean that the drive is supported by the drive-type table referenced by the SETUP routine.
- The low-level format routines for hard disk are not recommended for drive interface types other than MFM.
- The floppy disk format for the 1.44MB disk drive (3 1/2 inch) is not currently supported.
- The floppy disk format performs a low-level format only. No DOS or other operating system directory structure is written.
- The keyboard test does not sense the presence of the **[F11]** and **[F12]** keys.
- The communications test will require a special loop-back plug, which is described by the program when the test is initiated.

Chapter Four

Memory System

4.1 Expanded and Extended Memory

In the IBM PC-AT, conventional memory or base memory extends from 0 to 640 KB. This is the user area, and is available for use by application software. Physical memory address space from 640KB to 1MB is reserved for the system. DOS recognizes and knows how to use the memory area from 0 to 1MB only. Refer to figure 4-1 for the memory map.

One way of overcoming the 640KB limitation on conventional (user) memory is by using expanded memory. This requires the use of additional bank-switched physical memory (memory that is organized in banks which can individually be switched on or off) along with Lotus/Intel/Microsoft Expanded Memory Specification (EMS) compatible Expanded Memory Manager (EMM) software and an application program that is capable of working with the EMM software. The EMM manager first finds a 64KB page frame in the unused part of the system area, divides the frame into 4 16KB windows and swaps in 4 16KB pages from different areas of the additional physical memory. This additional paged memory that is used along with an EMS emulator is known as *expanded memory*.

There are application programs that are capable (although DOS is not) of making use of the physical memory beyond 1MB. This additional memory is referred to as *extended memory*. Extended memory can be used by the PC-AT only when it is operating in the protected mode. IBMs VDISK and the XENIX™ operating system use extended memory.

The Mylex MI386 mother board can have upto 8MB of DRAM, of which 7MB is extended memory. Additional extended memory (upto 8MB) can be installed using the 32-bit memory connector. The on-board memory is available in 1MB, 2MB, 4MB and 8MB sizes as mentioned earlier in chapters 1 and 2. The on-board memory can be con-

figured in different ways, depending on the amount of conventional memory and extended memory desired. Switches 1 through 4 on the mother board should be set as described in Section 2.3.

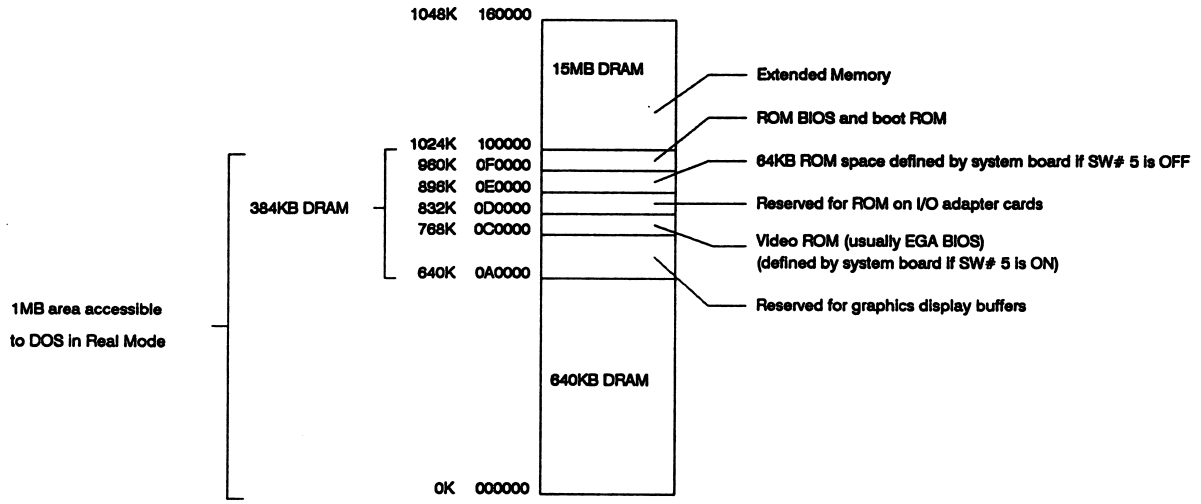


Figure 4-1: Memory Map

4.2 System Memory Map

The system memory map is as follows:

