Z-180 PC Series Laptop Computers

Owner's Manual

ZENITH | data systems

THE QUALITY GOES IN BEFORE THE NAME GOES ON

Z-180 PC Series Laptop Computers

Owner's Manual

ZENITH data systems

595-3853

THE QUALITY GOES IN BEFORE THE NAME GOES ON

Z-180 PC Series Laptop Computers

Owner's Manual

595-3853

Zenith data systems

REGULATORY INFORMATION

WARNING: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices terminals, printer, etc.) certified to comply with Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference with radio and TV reception.

NOTE: In order to meet Class B emission limits, the I/O cables that interconnect between this computer and any peripheral (such as a printer, external modem, etc.) must be shielded.

This equipment generates and uses radio frequency energy for its operation and if not installed and used properly, that is, in strict accordance with the instruction manual, may cause interference with radio and television reception. It has been tested and found to comply with the RF emission limits for a Class B computing device which are intended to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference with radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Move the computing device away from the receiver being interfered with.
- Relocate the computing device with respect to the receiver.
- Reorient the receiving antenna.
- Plug the computing device into a different AC outlet so that the computing device and receiver are on different branch circuits.
- Be certain that the computing device is plugged into grounded outlet receptacles. Avoid using AC cheater plugs. Lifting of the power cord ground may increase RF emission levels and may also present a lethal shock hazard to the user.

If you need additional help, consult your dealer or ask for assistance from the manufacturer. You may also find the following booklet helpful: How to Identify and Resolve Radio-TV Interference Problems. This booklet is available from the U.S. Government Printing Office, Washington D.C. 20402, Stock No. 004-000-00345-4.

FCC regulations Part 68 places three restrictions on using the modem:

1. The modem cannot be connected to a party line or coin-operated telephone line.

2. You must notify the telephone company that the modem is being installed and provide the following information: modem registration number, ringer equivalence number, and telephone number to which the modem is connected.

3. Notify the telephone company if you have questions about the operation of your telephone line when connected to the modem.

RESTRICTED RIGHTS LEGEND

This computer software and documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the Government is subject to restrictions as set forth in the governing Rights in Technical Data and Computer Software clause — subdivision (b)(3)(B) or DAR 7-104.9(a) (May 1981) or subdivision (b)(3)(ii) of DOD FAR Supp 252.227-7013 (May 1981). Contractor/Manufacturer is Zenith Data Systems Corporation of Hilltop Road, St. Joseph, Michigan 49085.

LIMITED RIGHTS LEGEND

Contractor is Zenith Data Systems Corporation of St. Joseph, Michigan 49085. The entire document is subject to Limited Rights data provisions.

Zenith Data Systems Corporation
St. Joseph, Michigan 49085

Contents

PART I — INSTALLATION

PART II — OPERATION

PART III — REFERENCE

PART IV — SERVICE

Figures

Tables

Listings

Welcome

Congratulations on your purchase of a new Z-180 Laptop computer. The Z-180 Laptop computer, shown in Figure 1, is designed for people on the go: executives, managers, sales reps, service reps, and writers, to name a few. These are people who spend a lot of time away from the office but need to keep in contact. For these people, the Z-180 is capable of satisfying both their business and personal needs.

With Zenith Data Systems, the future is here today. Your new computer can act as a stand-alone tool or function as the heart of a powerful and expandable business automation package. Combined with Zenith Data Systems software, your new computer can produce practical and affordable solutions to your business problems. It can easily satisfy a variety of business applications, such as data processing, telecommunications, networking, and financial analysis to aid you in making decisions that affect your business.

This Owner's Manual has been prepared for you, the new computer user. Welcome to computing at its best and accept Zenith's assurance and commitment that the quality goes in before the name goes on.

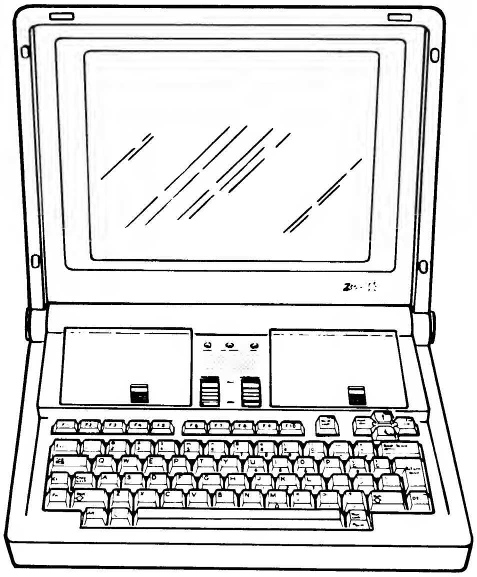

Figure 1. The Laptop Computer

This Owner's Manual is divided into four sections:

Installation — The first section tells you how to set up your new Z-180 Laptop computer. You will also learn how to connect peripherals to your computer, making your computer part of a system.

Operation — This section contains all the information needed for the first time user to begin operation of the computer.

Reference — This section is an expanded operation section. It contains technical information useful to an experienced programmer.

Service — The last section contains service information intended for use by service technicians, installation information for the various options, and information on replacement parts.

Part I — Installation

Chapter 1
System Installation

Your Computer System

Your new Z-180 Computer is a tool, much like a typewriter or calculator. You can use it in a wide number of applications, such as general accounting, inventory management, payroll, communications, maintaining mailing lists, filing and retrieval of information, and word processing. It is also an excellent educational aid that can make the learning process fun and challenging for the student while providing management support for the teacher and school.

Your computer is an extremely reliable machine. When it is properly installed and receives proper care, it will last for years with little need for service.

A typical computer system will consist of at least two units: the computer and a peripheral (any piece of equipment that is attached to and controlled by the computer). It may be a video display, printer, or additional disk drives.

The heart of the system is the computer with its keyboard, video display and storage devices. Refer to Figure 1-1.

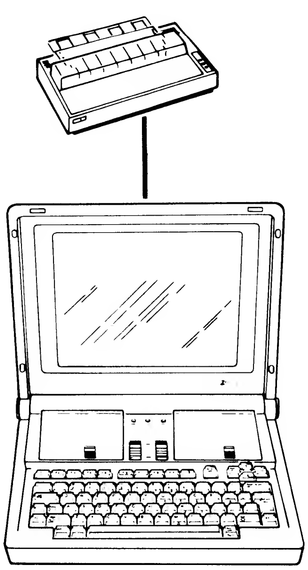

Figure 1-1. Typical System Components

The video display (sometimes called the LCD, or liquid crystal display) is like a television screen. This is where you will see and work with the programs you use in your computer system. A seperate video monitor that is connected to the computer with a cable also may be used.

You will use the keyboard to enter information into your computer. It has all the keys contained on a standard typewriter keyboard plus additional special function keys. There is also a calculator-style "keypad" designed into the right side of the main keyboard that will allow you to make faster numeric entries during some application programs. (Information can also be placed in the computer from disks, signals over telephone lines, and other special devices.)

The floppy disk capability that is built into your computer allows you to store large amounts of information on 3.5-inch plastic disks (refer to "Disk Drives and Disks" in Chapter 7). Each disk that your computer uses is capable of holding up to 300 pages of typewritten text. You can get even more storage by adding external floppy disk drives or a high-capacity rigid disk drive. You can greatly enhance the operation of your computer by adding a printer to your system. This makes it possible to produce copies of letters, accounting records, or any other material.

Finally, you may add computer-to-computer communications via telephone. This capability makes it possible for your computer to exchange information with other computers over the telephone, which makes available electronic mail service, news, shopping services, electronic banking and bill paying, stock market services, and so on.

Setting Up

The following paragraphs will describe setting up each connector on your computer. While examining the back panel and the connectors, be sure the computer is turned off and is disconnected from any external power source. Rotate the computer so you are looking at its back and open the rear door.

Back Panel

The following connectors are located on the lower back panel (refer to Figure 1-2).

EXT BUS — This 50-pin connector provides access to the system bus for future expansion or connection of a hard disk drive system.

FDD/BCR — This connector provides communications with an external floppy disk drive(s), or a bar code reader.

RGB/Video — Provides the signals for an external monochrome or RGB monitor.

Printer — Allows a Centronics-type parallel printer or other parallel device to be attached to the computer.

RS-232C — Allows a serial printer, electronic sketch pad, or other serial device to be connected via an RS-232 cable.

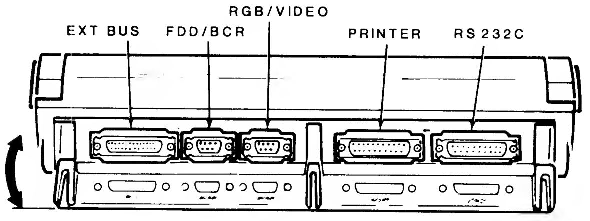

Figure 1-2. Back Panel

Peripheral Connections

You should select an area to set up your computer that:

- Accomodates the computer and all of its peripherals.
- Has a level work surface that is near a power source and a telephone.
- Has an environmental range of 50 to 104 degrees F (10 to 40 degrees C) and a 20 to 80 percent relative humidity.

Refer to Figure 1-3. Position the computer where it normally will be operated and so you can get to the back panel to make the following connections.

External rigid disk drive system — Connect the external rigid disk drive system to the connector marked EXT BUS.

External floppy disk drive system — Connect the external floppy disk drive system to the connector marked FDD/BCR.

RGB/composite video monitor — Connect the video monitor to the connector marked RGB/Video.

Parallel printer — If you have a Centronics-type parallel printer or other parallel device, connect it to the connector marked Printer.

Serial printer — If you have a serial printer or other serial device, connect it to the connector marked RS-232C.