Address	Name	Function
000000 to 07FFFF	640KB system board	System board memory
0A0000 to 0BFFFF	128KB video RAM	Reserved for graphics display buffer
0C0000 to 0CFFFF	64KB of ROM	Switch 5 OFF: EGA BIOS at 0E0000 Switch 5 ON: EGA BIOS at 0C0000
0D0000 to 0DFFFF	64KB I/O expansion ROM	Reserved for ROM on I/O adapter cards
0E0000 to 0EFFFF	64KB ROM available on system board if Switch 5 is OFF	Duplicate code assignment from address FE0000 to FEFFFF
0F0000 0FFFFF	64KB ROM on system board	Duplicate code assignment from address FF0000 to FFFFFF
From end of system board memory to FDFFFF		I/O channel memory
FE0000 to FEFFFF	64KB reserved on system board	Duplicate code assignment from address 0E0000 to 0EFFFF
FF0000 FFFFFF	64KB ROM on system board	Duplicate code assignment from address 0F0000 to 0FFFFF

4.3 Cache and Processor Speed Programming

This section describes how the cache can be enabled and how the processor speed can be switched from 8MHz to 20MHz or 16MHz and vice-versa.

The cache memory is enanbled by setting bit 0 on write-port hex 460. It is disabled by resetting the same bit. On powering on, the cache is disabled. If the cache is reported to be defective after the Power On Self Test (POST), it should not be enabled.

The processor clock frequency is determined by bit 0 of read/write port hex 461. Bit 0 is set for 20MHz (or 16MHz) operation, and reset for 8MHz operation.

The following C language program sets data bit 0 on port hex 461, thereby switching the processor to the slower clock speed.

```
#include <conio.h>

unsigned port=0x461;

int value=0;

main()

{

 outp(port,value);

}
```

4.4 Cache and DRAM Access

In 80386, a zero wait-state bus cycle requires two clock cycles. The required number of clock cycles increases by one with every wait-state introduced during memory or I/O access. Most processor accesses are memory read operations. To speed up memory read operations, the MI386 incorporates 64KB of 32-bit wide directly mapped cache, implemented with high-speed Static RAM. The cache has byte granularity and page size of 4 bytes, and is implemented using a write-through

algorithm. With the cache enabled, 81% of data will be available in the cache during memory read operations. (This is the statistical value for normal program execution.) This improves system speed approximately by a factor of two since cache access requires no wait states, whereas the main memory access would require at least two wait states.

The cache RAM architecture operates on a 16MB physical memory address range. Off board memory (from an extended memory card) is also cached, resulting in low wait state performance even with AT compatible extended memory boards.

4.5 Data Access: Cycle Times

The following table gives the number of wait states and the total bus cycle times for different cases of cache and on-board DRAM access:

Access Type	16MHz Wait/ns	20MHz Wait/ns
RAM Read: Cache hit	0/125	0/100
DRAM Read: Cache miss	3/312.5	3/250
DRAM Read: Cache disable	2/250	2/200
DRAM Write	2/250	2/200
16-bit I/O device: MEMW/IORD	4/375	4/300
16-bit I/O device: MEMR/IOWR	5/437.5	5/350
8-bit device MEMW/IORD	10/750	10/600
8-bit device MEMR/IOWR	11/812.5	11/650
16-bit to 8-bit: MEMW/IORD	22/1500	22/1200
16-bit to 8-bit: MEMR/IOWR	24/1625	24/1300
16-bit to 8-bit: MEMR/IOWR	24/1625	24/1300

4.6 Refresh Controller

The refresh controller operates at 6MHz. Each refresh cycle requires 5 clock cycles to refresh all the DRAM in the system. 256 refresh cycles are required every 4 ms.

Chapter Five

I/O Channel

5.1 Introduction

This chapter describes the I/O channel; it lists the pin assignments for the I/O channel connectors, describes each I/O channel signal line and gives the I/O address map. The I/O channel has the following features:

- I/O address space from hex 100 to hex 3FF
- 16MB memory address space
- Selection of data accesses (either 8-bit or 16-bit)
- 11 levels of interrupt
- 7 DMA channels
- I/O wait state generation
- Open bus structure
- Refresh of system memory from the system microprocessors
- I/O Channel connectors

There are eight 62-pin and six 36-pin edge connector sockets. In one position the 36-pin connector socket is absent and in the other position there is a 32-bit memory connector. In these positions, only 62-pin connectors can be connected.

The following tables show the pin assignments of the I/O channel connectors.

I/O Pin	Signal Name	I/O
A1	-I/O CH CK	I
A2	SD7	I/O
A3	SD6	I/O
A4	SD5	I/O
A5	SD4	I/O
A6	SD3	I/O
A7	SD2	I/O
A8	SD1	I/O
A9	SD0	I/O
A10	-I/O CH RDY	I
A11	AEN	O
A12	SA19	I/O
A13	SA18	I/O
A14	SA17	I/O
A15	SA16	I/O
A16	SA15	I/O
A17	SA14	I/O
A18	SA13	I/O
A19	SA12	I/O
A20	SA11	I/O
A21	SA10	I/O
A22	SA9	I/O
A23	SA8	I/O
A24	SA7	I/O
A25	SA6	I/O
A26	SA5	I/O
A27	SA4	I/O
A28	SA3	I/O
A29	SA2	I/O
A30	SA1	I/O
A31	SA0	I/O

I/O Channel Pin Assignments (A-Side, J1 through J8)

I/O Pin	Signal Name	I/O
B1	GND	Ground
B2	RESET DRV	O
B3	+5Vdc	Power
B4	IRQ9	I
B5	-5VDC	Power
B6	DRQ2	I
B7	-12Vdc	Power
B8	OWS	I
B9	+12VDC	Power
B10	GND	I
B11	-SMEMW	O
B12	-SMEMR	O
B13	-IOW	I/O
B14	-IOR	I/O
B15	-DACK3	O
B16	DRQ3	I
B17	-DACK1	O
B18	DRQ1	I
B19	-RefreshQ	I/O
B20	CLK	O
B21	IRQ7	I
B22	IRQ6	I
B23	IRQ5	I
B24	IRQ4	I
B25	IRQ3	I
B26	-DACK2	O
B27	T/C	O
B28	BALE	O
B29	+5Vdc	Power
B30	OSC	O
B31	GND	Ground

I/O Channel Pin Assignments (B-Side, J1 through J8)

I/O Pin	Signal Name	I/O
C1	SBHE	I/O
C2	LA23	I/O
C3	LA22	I/O
C4	LA21	I/O
C5	LA20	I/O
C6	LA19	I/O
C7	LA18	I/O
C8	LA17	I/O
C9	-MEMR	I/O
C10	-MEMW	I/O
C11	SD08	I/O
C12	SD09	I/O
C13	SD10	I/O
C14	SD11	I/O
C15	SD12	I/O
C16	SD13	I/O
C17	SD14	I/O
C18	SD15	I/O

I/O Channel Pin Assignment (C-Side; J12 through J18)

I/O Pin	Signal Name	I/O
D1	-MEM CS16	I
D2	-I/O CS16	I
D3	IRQ10	I
D4	IRQ11	I
D5	IRQ12	I
D6	IRQ13	I
D7	IRQ14	I
D8	-DACK0	O
D9	DRQ0	I
D10	-DACK5	O
D11	DRQ5	I
D12	-DACK6	O
D13	DRQ6	I
D14	-DACK7	O
D15	DRQ7	I
D16	+5Vdc	Power
D17	-MASTER	I
D18	GND	Ground

I/O Channel Pin Assignment (D-Side; J12 through J18)

5.2 Description of I/O Channel Signals

In this section, the various I/O channel signals on the MI386 system board are described. All signal lines are TTL-compatible.