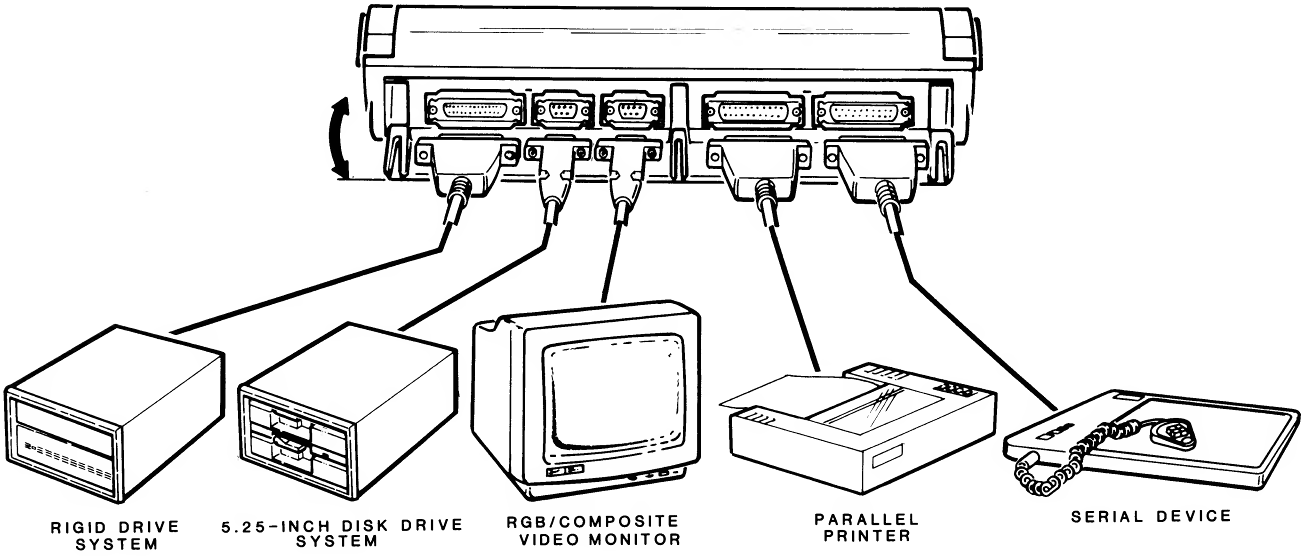

Figure 1-3. System Installation — Peripheral Connections

Internal Modem Connections

Refer to Figure 1-4. Position the computer so you can get to the left side of the computer (connectors marked LINE and TEL).

Disconnect the cable from the telephone and connect it to the LINE connector on the computer. Connect the cable packed with the computer from the telephone to the TEL connector on the computer.

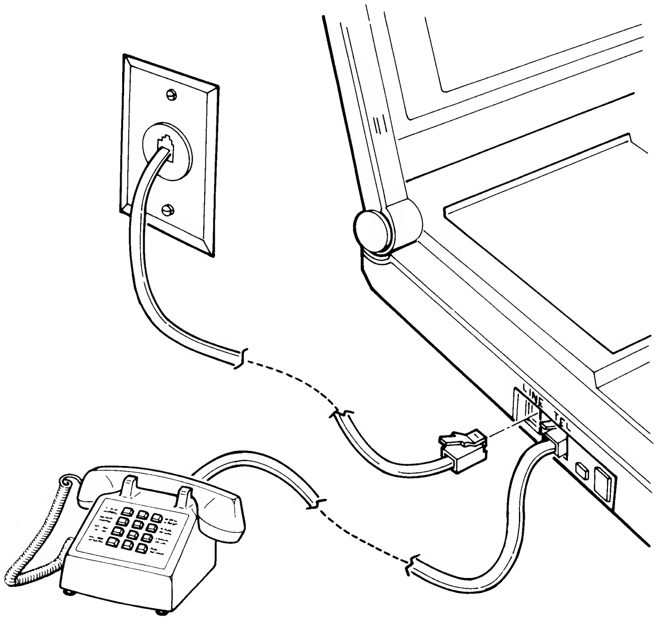

Figure 1-4. Connecting the Modem

Power Connections

The Laptop computer may be operated from either internal batteries (already installed) or an external AC or DC power source.

Normally you would operate the computer from its built-in battery pack. There will be occasions when you will need to recharge the computer or operate it at the same time.

The computer can be recharged or operated from two different external power sources: 120 VAC or 12 VDC (operate only). Each uses a different adapter.

The 120 volt AC adapter contains a step-down transformer and DC rectifier to produce approximately 18 volts DC.

The DC adapter contains no circuitry and is intended to be used as an automotive adapter that draws current through a cigarette lighter.

CAUTION: You should never attempt to start a motor vehicle while the DC adapter is being used with the computer. The starting circuits of most motors will send transients back through the power system which could damage the computer or cause it to malfunction.

Refer to Figure 1-5. The correct procedure for plugging the adapter into the computer while it is operating is to first, plug the adapter into its power source and then plug the adapter into the computer. The jack in the computer is designed so that power from the battery pack will not be interrupted while you plug in the adapter.

CAUTION: Do not leave the Laptop computer plugged in to an external AC or DC power source when it is not operating unless you are charging the built-in batteries.

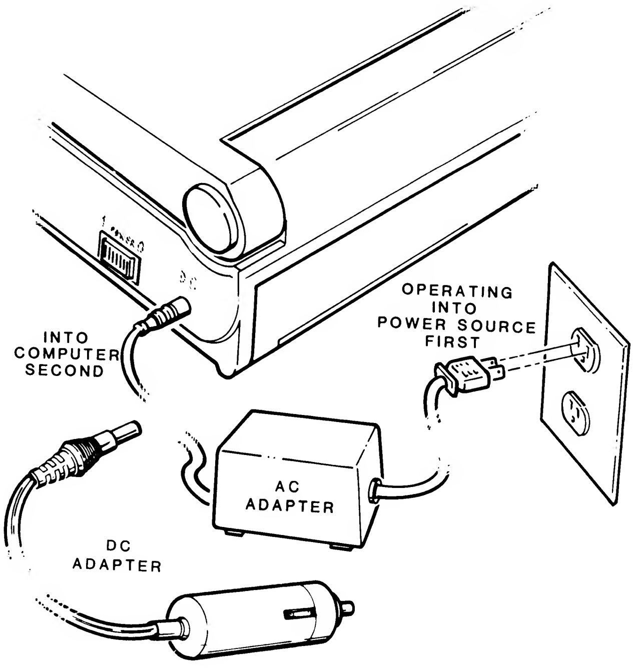

Figure 1-5. External Power Adapters

Refer to "Disks Drives and Disks" in Chapter 7 for information about disks and disk care.

In addition to the internal disk drives the Laptop computer can accept input from other storage devices. For instance, an optional 5.25-inch disk drive system may be attached to the rear panel of the computer, increasing storage capability and making it easier to transport information to other PC-compatibles (refer to Figure 2-4). High-capacity rigid disk systems can also be attached to the Laptop computer thereby increasing its speed and performance.

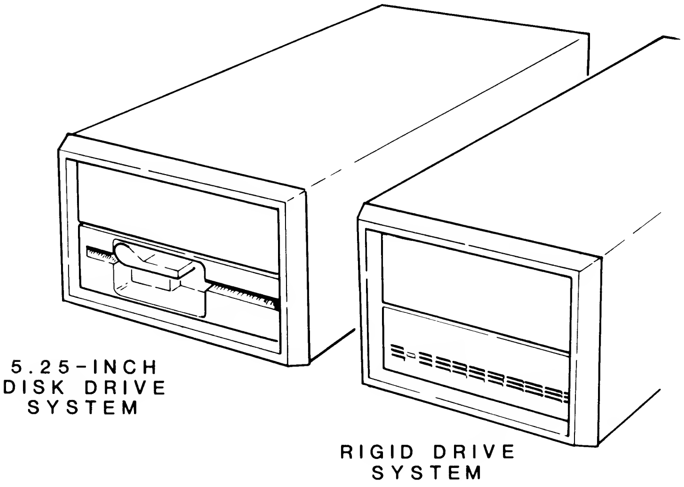

Figure 2-4. External Disk Systems

Disk Loading

Loading 3.5-Inch Disks

Refer to Figure 2-5 while reading this section.

Disk Drive latch — Push the drive latch toward the rear of the computer and let the drive assembly pop up.

Disk Load Slot — Insert a disk in this slot with the label up and toward you. If the disk is not oriented as shown, the computer will not operate properly and you may damage the disk or the disk drive. The disk should slide smoothly into the drive until you hear a click. When the disk has clicked into place, push down on the top of the drive to engage the disk.

NOTE: If you accidently insert a disk incorrectly and it becomes stuck, refer to Chapter 5 for information on how to remove the disk.

LED Disk Access Indicators — These lights, located between the two drives on your computer and labeled Drive A and Drive B, indicate that the computer is attempting to read from or write to a disk. The read/write operation will be successful only if the disk is inserted properly.

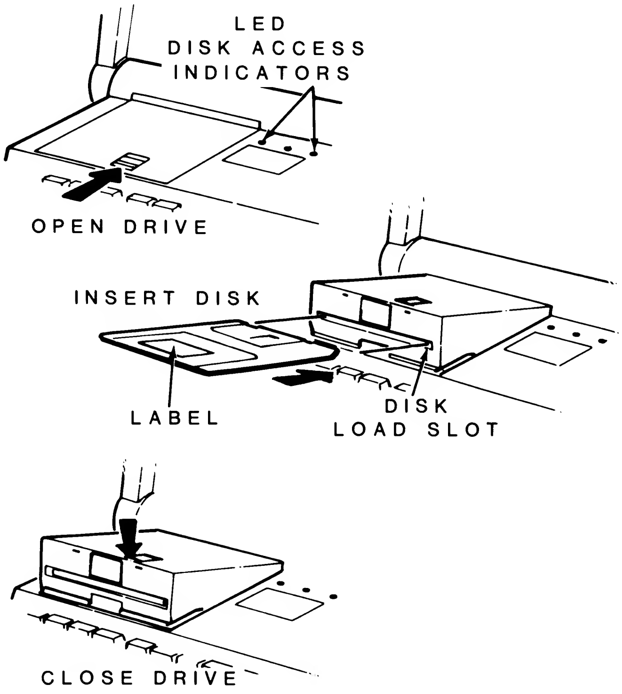

Figure 2-5. Loading 3.5-Inch Disks

Loading 5.25-Inch Disks

Refer to Figure 2-6 while reading this section.

Disk Drive Slot — Insert 5.25-inch disks in this slot with the label up and toward you. If the disk is not oriented as shown in the figure, the computer will not operate properly and you may damage the disk and/or the disk drive. If the disk meets resistance while you are inserting it into the drive, check to make sure that there is not a disk already in the drive. It should slide smoothly into the drive.

Disk Drive Latch — Your floppy disk drive may have a latch, door, or handle. It is used to secure the floppy disk in the proper position in the disk drive. Closing the latch engages the drive.

LED Disk Access Indicator — This light, which may be in one of several different locations on the front of your disk drive, indicates that the computer is attempting to read from or write to the disk. The read/write operation will be successful only if the disk is inserted properly and the disk drive latch is closed.