SA0-SA19(I/O)
The 20 address lines SA0 through SA19 are used to address the system memory and I/O devices. In conjunction with the signal lines LA17 through LA23, these address lines can address upto 16MB of memory. The signals on these lines are generated by the 80386, DMA Controller, or other bus masters that reside on the I/O channel. The signals SA0 through SA19 are gated on the system bus when 'BALE' is high and are latched on the falling edge of 'BALE'.

LA17-LA23(I/O)
As mentioned above, these signals are used to address memory and I/O devices in the system. These signals are driven by microprocessors or DMA Controllers on the channel. Also, these signals are valid when 'BALE' is high, but are not latched during the microprocessor cycle. These signals serve the purpose of generating memory decodes for 1 wait-state memory cycles. These decodes should be latched by I/O adapters on the falling edge of 'BALE'.

BALE(0)
This '(Buffered) Address Latch Enable' signal is generated by the 82288 Bus Controller and is used by the MI386 to latch valid addresses and memory decodes. It is used to indicate the presence of a valid microprocessor address to the I/O channel, or that the DMA address is valid, when used along with the 'AEN' signal. The signals on the address lines SA0 through SA19 are latched with the falling edge of 'BALE'. During DMA operations 'BALE' is held high.

SD0-SD15(I/O)
These are the data bus signals (SD0 is least significant and SD15 is most significant) for the processor, memory and I/O devices on the I/O adapter cards. Only D0 through D7 are used for 8-bit devices on the I/O channel. 16-bit data transfers to 8-bit devices are broken up into two 8-bit transfers by gating D8 through D15 to D0 through D7. These lines are active high.

SYSCLK(0)
For a MI386 board running at a processor clock speed of 20 MHz, this signal has a frequency of 10 MHz, and for a processor speed of 16MHz, the corresponding SYSCLK speed is 8MHz. At the corresponding lower speeds (10MHz and 8MHz respectively), the SYSCLK signal runs at the same speed as the processor. This signal has a 50% duty cycle and is synchronous with the processor clock. The SYSCLK signal should be used for synchronization only, and not for obtaining a fixed fiequency.

-I/O CH CK (I)
This '-I/O Channel Check' signal is used indicate the occurence of a parity error in the memory or in the I/O channel. An active '-I/O CH CK' signal indicates a fatal error.

I/O CH RDY (I)
This (I/O Channel Ready) signal is used by slow memory or I/O devices to lengthen I/O or memory cycles. The device drives this signal low soon after detecting its valid address, and and a Read or Write command. Machine cycles are extended by an integral number of clock cycles. This signal should be held low for no longer than 2.5 microseconds.

IORQ3-IORQ7, IORQ9-IORQ12, IORQ14-IORQ15 (I)
These Interrupt Request signals are used by I/O devices to signal to the microprocessor that they need to be serviced. Each signal has its own level of priority. IRQ9 has the highest priority and IRQ7 the lowest. An interrupt request signal is generated when one of these lines is raised. The signal should be held high until it is acknowledged by the microprocessor.

-IOR (I/O)
This 'I/O Read' signal is used by the system microprocessor or DMA Controller resident on the I/O Channel to instruct an I/O device to put its data on the data bus. It is active low.

-IOW (I/O)
The 'I/O Write' signal is used to instruct an I/O device to read the data on the data bus. This active low signal is driven by the microprocessor or the DMA Controller resident on the I/O channel.

-MEMR (O), -SMEMR (I/O)
These signals instruct memory devices to drive data onto the data bus. MEMR is active on all memory read cycles, and is driven by the microprocessor or the DMA Controller. Before driving '-MEMR' low, the microprocessor should have the address lines on the address bus valid for at least one system clock cycle.

'-SMEMR' is derived from '-MEMR' and the decode of the low 1MB of the system memory. That is, '-SMEMR' is active (low) when the decoded address lies within the lowest 1MB of the address space.

-MEMW (O), -SMEMW (O)
These signals instruct memory devices to store the data on the data bus. 'MEMW' is driven by a microprocessor or DMA Controller in the system, and is active during every memory write cycle. The address on the address bus should be held low for at least 1 system clock cycle before -MEMW' is driven active (low).

'SMEMW' is derived from '-MEMW' and the decode of the low 1MB of the system memory. This signal is active (low) when the decoded address lies within the lowest 1MB of the address space.

DRQ0-DRQ3, DRQ5-DRQ7 (I)
DMA Requests 0 through 3 and 5 through 7 are asynchronous channel requests used by peripheral devices (and microprocessors) on the I/O channel to obtain DMA service (or gain control). 'DRQ0' has the highest priority and 'DRQ7' has the lowest. To request DMA, one of the DRQ lines should be made active and held active until the corresponding 'DMA Request Acknowledge' (DACK) signal becomes active. DRQ0 through DRQ3 are used for 8-bit transfers and DRQ5 through DRQ7 are used for 16-bit DMA transfers. 'DRQ4' is used on the system board, and is not available on the I/O channel.

-DACK0 to -DACK3 , -DACK5 to -DACK 7 (O)
-DMA Acknowledge 0 to 3 and 5 to 7 are used to acknowledge DMA requests (DRQ0 to DRQ7). They are active low.

AEN (O)
The 'Address Enable' signal is used to transfer control to the DMA Controller by degating the microprocessor and other devices from the I/O channel. When this line is raised, the DMA Controller has control of the address bus, the data bus and the (Memory and I/O) Read and Write command lines.

-REFRESH (I/O)
This signal is made active by a microprocessor on the I/O channel during every memory refresh cycle.

T/C (O)
The 'Terminal Count' signal provides a pulse when the terminal count of any DMA channel is reached.

SBHE (I/O)
The '(System) Bus High Enable' signal is active during the transfer of data on the upper byte of the data bus (SD8-SD15), and is used by 16-bit devices to condition the data bus buffers connected to SD8 through SD15.

-MASTER (I)
This signal is used along with a DRQ signal to take over control of the system buses for DMA operations. A processor or DMA controller on the I/O channel may issue a DRQ to a DMA channel in cascade mode and receive a '-DACK'. On receiving an active (low) '-DACK' signal, the DMA Controller (or processor) can make -MASTER active (low) and gain control of the system buses. The DMA Controller should wait for at least one clock period (after -MASTER is pulled low) before putting data or addresses on the respective buses, and for at least two clock cycles before driving the control bus lines (read and write). If this line is held low for more than 15 microseconds, system memory may be lost because of lack of refresh.

-MEM CS16 (I)
'-MEM Chip Select' goes active if the current data transfer is a 16-bit, 1 wait-state I/O cycle. It is obtained from the decode of LA17 through LA23. '-MEM CS 16' should be driven with an open collector or tri-state driver capable of sinking 20mA.

-0WS (I)
The 'Zero Wait State' signal indicates to the microprocessor that the current bus cycle can be completed without inserting any additional wait cycles. In the case of a memory cycle with no wait states, involving a 16-bit device, '0WS' is derived from an address decode gated with a Read or Write signal. To run a memory cycle to a 8-bit device with a minimum of two wait states, the '0WS' signal should be made active one system clock after the Read or Write command is active, gated with the address decode of the device.

OSC (O)
The 'Oscillator' is a high frequency clock with a period of 70 nanoseconds. It has a 50% duty cycle, and is not synchronous with the system clock.

RESET DRV (O)
The 'Reset Drive' signal is used to reset the system hardware when it is powered-up.

5.3 I/O Address Map

The I/O address map of the MI386 is given below. Note that the I/O addresses from hex 000 to hex 0FF are reserved for the system board I/O, and the addresses from hex 100 to hex 3FF are availiable on the I/O channel.