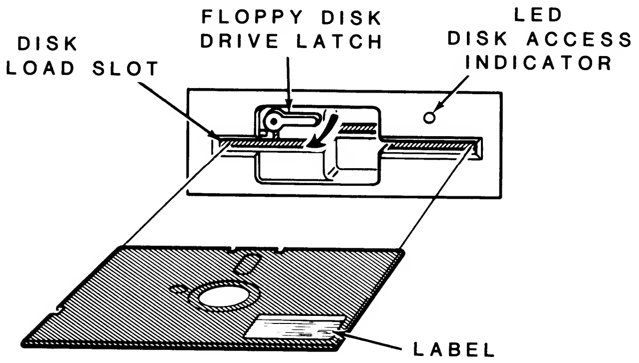

Figure 2-6. Loading 5.25-Inch Disks

Power Up

You are now ready to see your new computer in operation. The following directions will help you power up your computer and load the operating system from a disk.

Open the disk drives and remove the disk drive head protectors.

The power switch is located on the right side of the computer (refer to Figure 2-7). To turn the power on, slide the switch toward the front of the computer.

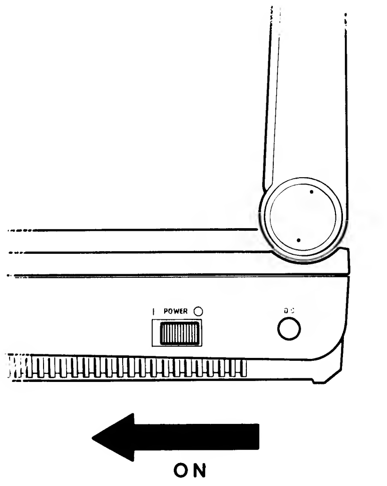

Figure 2-7. Power Switch

Turn on the power switch. A blinking cursor should appear in the upper left corner of the LCD and disk drive A's access indicator light should glow.

After about 10 seconds, the following message should appear on the display:

```
+++ DISK ERROR: Drive not ready! +++
```

Power Down

To power down and close your computer, first remove the disks from the disk drives. Remember to save any data you may have been working on.

Slide the power switch toward the rear of the computer to turn off the power.

Insert the disk drive head protectors into the disk drives and close the disk drives.

Close the cover and push the latches toward the edge of the computer.

If connected, disconnect the AC adapter from the computer unless you are charging the battery.

Chapter 3

Operating the Computer

The Monitor Program

Your computer contains a special program, called Monitor, that is designed to help you get started. One of its functions is to load (boot) the disk operating system from the disk into memory. In addition, Monitor performs some initial self-tests to see if the computer is operating properly. Chapter 8 contains detailed information on the Monitor program.

Autoboot

Your computer is set at the factory to autoboot, that is, to automatically load the operating system when the computer is turned on. The computer will attempt to load the operating system from drive A. If there is a hard disk drive system attached to the computer and there is no disk in drive A, then the computer will attempt to load the operating system from the boot partition of the hard disk system.

There are two ways to defeat the autoboot function:

- Pressing the **ESC** key just after the access indicator LED for drive A begins to glow.

- Pressing **CTRL-ALT-FN-INS**. Press and hold the **CTRL** key first then press the other keys in sequence. Release all keys at the same time.

If your computer is not on, turn it on. Refer to Chapter 2, Power Up.

Press the **ESC** key and the screen will show:

```
MFM-180 Monitor. Version 2.7B
Memory Size:640K bytes
Enter "?" for help.
->_
```

Your computer is now in the manual mode and you can use the Monitor commands.

Introduction to the Commands

The Monitor program will execute 18 different commands, a summary of which is shown in Table 3-1. Refer to Chapter 8 for information about the Monitor program.

Table 3-1. Monitor Command Summary

COMMAND	NAME	DESCRIPTION
?	Help	Displays a list of commands.
B	Boot from disk	Loads the operating system from a disk.
C	Color bars	Displays a color bar pattern on the screen.
D	Display memory	Displays a block or portion of memory in hexadecimal.
E	Examine memory	Examines and/or changes the contents of a memory location.
F	Fill memory	Fills the specified range of memory with the data specified.
G	Execute (Go)	Executes the program beginning at the specified address or breakpoint.
H	Hex math	Displays sum and difference of two specified hexadecimal numbers.
I	Input from port	Returns the contents of the specified port.
M	Move memory block	Moves a block or portion of memory to the specified destination.
O	Output to port	Sends a value to the specified port.
R	Examine registers	Displays the content(s) of a register or registers.
S	Search memory	Searches memory for specified "string."
T	Trace program	Single steps through a user program.
U	Unassemble program	Unassembles (or disassembles) a portion of memory into assembly-language form.
V	Set Video/Scroll	Sets (changes) the video or scroll to the specified mode.
TEST	Extended diagnostics	Displays the diagnostics menu.

Loading MS-DOS

A disk operating system, or DOS, is a program that lets you do certain tasks, such as organize the disk (with the FORMAT command), copy information from one disk to another (with the COPY command), and load other programs.

MS-DOS was prepared for Zenith Data Systems Z-180 Laptop computers by Microsoft. It is compatible with the same MS-DOS that runs on other PC-compatible computers.

SETUP is an MS-DOS utility that can be used to make working copies of your MS-DOS distribution disks. SETUP can also be used to make "working disks." These are disks that contain system files so you can boot up from the disk without loading MS-DOS separately.

SETUP will also set up a rigid disk. it will format the rigid disk partition, transfer the system files, set up a *BIN* (subdirectory), transfer the utility files to the subdirectory and establish a path to it.

Refer to "Getting Started" in your *MS-DOS User's Guide* for information on loading and using the SETUP distribution disk.

NOTE: The MS-DOS documentation tells you to use the CTRL-ALT-INS combination to reset the computer. Be sure to use the CTRL-ALT-FN-INS instead. Table 7-3 in Chapter 7 list a cross reference of the key combinations used on a PC to those used by the Laptop computer.

Setting the Real-Time Clock

Your computer is equipped with a real-time clock that MS-DOS uses to set its DATE and TIME functions. The clock can maintain the proper date and time for up to three days without having power applied to the computer. If the computer is not powered up for a period of three days or more, you will have to reset the clock.

The MS-DOS DATE and TIME commands will not set the real-time clock. To set the real-time clock date and time, use a program named RTCLOCK.COM that is included on the MS-DOS distribution disk. Refer to your *MS-DOS User's Reference* manual for information on setting the real-time clock.

LCD Backlight Illumination

Your computer is equipped with a circuit that, after a period of inactivity on the keyboard, turns off the backlight on the display to conserve power during battery operation. Pressing any key will turn the backlight back on. The length of time the backlight remains on can be varied using a program named EL.COM that is included on the MS-DOS distribution disk.

To determine the current saving time, that is, the length of time the backlight will remain lit with no activity on the keyboard, enter the command EL at the MS-DOS prompt and press RETURN. The computer will display a message similar to:

```
A>EL

Current Saving Time is 2 minute(s)
```

Entering the command EL followed by a number between 1 and 10 will change the saving time to that number of minutes. For example, entering:

```
A>EL 3
```

Will change the saving time to three minutes.

Entering EL followed by 0 (zero) will cause the backlight to remain on constantly. This should only be done while operating from one of the external power adapters. Leaving the backlight on constantly during battery operation can reduce the battery life.

Turning on the Modem

Your computer is equipped with an internal modem. When the computer is powered up, the modem is turned off (power saver mode). There are two ways to turn the modem on: one is entering an out command at the Monitor prompt and the other is using a utility named MODE.COM included on the MS-DOS distribution disk at the MS-DOS prompt.

To turn the modem on from the Monitor prompt, enter:

```
->0C0,1      (enable control lines to modem).
->0 2ff,1    (power on to modem, power saver mode
             off)
```

To turn the modem off from the Monitor prompt, enter:

```
->0 2ff,0    (power to modem off, power saver mode
             on)
->0C0,0      (disable control lines to modem)
```

To turn the modem on from the MS-DOS prompt, enter:

```
A>MODE MODEM ON
```

To turn the modem off from the MS-DOS prompt, enter:

```
A>MODE MODEM OFF
```

Entering just the command `MODEM` will cause the computer to display the status of the modem, that is, `modem on` or `modem off`.

Refer to "Modem Operation" in Chapter 7 for information on using the modem and its commands.

Setting up the Hard Disk

The hard disk is, in most cases, actually a stack of round metal platters coated with a magnetic material similar to that used by floppy disks and audio tape. Hard disks come in a variety of sizes, capacities, and configurations.

The read/write head (which reads information from and writes information to the surface of the disk) floats on a cushion of air above the disk as it turns. The distance between the head and the surface is so small that even a tiny smoke particle could cause irreparable damage to the disk. For this reason, most hard disk systems are sealed against contamination.

The hard disk can be divided into one or more partitions. A partition is a portion or even the entire hard disk that is used as a logical drive for a particular application or operating system. Hard disk systems from Zenith Data Systems are prepared before they leave the factory and are preallocated to one standard partition for immediate use (IBM-compatible standard).

A logical drive performs similarly to a floppy disk drive. Because a partition is considered to be a logical drive, it will be treated the same as any individual floppy disk drive and identified with a drive name, such as C, D, or E. It may contain an operating system or may be used strictly for data storage.

Several utilities are supplied on the MS-DOS distribution disk to support the hard disk system. Refer to "Hard Disks" in the *MS-DOS User's Guide* and to "Hard Disk Commands" in the *MS-DOS User's Reference* manual.

Booting from the Hard Disk

The Z-180 Laptop computer is designed to search for the operating system when it is first turned on. It will first check drive A for a bootable disk. If no disk is present it then attempts to boot from the default boot partition of the hard disk or from the external floppy disk drive.

The autoboot procedure can be defeated by pressing the ESC key when the access indicator for drive A begins to glow or by pressing CTRL-ALT-FN-INS.

At the Monitor prompt enter `BW` and press **RETURN**. The computer will then boot from the default boot partition.

If you have another partition with an operating system on it, partition 3 for example, and you want to boot from it, enter:

```
BW:3
```

Battery Operation

When it is inconvenient to use a power outlet, the NiCad battery installed in the computer will supply the necessary power. When the battery voltage runs low, the LOW POWER indicator will light.