Address Range	Device
000-01F	DMA Controller 1, 8237A-5
020-03F	Master Interrupt Controller, 8259A
040-05F	Timer, 8254-2
060-06F	Keyboard Controller (8742)
070-07F	Real-time clock, NMI (non-maskable interrupt) mask
080-09F	DMA Page Register, 74LS612
0A0-0BF	Interrupt Controller 2, 8259A
0C0-0DF	DMA Controller 2, 8237A-5
0F0	Clear Math Coprocessor Busy
0F1	Reset Math Coprocessor
0F8-0FF	Math Coprocessor
460-46F	Reserved for cache memory and system clock switching

Address Range	Device
1F0-1F8	Fixed Disk
200-207	Game I/O
278-27F	Parallel printer port 2
2F8-2FF	Serial port 2
300-31F	Prototype card
360-36F	Reserved
378-37F	Parallel printer port 1
380-38F	SLDC, 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

Chapter Six

Keyboard Controller

6.1 Introduction

The keyboard controller is based on the 8742 single-chip microcomputer and is used to support the MI386 keyboard interface. The keyboard controller performs the following functions:

- Receives serial data from the keyboard, checks parity, and translates it into a system scan code, if necessary. Transfers data to the data buffer and interrupts the processor.
- Executes system commands, places the results in the data buffer and interrupts the processor if necessary.
- Transmits system data in the data buffer to the keyboard in serial format along with the parity bit. Reports the response of the keyboard to the system.
- Reports errors to the system through the status register.

6.1.2 Receiving Data from the Keyboard

The keyboard uses a 11-bit frame to send serial data. The frame comprises:

- 1 start bit
- 8 data bits
- 1 odd parity bit
- 1 stop bit

The data sent is synchronized with a clock in the keyboard. The keyboard controller places the received byte of data in its receive-data buffer and disables the keyboard-interface until the data is read by the processor. In case the data byte received is detected to have a parity

error, the controller requests the keyboard to resend the byte. If the controller still does not receive error-free data, it places hex FF in the output buffer of the keyboard controller and sets the parity bit in the status register to 1, to indicate a receive parity error. The keyboard controller also times the transmission of data from the keyboard. If data transmission does not end in two milliseconds, hex FF is placed in the output buffer of the keyboard controller and the receive time-out bit in the status register is set.

6.1.3 Scan Code Translation

The keyboard scan code received from the keyboard as described above is converted into a system scan code. The system scan code is then placed in the output buffer of the keyboard controller.

6.1.4 Sending Data to the Keyboard

Data is sent to the keyboard using the same 11-frame serial format. If the keyboard does not begin reading in data within 15 milliseconds, or the duration of transmission exceeds 2 milliseconds, hex FE is placed in the output buffer of the keyboard, and the transmit time-out error bit is set in the status register.

The keyboard has to respond to every transmission from the controller. If a parity error is detected in the keyboard response, the controller places hex FE in its output buffer, and the transmit time out and parity error bits are set in the status register. In addition, the controller times the response time of the keyboard. If it exceeds 25 milliseconds, the transmit and receive time out error bits are set in the status register, and hex FE is placed in the output buffer.

6.1.5 Keyboard Inhibit

The keyboard interface may be inhibited by a keyboard-controlled hardware switch. However, transmission from the keyboard controller is allowed. Every transmission from the keyboard is tested to determine if it contains data or is the keyboard response to a command from the controller. If it is a scan code, it is ignored; otherwise, it is placed in the output buffer.

6.2 Keyboard Controller System Interface

The system communicates with the keyboard controller through the input buffer, output buffer and the status register. The status register is 8 bits wide and read-only, and may be read at any time to determine the state of the keyboard controller and the interface. The output buffer can be read through I/O port hex 60. The input port can be written into through I/O port hex 64 and hex 60. When the input buffer is written to through I/O port hex 64, it is interpreted as a command, and if it is written to through hex 60, it is taken to be a parameter of a command to the controller, or as data to be transmitted to the keyboard.

6.2.1 Status Register Bit Definitions

Bit 0 **Output Buffer Full**
This bit is set when the output buffer is full and contains data for the system to read. This bit is reset after the output buffer is read by the system. If this bit is 0, the output buffer is empty.