The computer will operate for approximately three minutes after the LOW POWER indicator lights the first time. After the LOW POWER indicator lights, if possible, connect the AC adapter. If it is not possible to connect the AC adapter, save the data.

It is possible to connect the AC adapter to the computer while it is operating without interrupting operation by first connecting the adapter to a 120V outlet and then connecting the adapter to the computer. The battery can be charged while you are working on the computer.

Battery Pack Care and Use

IMPORTANT: It is essential to charge the battery for a minimum of eight hours before you use it for the first time. Do not leave the AC adapter plugged in for more than 12 hours.

For **maximum** battery life, use the following procedures.

First Time Use — There are several things that affect battery life, such as how often the disk drives are used, the level of backlight used, and whether the modem is turned on.

To condition your battery to obtain the longest battery life, completely discharge and fully recharge the battery three times. You can do this by turning the computer on and allowing the computer to run, discharging the battery until the LOW POWER indicator lights. Allow the computer to run until the backlight ceases to function and the display on the screen becomes garbled. Connect the AC adapter and allow the battery to recharge for eight hours. Repeat this procedure two more times.

Regular Use — After you have initially conditioned your battery, keep the following points in mind for long-lasting battery use:

- Never charge the battery pack for over 12 hours.
- Minimize the use of disk drives as much as possible.
- Use the minimum backlight possible for comfortable viewing.

After initial conditioning of your battery, if you experience any of the conditions noted below, refer to "Battery Care" in Chapter 15. If the conditions persist, have the battery tested and replaced if necessary.

- The low power indicator lights after less than 30 minutes of use from a fully-charged battery.
- Battery life is less than 1 hour.

Chapter 4

Installing Application Packages

Most application software will run with few or no problems. It is usually set up to operate in a dual floppy disk environment, where drive A represents the first 3.5-inch floppy disk drive and drive B represents the second.

Many application packages supplied by Zenith Data Systems have an install utility which allows you to install and run your software on a hard disk. For those application packages which do not have an install utility, you will need to remap (with the MS-DOS ASSIGN command) drive names A and B to the respective partitions.

Disk Drive Name Assignment

Under MS-DOS drive names are assigned only to the floppy disk drives and to the default boot partition. The actual drive name assigned to each disk drive and partition depends on the setting of sections 1, 5 and 6 of the configuration switch (refer to Chapter 8 for the location and settings of the configuration switch).

Table 4.1 lists the possible floppy disk drive and partition combinations and the drive names assigned to each when switch section 1 is set to ON (factory set position).

Table 4-1. Floppy Disk Drive and Partition Name Assignments, Switch Section 1 ON

NUMBER OF FLOPPY DISK DRIVES	CONFIGURATION SWITCH SECTION 5	6	INTERNAL DISK DRIVE(S)	EXTERNAL DISK DRIVE(S)	EXTERNAL HARD DISK
1	ON	ON	A		C
2	OFF	ON	A,B		C
3	ON	OFF	A,B	C	D
4 *	OFF	OFF	A,B	C,D	E

*Factory set position

Your computer is set at the factory with switch sections 5 and 6 OFF. Therefore, the two internal disk drives would be named A and B, the two external disk drives, if attached, would be named C and D, and the default boot partition of the hard disk system would be named E.

You may establish up to four partitions on the hard disk system with the PART utility. You would use the ASGNPART utility to assign the additional partitions the names F, G and H. You can use AUTOEXEC.BAT to automatically assign the drive names to the partitions upon bootup or, if you prefer, you may assign some other easily remembered name to a batch file for that purpose. Refer to the MS-DOS documentations for details.

With switch sections 5 and 6 set for 3 or 4 drives and switch section 1 set to off, the drive names for the first internal disk drive and the first external disk drive are swapped. This makes the first internal disk drive C and the first external disk drive A. This arrangement allows you to run 5.25-inch copy-protected software or software that requires a key disk.

Table 4.2 lists the possible floppy disk drive and partition combinations and the drive names assigned to each when switch section 1 is set to OFF.

Table 4-2. Floppy Disk Drive and Partition Name Assignments, Switch Section 1 OFF

NUMBER OF FLOPPY DISK DRIVES	CONFIGURATION SWITCH SECTION 5	6	INTERNAL DISK DRIVE(S)	EXTERNAL DISK DRIVE(S)	EXTERNAL HARD DISK
1	ON	ON	A		B
2	OFF	ON	A,B		C
3	ON	OFF	C,B	A	D
4 *	OFF	OFF	C,B	A,D	E

*Factory set position

Copying Files to the Hard Disk Partitions

Now, you must select the appropriate MS-DOS partition (and, optionally, directory path) that you will use for your working copy of the application package. The software package may include program files, data files, operating system files, and computer language files. You will need to make sure that the amount of space available on the target hard disk partition is sufficient. Use the MS-DOS DIR command to determine the amount of space required for your files and the CHKDSK command to determine the amount of space available on your selected partitions.

Generally, one partition may be used to hold all of your files, or you may choose to store selected data files in a second partition. After you have selected the target partitions, determine the names of the files that are to be transferred from the distribution disks to the partitions.

Copy the selected files from the distribution disks to your partitions. When you have finished copying all of the selected files to your hard disk partitions, store the distribution disks in a safe place.

Consult the documentation that comes with your software to determine if you need additional software files (such as MS-DOS utilities or perhaps a computer language like BASICA) to operate your programs. If you need additional files, such as those mentioned, then transfer them from the appropriate distribution disks.

When you are finished, you have completed the installation procedure for a working copy of your software. You may now proceed with the normal operation of your software as described in its documentation.

Chapter 5

Backing up Disks

Because it is possible to store the equivalent of several dozen floppy disks on the hard disk, two commands, BACKUP and RESTORE, have been added to the MS-DOS operating system to aid you in backing up your partition. You may elect to copy only certain files, files updated recently, or the entire partition. The basic operation of BACKUP is to create a directory of the specified files and then copy them to a single file. The single file may actually be several files that each contain a sequentially numbered extension such as BACKFILE.000, BACKFILE.001, and so on. The system backs up your files end to end to save floppy disk space.

The operation is further enhanced by providing an automatic FORMAT procedure for the floppy disks, protection against unwanted overwriting of existing files on the hard disk, verification, and so on. For complete details of the features and operating of BACKUP and RESTORE, refer to your MS-DOS software documentation for the hard disk.

Part III — Reference

Chapter 6
Specifications

The following specifications are for the Laptop computer. Optional features that may or may not be present on some models are indicated.

CPU	
Processor	80C88 CMOS 16-bit processor.
Type	16-bit internal.
Clock speed	4.77 MHz (IBM standard)/8 MHz.
Memory	640K dynamic RAM.
Display	LCD device.
Capacity	80 × 25 characters in text mode. 640 × 200 pixels in graphics mode. PC-compatible in normal text and graphics modes.
Size	8″ wide × 6.13″ high (49 sq. in.), standard. 9.44″ wide × 4.13″ high (39 sq. in.), optional.
Contrast ratio	12:1 backlit. 6:1 nonbacklit. Measured using Weston Gray Scale Comparison.
Tilt angle	135° maximum (150° relative to keyboard).
Viewing angle	22.5° minimum from nominal viewing point. Nominal viewing point adjustable by brightness and contrast controls.
Life of display	4000 hour backlit. 6000 hours nonbacklit.
Sound	30 mm, piezoelectric element.
Input/output	
Serial port	Asynchronous serial RS-232C (DB-25 connector) port — 1 start bit; 7- or 8-bit word length; 1 or 2 stop bits; selectable baud rates of 110, 150, 300, 600, 1200, 2400, 4800, or 9600 baud; RD, CTS, DSR, CD signals recognized; TD, RTS, DTR control signals generated; half-or full-duplex operation.
Parallel port	Centronics-type parallel output port (DB-25 connector).
Video	Composite monochrome video and RGB color video with intensity signals from a single 9-pin connector.
Modem	RJ-11 modular line and telephone connectors. 300/1200 baud, Hayes AT command set compatible.
Disk drives	Internal 3.5-inch double-sided, double-density disk drives, 720K formatted capacity each drive, 135 tpi, 9 sectors per track. Write-protection recognized. Optional external 5.25-inch double-sided, double-density floppy disk drives, 360K formatted capacity each drive, 48 tpi, 9 sectors per track. Write-protection recognized. Optional external rigid disk drive, 10M formatted capacity. Optional external rigid disk drive, 20M formatted capacity (5 MB/sec transfer rate).

Operating system	MS-DOS version 3.2.
Keyboard	75 keys: 60-key alphanumeric typewriter arrangement, 1 multifunction key, 4 cursor control keys, and 10 function keys. Full PC keyboard compatibility maintained by using mode switching and multiple keys to duplicate keypad and special function key operation.
Power requirements	+12 VDC at 1.0 amperes (12.0 watts), or 120 VAC at .175 amperes (21 watts).
External power adapters	
Input voltages	120-volt adapter: 120 VAC (95 — 135 VAC). Auto adapter: 12 VDC (13.5 VDC nominal).
Input frequency	120-volt adapter: 50 or 60 Hz (48 — 62 Hz). Auto adapter: not applicable; DC voltage input.
Current	120-volt adapter: .175 amperes maximum continuous under full load. Auto adapter: 1.05 amperes maximum continuous under full load.
Output voltage	120-volt adapter: +12 VDC at 1.0 A. Auto adapter: input voltage (no voltage conversion).
Internal battery pack	
Capacity	2.0 AHr.
Output voltage	+12 VDC at 1.0 amperes under full load.
Life	200 charge-discharge cycles (minimum)
Weight	2.0 AHr pack - 1.5 lbs. (.680 kg) approximate.
Shelf life	3 to 5 years.
Operating length	3 hours minimum before low battery indicator during heavy use (25% disk accesses), 5 disk copies possible before battery failure.
Environment	
Operating	40 — 105 degrees Fahrenheit (5 — 40 degrees Celsius) at 20 — 80% relative humidity (non-condensing).
Storage	−13 — +133 degrees Fahrenheit (−25 — +56 degrees Celsius) at 20 — 80% relative humidity (non-condensing) with 1 hour recovery period.
Shock vibration	100 G for 1 ms during transportation. .15 G @ 5 to 300 Hz during operation. 1 G @ 5 to 300 Hz during transportation.
Dimensions	13.4″ wide × 11.75″ deep (front to back) × 3.1″ tall at back (34.29 cm × 29.85 cm × 8.89 cm) with display closed. 13.375″ tall (33.97 cm) with display in full upright position.
Weight	11.8 lbs. (5.31 kg) with standard configuration (two drives, 640K RAM, modem and battery). Additional options/accessories will affect system weight.