Bit 1 **Input Buffer Full**
A 1 indicates that the keyboard controller's input buffer is full, and that it has not yet been read by the controller. It is reset when the controller reads the input buffer.

Bit 2 **System Flag**
This bit can be set to 1 or 0 depending on the command from the system. It is set to 0 after a power on reset.

Bit 3 **Command/Data**
This bit is used by the keyboard controller to interpret whether the the input buffer contains a command or data. When the input port is written to using the I/O address hex 64, this bit is set to 1, and the contents of the input buffer are interpreted as a command. When address hex 60 is used to

write to the input buffer, this bit is reset to 0 and the contents of the input buffer are interpreted as data.

Bit 4 **Inhibit Switch**
This bit indicates whether the keyboard is inhibited or not. It is updated whenever the output buffer of the controller is written into. A 0 indicates that the keyboard is inhibited.

Bit 5 **Transmit Time-Out**
A 1 indicates that transmission from the keyboard controller to the keyboard was not completed properly in the predefined time limit. If the transmitted byte was not clocked out in the predefined time limit, this will be the only error bit set. If the transmit byte was sent out within the time limit, but a response was not received in the time limit, the transmit time-out and receive time-out error bits are set (to 1). If the received response contained a parity error, but the transmit byte was clocked out successfully, this bit and the parity error bits are set (to 1).

Bit 6 **Receive Time-Out**
A 1 indicates that transmission from the keyboard to the keyboard controller was not completed properly 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 transmits with odd parity.

6.2.2 Keyboard Controller Commands

As mentioned earlier, keyboard controller commands can only be entered using I/O address hex 64. Writing to this address sets a flag, indicating that the input buffer contains a command for the keyboard controller. Each keyboard controller command is described below.

20 **Read Keyboard Controller's Command Byte**
Command for the controller to send its current command byte to its output buffer

60 **Write Keyboard Controller's Command Byte**
Command to place the next byte of data written to I/O address hex 60 in the controller's command byte.

Each bit is described below:

Bit 7 **Not used. Should be 0.**

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. The two-byte break sequence is converted into a 1-byte IBM PC break code.

Bit 5 **Not Used.Should be 0.**

Bit 4 **Disable Keyboard**
When this bit is set to 1, the keyboard interface is disabled by driving the 'clock' line low. Data is not sent or received.

Bit 3 **Inhibit Override**
The keyboard inhibit function is disabled when this bit is a 1.

Bit 2 **System Flag**
The value of this bit is written into the system flag bit (bit 2) of the status register.

Bit 1 **Not Used**

Bit 0 **Enable Output Buffer Full Interrupt**
When this bit is 1, the controller generates an interrupt when it places data in its output buffer.

AA **Self Test**

This command causes the controller to perform internal diagnostic tests. If no errors are detected, a hex 55 is placed in the output buffer.

AB **Interface Test**

This instructs the controller to test the keyboard clock and data lines. The test result may be learnt by examining the contents of the output buffer:

00	No error detected.
01	The 'keyboard clock' line is stuck low.
02	The 'keyboard clock' line is stuck high.
03	The 'keyboard data' line is stuck low.
04	The 'keyboard data' line is stuck high.

AC **Diagnostic Dump**

This commands the controller to send 16 bytes of the controller's RAM, the current state of the input port, the current state of the output port, and the controller's program status word to the system.

AD **Disable Keyboard Interface**

This command disables the keyboard interface by setting bit 4 of the controller's command byte and driving the clock line low. Data will not be sent or received.

AE **Enable Keyboard Feature**

This command clears bit 4 of the command byte, thereby enabling the keyboard.

CO **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 causes the next byte of data written to I/O port hex 60 to be placed in the controller's output port. Note that bit 0 of the output port should not be written 0 because it is connected to System Reset.

E0 **Read Test Inputs**
This commands the controller to read its T0 and T1 inputs and place the data in the output buffer. Data bit 0 corresponds to T0 and data bit 1 represents T1.