Chapter 7
User Reference

The material in this chapter expands upon the material presented in Part II of this manual. It also presents the theory behind the operation of many parts of the computer, along with configuration information and material concerning battery and AC operation.

Configuration

The configuration switch sections may be changed without removing any of the covers on the computer. In some computers the configuration is controlled by hardware jumpers and switches while other computers utilize a software or firmware setup procedure. This computer uses a combination of switch settings and firmware to configure it for proper operation.

Only one 6-position switch is used for configuring the computer. It is located on the main board and is protected by a small rubber plug on the bottom of the computer's cabinet. To access the switch, remove the rubber plug. Refer to Figure 7-1 for its location and Table 7-1 for a description of each switch section. The text that follows the table explains each switch option.

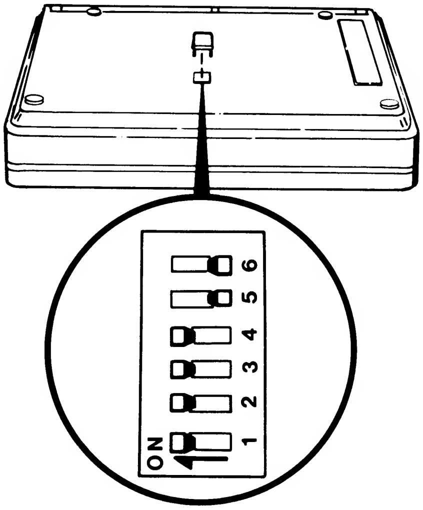

Figure 7-1. Configuration Switch Location

Table 7-1. Configuration Switch Settings

SWITCH SECTION	DESCRIPTION	SETTINGS
1	Drive A select	
	Internal floppy disk drive	ON*
	External floppy disk drive	OFF
2	Clock frequency	
	4.77 MHz (IBM standard)	ON*
	8 MHz	OFF
3	Display width	
	80 characters	ON*
	40 characters	OFF
4	Display type	
	Internal (LCD)	ON*
	External (CRT)	OFF
5 and 6	Drive Count	
	One internal drive	ON, ON
	Two internal drives	OFF, ON
	Two internal, one external drive	ON, OFF
	Two internal, two external drives	OFF, OFF*

* factory setting

Drive A select — This switch tells the computer where drive A is located. Place this switch ON to have the internal 3.5-inch floppy disk drive as drive A. The external 5.25-inch floppy disk drive would then be drive C. Place this switch OFF to have the external 5.25-inch floppy disk drive as drive A. The internal 3.5-inch floppy disk drive would then be drive C.

Clock frequency — The computer can be set to operate at either of two speeds. Place this switch ON to operate at 4.77 MHz (IBM standard). Place this switch OFF to operate at 8 MHz.

Display width — The computer can be turned on in one of two display width modes: 40 or 80 characters per line. Select the ON position for a display width of 80 characters per line. Select the OFF position for a display width of 40 characters per line. To provide PC compatibility, 80 characters per line is the most widely used mode.

Display type — The computer supports only one type of display at a time to conserve power. Select the ON position to use the built-in display. Select the OFF position to use an external monitor. The ON position is the factory setting since the computer is normally used with the LCD.

Drive count — These two switches tell the computer how many drives are connected. Place switches 5 and 6 ON to specify a one-drive machine. Place switch 5 OFF and switch 6 ON to specify a two-drive machine. Place switch 5 ON and switch 6 OFF to specify a two-drive machine with one external drive. Place switches 5 and 6 OFF to specify a two-drive machine with two external drives. Normally, the computer will have two built-in drives and no external drives connected.

Since different types of switches may be supplied, refer to Figure 7-2 to determine the switch type in the computer and how to determine the on position.

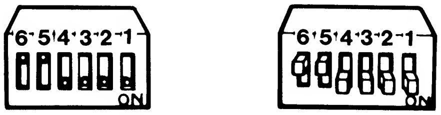

Figure 7-2. Switch Types

Power

The computer may be operated from one of three power sources: built-in battery, 120-volt AC power adapter, and 12-volt DC automotive cigarette lighter power adapter.

Refer to Figure 7-3. To use the computer with an external power source, plug the adapter or cable into the power source first, then into the computer. The design of the jack in the Laptop computer allows you to connect the computer to an external power source without interrupting operation of the computer.

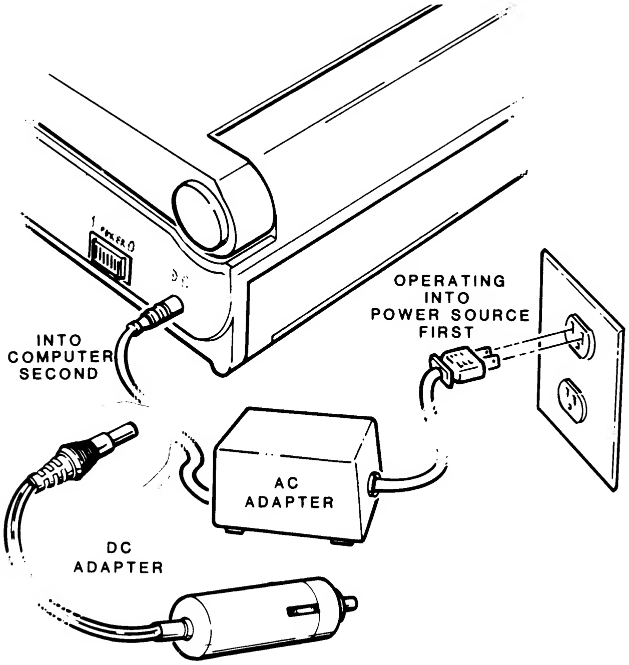

Figure 7-3. Connecting an External Adapter

Powerup Sequence

When you turn on the power, a sequence of operations takes place inside the computer to make sure that everything is ready to go. These powerup self-tests check handshaking between the CPU and other parts of the system. The disk drive heads make some noise as they are synchronized to the system during this sequence.

Then the computer will attempt to autoboot a disk in drive A, which you can defeat by pressing the ESC key. If no disk is in drive A, the computer will check any external drives that may be attached, starting with floppy disks and then the hard disk system. The autoboot feature cannot be permanently disabled.

With no disk in any drive, the following error message will appear on the screen about 10 (floppy disk only systems) to 20 seconds after you turn on the computer.

```
+++ DISK ERROR: Drive not ready! +++
```

This normal error message indicates that no disk was placed in a drive. If the message appears, you need to place a bootable disk in drive A.

Battery Operation

Your computer uses rechargeable nickel-cadmium batteries (ni-cads) for portable operation. The length of time you can operate your computer from fully charged batteries depends upon a number of factors, including computer model, features (backlight, the amount of disk activity, and if the modem is turned on, and full memory), and the battery pack capabilities.

Rechargeable nickel-cadmium batteries (ni-cads) have an unusual operating characteristics that are not experienced with other types of batteries. This characteristic is described here to help you obtain the longest possible operating time from a fully charged battery pack.

Ni-cads will develop a reduced operating cycle if they are recharged on a continuous basis or if they are not operated as long as possible before being recharged. Typically, the operating period can be as short as 10 minutes before the low-battery indicator on the computer will start flashing. Therefore, keep the following in mind when you use your computer.

- Charge the battery pack for eight to twelve hours. Eight hours will usually suffice to charge a used battery pack. Twelve hours is the maximum charging time before you start overcharging the battery pack.

- You may operate the computer from an external power source without overcharging the installed battery pack. The computer is equipped with an internal charge cut-off circuit designed to prevent overcharging the battery; however, you should disconnect the external source when you turn your computer off to avoid inadvertently overcharging the battery pack.
- If the computer is not going to be used for portable operation, remove the battery pack and operate it from an external AC adapter.
- Operate the computer from batteries for as long a period of time as possible before recharging them or plugging in the external adapter.

If you experience shortened operating periods with the ni-cads in your computer, refer to Chapter 15 to remedy the problem.

Keyboard

The computer's keyboard has 75 keys, which are discussed in groups. All keys except the ALT, SCROLL LOCK/BREAK, CAPS LOCK, CTRL, INS/DEL, NUM LOCK, and SHIFT keys will generate repeated entries. If the key is held down, the letter or symbol will start repeating at about 10 times a second. After about one second, the rate will increase to about 20 times a second.

Alphabetic Keys

Refer to Figure 7-4. The computer has the standard 26 letters of the alphabet arranged as they are on a typewriter. These keys allow you to enter either lower- or upper-case letters. Hold either SHIFT key down or activate the CAPS LOCK mode to enter uppercase letters.

NOTE: The SHIFT KEY reverses the action of the CAPS LOCK key. If the CAPS LOCK feature mode is active, normally producing uppercase characters, the SHIFT key will produce lowercase characters. If the CAPS LOCK mode is not active, the SHIFT key will produce uppercase characters.

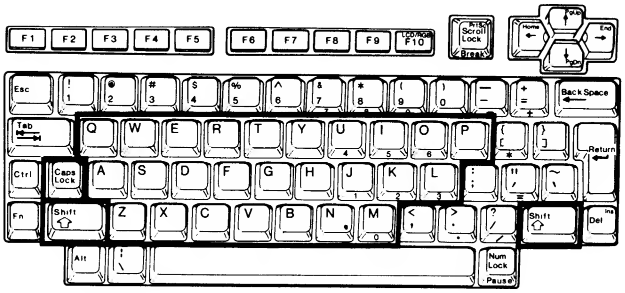
F1 F2 F3 F4 F5 F6 F7 F8 F9 LCD/RGB F10 PrtSc Scroll Lock Break Home End PgUp PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Back Space
Tab Q W E R T Y U I O P { [}] Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ \
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt | \ Num Lock Pause

Figure 7-4. Alphabetic Keys

Nonalphabetic Keys

The nonalphabetic keys are those outlined in Figure 7-5. They contain the numbers 0 through 9, punctuation marks, and special characters. The lower character is generated normally. If either or both SHIFT keys are pressed, the upper character will be generated. The CAPS LOCK mode does not affect the operation of these keys.

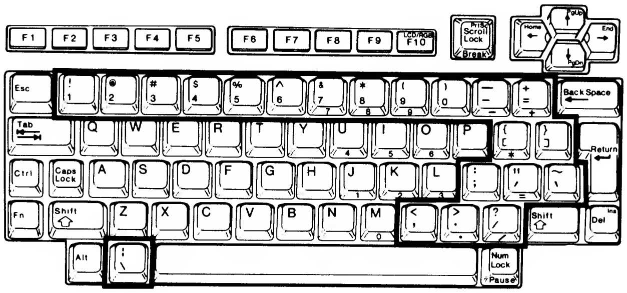
F1 F2 F3 F4 F5 F6 F7 F8 F9 LCD/RGB F10 PrtSc Scroll Lock Break Home End PgUp PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Back Space
Tab Q W E R T Y U I O P { [}] Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ \
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt | \ Num Lock Pause

Figure 7-5. Nonalphabetic Keys

Control Keys

A number of keys on the keyboard provide control over the computer, the screen, and the keyboard. The following discussion describes the normal function of each key. However, through software you can direct almost any key to cause a defined function to occur. Any key that performs a special function is usually described in the documentation that accompanies the software.

Refer to Figure 7-6 for the following discussion.

- SPACE BAR — Enters a blank character (space).
- BACK SPACE — Moves the cursor one space to the left. Software usually erases the character. The cursor is an indicator on the display that lets you know where the next key will appear. Software controls the shape of the cursor.

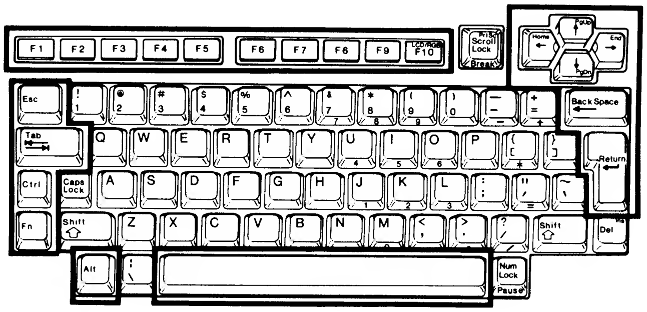
F1 F2 F3 F4 F5 F6 F7 F6 F9 LCD/RGB F10 PrtSc Scroll Lock Break Home End PgUp PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Back Space
Tab Q W E R T Y U I O P { [}] Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ \
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt | \ Num Lock Pause

Figure 7-6. Control Keys

- TAB — Moves the cursor to the next tab column, which is set to every eighth column by the system. Most software programs, especially word processing programs, modify this value.

- RETURN — Returns the cursor to the left side of the display. In most cases, software will add a line feed, so this key will move the cursor to the beginning of the next line on the screen. This key is also used to tell the computer that data or instructions have been entered and may be processed.

- Function keys (F1 through F10) — Used for special purposes by some programs.

- Cursor movement keys — Used to move the cursor up, down, left, or right by some programs.

- ESC — The escape key performs special functions when you press it and another key in sequence; that is, one followed by the other. The ESC key does not have to be held down once it has been pressed.

- CTRL — The control key performs special functions when you press and hold it down while you press one or more other keys. Most of the time, you will see instructions to use this key expressed as a combination of keys. For example, CTRL-C tells you to press and hold the CTRL key while you press the C key.

  NOTE: Even though other computer publications discuss pressing one or more keys simultaneously, the key combination actually indicates pressing and holding down those keys in the sequence in which they are printed. Once you have pressed all the keys they may be simultaneously released.

- ALT — The alternate key acts as a second CTRL key, providing additional code capabilities to many keys on the keyboard. It is used the same way; press and hold the **ALT** key while you press the other keys in the combination.

- FN — The function key acts similar to the CTRL key, providing additional codes when you hold it down and press another key. The specific functions performed by this key are unique to the hardware of the Laptop computer and are discussed later in this chapter.

Special Purpose Keys

Refer to Figure 7-7. The following keys are used by the computer to perform unique functions on the Laptop computer.

- CAPS LOCK — This key toggles the CAPS LOCK mode, which, when active, causes the alphabetic keys to produce uppercase letters, and when not active, to produce lowercase characters. The SHIFT keys reverse the action of the CAPS LOCK mode.

- NUM LOCK — This key serves two functions. When it is used with the FN key (FN-NUM LOCK), it toggles the keyboard lock mode. When the keyboard lock mode is active, it toggles the keypad between the numeric mode and the cursor/screen control mode. These modes are discussed later in this chapter.

- SCROLL LOCK — This key does not affect the operation of the computer, except under software control. Its function varies from package to package and is documented where it is used.

- DEL — This key is used by most text editing and word processing packages. It is also used in conjunction with the CTRL and ALT keys to reset the computer.

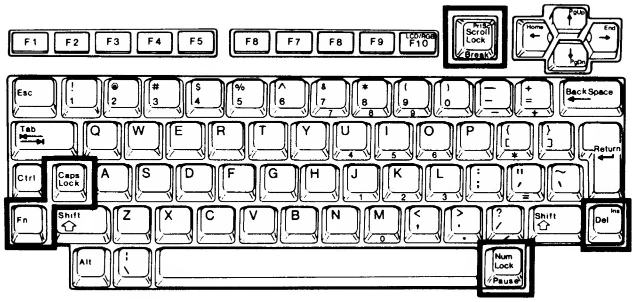
F1 F2 F3 F4 F5 F6 F7 F8 F9 LCD/RGB F10
PrtSc Scroll Lock Break
Home PgUp End PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = BackSpace
Tab Q W E R T Y U I O P { [}] Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ `
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt | \ Num Lock Pause

Figure 7-7. Special Purpose Keys

Key Label Colors

The key labels are printed in three different colors to indicate what other key must be pressed or mode of operation must be active to obtain the desired keyboard code. The codes are described in detail in Chapter 10.

The black labels on the tops of the keys represent the standard operation. The black labels on the fronts of the keys represent operation in combination with the CTRL (control) key. For example, to obtain the the CTRL-BREAK operation, press and hold the CTRL key and then press the SCROLL LOCK key, which has the BREAK label printed on its front. The CTRL key also works with most other keys on the keyboard, producing standard control codes.

The blue labels on the tops of the keys represent operation in combination with the FN (function) key. The operation of the FN key is similar to the CTRL key. For example, to obtain the INS code, press and hold the FN key and then press the DEL key, which has the INS label printed in blue on its top.

The orange labels on the fronts of the keys represent operation in combination with the FN key or when the keypad lock mode is active. For example, to obtain the keypad 4 key, press and hold the FN key and then the orange 4 key, or press FN-NUM LOCK to make the keypad lock key active, and then press the orange 4 key.

Keyboard Operation

The keyboard operates in a number of different modes. Four modes affect what codes are generated by the keyboard: unshifted keyboard mode, CAPS LOCK mode, keypad locked numeric mode, and keypad locked cursor/screen control mode. Refer to Figure 7-8. If you wish to test the operation of the keyboard, use the keyboard test described in Chapter 17.

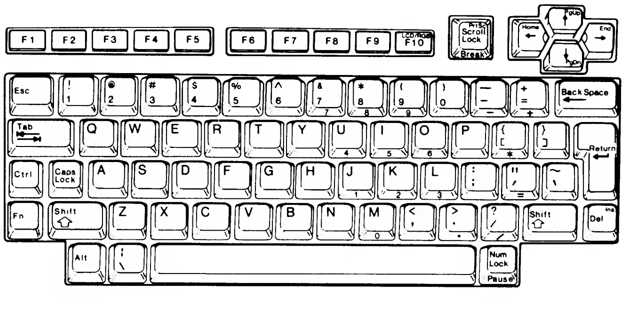

Figure 7-8. Laptop Computer Keyboard

The unshifted keyboard mode generates lowercase alphanumeric codes for the letters printed in black on the tops of the keys. The SHIFT keys cause uppercase letters and the upper character on the numeric/punctuation keys to be generated. The FN key causes the keypad characters (those printed in orange) and the functions printed in blue to be generated.

The CAPS LOCK mode generates uppercase alphabetic codes for the alphabetic keys and the lower character on the numeric/punctuation keys. The SHIFT keys cause lowercase letters and the upper character on the numeric/punctuation keys to be generated. The FN key causes the keypad characters (those printed in orange) and the functions printed in blue to be generated.

The FN-NUM LOCK key combination toggles the keypad lock on and off. The first time the keypad lock mode is activated, the numeric mode is active. When the keypad lock is active, the FN key causes the keypad keys to generate keyboard mode characters; however, the codes generated are dependent upon the state of the CAPS LOCK mode. Figure 7-9 illustrates the keypad characters that are activated by the keypad lock.

While the keypad lock is active, the NUM LOCK key toggles the keypad between the numeric and the cursor/screen control modes. The numeric keys are printed in orange. These codes are different than those generated by similar keys in the unshifted keyboard mode. The codes are listed in Chapter 10.

The SHIFT keys reverse the action of the NUM LOCK key. If the keypad is in the numeric mode, the SHIFT keys generate cursor/screen control codes. If the keypad is in the cursor/screen control mode, the SHIFT keys generate numeric codes.

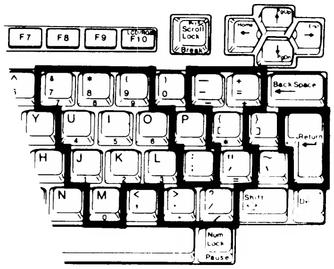

Figure 7-9. Laptop Computer's Keypad

Disk Drives

This computer will support up to two 3.5-inch disk drives, two 5.25-inch disk drives, and a hard disk drive. These drives are identified by most operating systems, such as MS-DOS, as drive A through drive H. The actual drive name assignment is determined by the settings of configuration switch sections 1, 5, and 6 (refer to Table 4-1). With switch section 1 ON and sections 5 and 6 OFF drive names A and B are reserved for the two 3.5-inch disk drives that can be mounted in the computer. Drive names C and D are reserved for the two 5.25-inch disk drives that can be connected externally. Setting switch section 1 OFF reverses drives A and C, that is, the first internal 3.5-inch drive is drive C and the first external 5.25-inch drive is drive A. And disk names E through H are reserved for the four partitions that can be created in a hard disk drive system that can be connected externally. In dual drive systems (refer to Figure 7-11) the left drive is identified as drive A and the right drive as drive B. If only one drive is installed in the computer, it is drive A.