F0-FF **Pulse Output Port**
Bits 0 through 3 of the controller's output port may be pulsed low for approximately 6 microseconds. Bits 0 to 3 indicate which bits are to be pulsed. A 0 indicates that the bit should not be modified.

6.3 I/O Ports

The keyboard controller has two 8-bit ports. One of them serves as an input port, while the other serves as the output port. Besides these, there are two test inputs. One of these, namely T1, is used to monitor the state of the 'clock' line, while the other, namely T2, is used to read the state of the keyboard's 'data' line. Bit definitions for the input, output and test input ports are given below:

Input Port Bit Definitions

Bit 0 **Undefined**

Bit 1 **Undefined**

Bit 2 **Undefined**

Bit 3 **Undefined**

Bit 4 **RAM on system board**

0 Disable second 256KB of system board RAM

1 Enable second 256KB of system board RAM

Bit 5 **Undefined**

Bit 6 **Type of display**

0 Primary display is color graphics adapter

1 Primary display is monochrome

Bit 7: **Keyboard inhibit switch**

0 Keyboard inhibited

1 keyboard not inhibited

Output Port Bit Definitions

Bit 0 **System reset**

Bit 1 **Gate address line 20 of system processor**

0 Address line 20 on system bus inhibited

1 Address line 20 on system bus not inhibited

Bit 2 **Undefined**

Bit 3 **Undefined**

Bit 4 **Output buffer full**

0 Output buffer not full

1 Output buffer full

Bit 5: **Input buffer empty**

0 Input buffer not empty

1 Input buffer empty

Bit 6 **Keyboard clock (output)**

Bit 7 **Keyboard data (output)**

Test Input Port Bit Definitions

T0 Keyboard clock (input)

T1 Keyboard data (input)

Chapter Seven

System Timers

7.1 System Timers

The MI386 system board has three programmable timer/counters controlled by an Intel 8254-2 programmable interval timer chip. The Intel 8254-2 has three independent 16-bit counters and six software-programmable counter modes. To system software it appears as an array of four external I/O ports. Three ports are used as counters, and the fourth is a control register for mode programming. Each timer interrupts the CPU at the end of the programmed delay period. The timer channels are defined as channels 0, 1 and 2. They are used as follows:

Channel 0 System Timer

GATE 0	Always enabled
CLK IN 0	1.190 MHz clock
CLK OUT 0	Interrupt Controller, 8259A IRQ0

Channel 1 Refresh Request Generator

GATE 1	Always enabled
CLK IN 1	1.190 MHz clock
CLK OUT 1	Request Refresh Cycle

Channel 2 Speaker Tone Generation

GATE 2	Controlled by bit 0 of I/O port hex 61
CLK IN 2	1.190 MHz clock
CLK OUT 2	Audio frequency output to speaker

System Interrupts

The CPU may be interrupted by two Intel 8259A Interrupt Controler chips as well as the NMI signal. This allows 16 levels of interrupt, each with its own level of priority. Any of the interrupts including NMI can be disabled. The following table shows the interrupt level assignments:

Level		Function
NMI		System board memory parity or I/O Channel Check
Interrupt Controllers		
CTLR 1	**CTLR 2**	
IRQ0		Timer Channel 0 output
IRQ1		Keyboard Controler (output buffer full) interrupt
IRQ 2		Interrupt from CTLR 2
	IRQ 8	Real time clock interrupt
	IRQ 9	Software redirected to INT 0AH (IRQ 2)
	IRQ 10	Reserved
	IRQ 11	Reserved
	IRQ 12	Reserved
	IRQ 13	Coprocessor interrupt
	IRQ 14	Fixed Disk Controller
	IRQ 15	Reserved
IRQ 3		Serial Port 2
IRQ 4		Serial Port 1
IRQ 5		Parallel Port 2
IRQ 5		Paralalel Port 2
IRQ 6		Diskette Controller
IRQ 7		Parallel Port 1