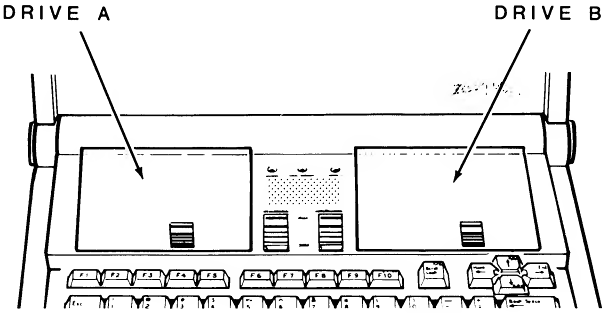

Figure 7-11. Disk Drive Positions

The 3.5-inch disk drives used in the computer are double-sided (each side has one read/write head), double-density disk drives capable of recording up to 8717 bits per inch (using MFM data encoding). Eighty soft-sectored, 2-track cylinders are recorded at 135 tpi (tracks per inch). The disk rotates at 300 rpm and storage capacity is 720K of information.

The external 5.25-inch disk drives used with the computer are double-sided, double-density disk drives capable of recording up to 5977 bits per inch. Forty soft-sectored, 2-track cylinders are recorded at 48 tpi. The disk speed is 300 rpm and storage capacity is 360K of information.

The disk controller and operating system can also support 96 tpi, 80-track double-sided 5.25-inch disk drives that rotate at 300 rpm and store up to 960K of information, and 96 tpi, 80-track high density 5.25-inch disk drives that rotate at 360 rpm and store up to 1.2 megabytes of infoımation on a disk.

The external rigid disk drive contains its own controller and is subject to the limitations of the controller and the operating system used. Since a number of different rigid disk drive types can be used in a system that can be attached to this computer, they are not described here.

Each read/write head typically consists of three ferrite cores. Each core is shaped similar to a split ring, one of which is the read/write core (refer to Figure 7-16 later in this chapter). Erase cores on both sides of the read/write core erase the space between the tracks, providing tunnel erase and a consistently low signal-to-noise level.

The heads, between which the disk rotates, are spring-mounted and face each other. When you move the computer and/or external drives, place the special shipping inserts that are supplied with the disk drives into the drives and close them. Otherwise, the heads could damage each other through vibration or jarring.

Disks

The computer uses 3.5-inch disks in the built-in drives and 5.25-inch disks in the external disk drives.

Refer to Figure 7-12. Four parts make up the 3.5-inch disk: the disk, the disk liner, the disk case, and a sliding metal cover.

The disk liner cleans the disk and traps dust particles. The disk case has three openings, one of which is under the sliding metal cover. The disk drive spindle grips the disk's metal drive hub, which is designed so that the drive-to-disk relationship is fixed. When the disk is placed in the disk drive, the metal cover is moved to one side, allowing the drive's read/write heads to come in contact with the disk.

At one corner of the disk is a small hole and movable tab. When the hole is exposed, the disk is write protected. When the hole is covered, data can be written on the disk.

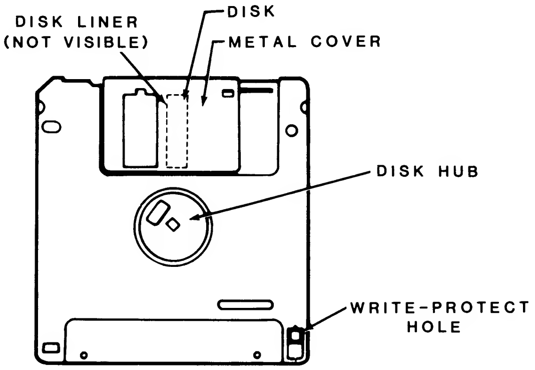

Figure 7-12. A 3.5-Inch Disk

Refer to Figure 7-13. Three parts make up the 5.25-inch disk: the disk, the disk liner, and the disk jacket. A disk envelope is provided for additional protection when the disk is not in the disk drive.

The disk liner cleans the disk and traps dust particles. The outer jacket has three openings and a notch in one side. The disk drive spindle grips the disk through the large center opening. The small circular hole is used for timing and the read/write heads access the disk through the long slot.

The notch on one side of the disk is used for write protection. When it is covered, the disk is write protected. When it is exposed, data can be recorded (written) on the disk.

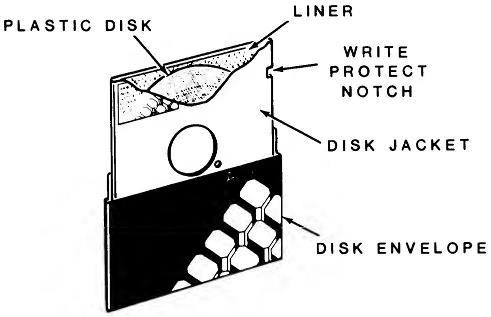

Figure 7-13. A 5.25-Inch Disk

The disk itself is similar to audio and video tape, except it is shaped differently. The base is a very thin plastic material that is flexible and coated with a magnetic substance. Since the disk is thin, it can be permanently distorted, rendering it unusable. Therefore, keep the following in mind when you handle disks.

When you prepare a label for a disk, write on it before you place it on the disk. If you must write on a label that is already on the disk, use a felt-tip pen. A disk should be stored in its protective envelope or a container in an upright position and away from heat or direct sunlight when it is not in use. Do not bend the disk or use paper clips on it. Do not touch any exposed areas of the disk or attempt to open the metal shield on the 3.5-inch disk. Keep the disk away from magnets and magnetized objects, including power supply adapters and telephones.

Disk Organization

Refer to Figure 7-14. A disk's surface is divided into cylinders, tracks and sectors. A track is that portion of the disk that passes under the read/write head. In double-sided disk drives, there are two read/write heads, one for each side of the disk. Therefore, each head has a track and these two tracks make up a cylinder. A cylinder has as many tracks as there are heads in any particular disk system.

Each track is divided into sectors by software. The software looks for either a timing hole placed in the disk itself (as is the case in 5.25-inch disks), the position of the drive mechanism (as is the case in 3.5-inch and rigid disk drives). The amount of information each sector can hold determines the disk's density. The Laptop computer uses double-density disk drives that can record either single-density or double-density data.

Other factors, such as track density, contribute to the total amount of data that can be stored on a disk. For example, 5.25-inch disks use either 40 tracks, recorded at a density of 48 tpi, or 80 tracks, recorded at 96 tpi. The 3.5-inch disks use 80 tracks, recorded at a density of 135 tpi.

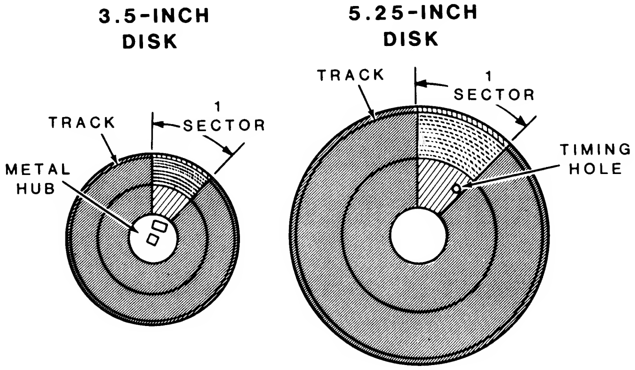

Figure 7-14. The Disk Surface

Data Encoding Methods

Data can be encoded in one of three ways on the disk, although this computer uses only two of these methods. FM encoding is used for single-density recording and MFM encoding is used for double-density recording. MMFM encoding, which is not used in PC-compatible computers, is also used for double-density recording.

In the figure, which is divided into three sections, the video signal itself is not illustrated. The top section illustrates the timing that makes up one raster-scan line. During one character clock cycle, the video signal will transmit 8 pixel locations to the screen. This will continue until the number of character clock cycles equal the number of characters to be displayed in one horizontal line. The character clock will continue to cycle while the beam in the CRT is returned to the left side of the screen (this is called the horizontal retrace period). From that point the character clock will continue to cycle until the total number of horizontal character positions has been reached, from which point the process will start over again, this time with the next row of pixels being sent to the screen. The video controller contains registers that hold the number of horizontal characters to be displayed, the time (in characters for horizontal retrace), and the total number of character positions in one line. Refer to Chapter 14 for more information on the video controller.

The second section of the figure illustrates the timing needed for one character line. The character clock and position are illustrated on the first two lines of this section. The third line, the raster line, illustrates the time required for one raster-scan line, as illustrated in the first section of this figure. The raster lines continue to be scanned until the maximum raster address for a character, which is stored in a register of the video controller, is reached. This value is usually 8, representing the 8 rows of pixels that make up a character frame. Since this value can be reprogrammed, it is possible to display characters that contain more than 8 rows of pixels. For instance, video mode 7, which is not supported in the Laptop computer, supports characters that are 9 pixels wide and 14 rows deep.

The third section of the figure illustrates the relationship between one character line and the entire screen, called a frame. In this section, the top line illustrates the characters line, one of which is illustrated in the second section of this figure. After a predetermined number of lines have been displayed, additional blank lines are sent to the display to fill the screen, both at the bottom and the top. During this time frame, the scan returns to the left and top of the screen.

Since each pixel may be selectively activated, individual alphanumeric and graphics characters or images that take part or all of the display's area are produced. The video memory of the computer, which determines whether pixels are activated or not, is addressed in the same manner that scanning takes place, that is, in a left-to-right, top-to-bottom pattern, whether an LCD or CRT display is used.

CRT Operation

The scanning of a typical CRT is illustrated in Figure 7-18. The display area is that area of the CRT where data is displayed and is 640 pixels wide by 200 pixels high. Electronic circuits in the monitor supply the necessary operating voltages to fire a beam of electrons at phosphors that are deposited on the inside face of the CRT. The beam, as it strikes the phosphors, agitates or excites them, producing visible light for each pixel. The color of the phosphor determines the color of the pixel. In monochrome monitors and televisions there is only one type of phosphor and only one electron beam. The intensity of the beam determines the amount of light given off by each pixel, giving the monitor the ability to produce different shades of light, called a gray scale.

In color monitors and televisions, three beams are fired at three differently colored phosphors that are deposited on the inside face of the CRT, producing red, green, and blue light. By firing one, two, or three beams at the different phosphors at any one pixel location, and in varying intensities, the monitor can display the different colors you see on a color CRT.

In the illustration, the border area is that area outside the display area that is scanned by the beam of electrons. Since the phosphor coating does not extend fully to the edge of the tube, any areas that are not coated with phosphors are illuminated by the electron beam.

However, scanning the CRT actually exceeds the area coated by phosphors, and, for that matter, often exceeds the width and height of the CRT. This overscan area, which is shown in the illustration, allows the full screen of the CRT to be used for display purposes. The beginning and end of the synchronization pulses, which are illustrated in Figure 7-17 as horizontal sync (in the first section) and vertical sync (in the third section), cannot be observed, because this takes place outside the overscan area. These pulses establish signal blanking, wherein the beam of electrons is either turned off or reduced to the point where it no longer excites the phosphors on the face of the CRT.

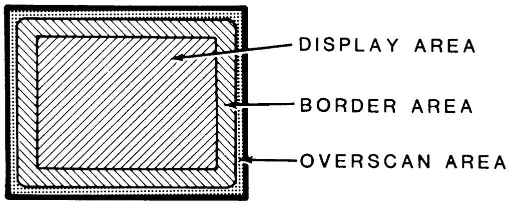

Figure 7-18. Scanning the Face of the CRT

LCD Operation

Liquid crystal displays do not emit light. Instead, they depend upon two ways to produce a readable display: reflected light or light sent through the LCD. The majority of clocks, including watches, that have LCDs use reflected light to produce a display. Even at night, when you press the buttons on your watch to light its display, the light is reflected through the LCD. Some newer LCD displays, send light through the LCD. By selectively blocking this light, whether it is reflected or transmitted, and image can be produced on the LCD.

One of the materials used in liquid crystal displays is held between a liquid and crystalline (solid) state through a wide temperature range (32 — 140 degrees Fahrenheit or 0 — 60 degrees Celsius). When the material is electrically charged, the electrons in the material align themselves so that light can pass through the material. When they are not aligned, or polarized, the material is opaque and light cannot pass through it. By allowing light to pass through the material while in a transparent state (charged), the opaque (not charged) areas become visible to the viewer. High contrast and a wide viewing angle is obtained by using high polarization angles and a very fine grid. An optional backlight makes it possible to use the display under all lighting conditions.

In this computer, the liquid crystal material is encased between two transparent plates. A conductive coating is etched on the inside of one of the plates in the form of a matrix 640 columns wide by 200 rows deep. This matrix is driven by row and column drivers which can be turned on to charge the liquid crystal material and make it transparent (lighting a pixel) or turned off (leaving the pixel opaque). The drivers are controlled by the video controller which addresses (in effect, scans) the drivers by column and row, starting at the uppermost and leftmost point on the screen, moving to the right until the end of the row is reached, returning to the beginning of the next row, and so on, until the entire screen has been "scanned." Unless the state of the pixel in video memory has changed, the state of the driver for that pixel is not changed. Also, during the border scan, overscan, and retrace times, the states of the drivers are not changed. The alignment of the material for individual pixels can be precisely controlled so that the pixel can be turned partially on, producing a gray scale display.

Modem Operation

Modems are a combination of hardware and software that let computers communicate with each other over telephone lines. Using the built-in modem, your computer can call data bases, electronic bulletin boards, and other personal computers. This section of the manual is designed to instruct users how to operate the modem.

If you are using the modem with communications software, you will need to refer to that software's user's manual to set up and operate the modem. In other cases, you will operate the modem using the AT commands and S-registers explained in the following pages.

There are two possible modems (1200 or 2400 baud) that may be installed depending on the model of computer. Commands for both modems are explained in this section of the manual. In cases where the information is identical for both modems, it is presented together. If there is a difference in a command or instruction, it is labeled (1200 only) or (2400 only).

The following discussion also includes instructions on how to turn the modem on, actually dialing a phone number using the AT commands, and testing the modem to verify correct operation.

Turning the Modem On

When your computer is turned on, the internal modem is deallocated (turned off) from the system. In order to use the internal modem you must use the DOS-based utility MODE.COM supplied on the MS-DOS distribution disk to turn the internal modem on and off.

COMMAND	RESULT
`MODE modem on`	Allocates (turns on) the internal modem.
`MODE modem off`	Deallocates (turns off) the internal modem.

Table 7-6. Additional 2400 Modem Commands

COMMAND	FUNCTION	
X	Enables optional call monitoring features and result codes:	
	X0	result codes 0-4
	X1	result codes 0-5, 10 (default)
	X2	result codes 0-6, 10
	X3	result codes 0-5, 7, and 10
	X4	result codes 0-7, 10
&G	0	no guard tone (default)
	1	550 Hz guard tone
	2	1800 Hz guard tone
&P	Pulse dial make/break ratio:	
	0	39/61 (default)
	1	33/67

Result Codes

After the modem is issued a command, it returns one of the result codes listed in Table 7-7. Result codes 0 through 5 are the basic result code set enabled at powerup by default.

Result codes 6 through 8 are the extended result code set and must be enabled specifically.

Table 7-7. Result Codes

DIGIT CODE	WORD CODE	DESCRIPTION
0	OK	The command was executed.
1	CONNECT	The modem was connected to another modem at 0-300 bps.
2	RING	A ringing signal was detected by the modem.
3	NO CARRIER	The modem tried to make a connection but couldn't or a connection was made and lost.
4	ERROR	One of several possible errors: command not recognized by the modem, the command line exceeds 40 characters, invalid character format at 1200 bps, or invalid checksum.

Table 7-7 (cont'd). Result Codes

DIGIT CODE	WORD CODE	DESCRIPTION
5	CONNECT 1200	The modem connected to a modem at 1200 bps.
6	NO DIALTONE	The modem does not detect a dial tone and will not continue processing the command.
7	BUSY	The modem dialed the number and detected a busy signal.
8	NO ANSWER	The modem waited for silence when dialing a number and did not detect it. Enabled by the @ dial modifier.
10	CONNECT 2400	(2400 modem only) connection established at 2400.

Command Buffer

After pressing the RETURN key, the command line is placed in the command buffer and then executed. This command line stays in the buffer until cleared when another AT command is entered. Entering only AT clears the command buffer and prepares it to receive a new command line.

The command buffer can contain up to 40 characters. Punctuation is included in the 40-character limit but spaces, the attention code (AT), and the RETURN key are not. Commands of more than 40 characters receive an ERROR message.

Commands are not loaded into the command buffer or executed while the phone is ringing. Commands are recognized and executed between rings.

S-Registers

S-registers store operating instructions for the modem such as timing parameters, counters, and the ASCII values for frequently used characters. The modem contains 17 eight-bit registers (S0-S16).

Each register is assigned a default value at powerup and when the modem is reset. The values assigned to the S-registers can be read or changed.

The Z command restores all S-registers to their default settings.

Reading an S-Register Value (Sr?)

To determine the value currently assigned to an S-register, enter **ATSr?** where r is the register number.

Example: **ATS4?** (reads the value of S-register 4)

The value of the register is then displayed on the screen in decimal notation.

010 (the current ASCII value of the line feed character)

The values of several S-registers can be read with one command line.

Example: **ATS0?S7?**

The modem will respond:

```
001
030
OK
```

Changing an S-Register Value (Sr = n)

To change the value of an S-register with the modem commands, use the Sr = n command, where r is the S-register number and n is the value to be written into the register. Valid values of r are 0 through 16. The range of valid values for n depend on the individual S-register.

Table 7-8. Register Summary

REGISTER	RANGE	UNIT	FUNCTION	DEFAULT
S0	0-255	rings	Number of rings before answer	0
S1	0-255	rings	Number of rings occurred	0
S2	0-127	ASCII	Escape code character	43
S3	0-127	ASCII	Carriage return character	13
S4	0-127	ASCII	Line feed character	10
S5	0-32, 127	ASCII	Back space character	8
S6	0-255	seconds	Wait before dialing	2
S7	1-255	seconds	Wait time for carrier	30
S8	0-255	seconds	Length of pause caused by,	2
S9	1-255	.1 second	Carrier detect response time	6
S10	1-255	.1 second	Delay between lost carrier and hanging up the telephone	7
S11	20-255	millisec	Duration and spacing of touch-tones	70
S12	0-255	.02 second	Escape code guard time	50
S13	——	——	Bit-mapped register	—
	Bit 0	reserved		
	Bit 1	reserved		
	Bit 2	0 = parity disabled 1 = parity enabled		
	Bit 3	0 = odd parity 1 = even parity		
	Bit 4	0 = 7 data bits 1 = 8 data bits		
	Bit 5	reserved		
	Bit 6	1 = buffer overflow flag (causes the ERROR result code to be sent)		
	Bit 7	0 = parity bit set to space (if bit 4 = 0) 1 = parity bit set to mark (if bit 4 = 0)		

NOTE: Writing to S13 may produce unpredictable results.

Table 7-8 (continued). Register Summary

REGISTER	RANGE	UNIT	FUNCTION		DEFAULT
S14	——	——	Sets the product code, returned when ATI0 is issued.		—
S15	——	——	Bit-mapped register		—
	Bit 0	same as setting of bit 4			
	Bit 1	same as setting of bit 5			
	Bit 2	0 = answer 1 = originate			
	Bit 3	0 = half-duplex 1 = full-duplex			
	Bits 4 and 5 constitute the bps rate code:				
		Bit 5	Bit 4	bps rate	
		0	0	undefined	
		0	1	110	
		1	0	300	
		1	1	1200	
	Bit 6	0 = carrier OFF 1 = carrier ON			
	Bit 7	reserved			
	NOTE: Writing to S15 may produce unpredictable results.				
S16		——			0
	0	Normal mode (non-test)			
	1	Analog loopback			
	2	DTMF tone test			
	4	Test Pattern			
	5	Analog loopback with test pattern			

Testing the Modem

When errors in data transmission happen, you can test the performance of the internal modem to determine if it is the source of the errors.

Begin by turning the modem on as described earlier in this chapter.

When the modem is first turned on, the local echo is ON. If your typed command in not displayed, your local echo is OFF. To turn the local echo ON, enter the following command:

```
ATE1
```

and press **RETURN**.

If double characters appear on the screen, both your modem and software are set to local echo ON. Either set your software to local echo OFF, or turn the modem's echo OFF with the command:

```
ATE0
```

and press **RETURN**.

Checking the Product Code, ROM Checksum, and Internal Memory

The I command can be used to check the modem's product identification code, ROM checksum, and status of the internal memory.

To display the modem product code, use the command ATI0. The modem issues a three-digit number representing its product number and firmware revision number. The two left digits represent the product. The right digit is the firmware revision number.

To check the sum of the modem's ROM bytes, type ATI1. The modem returns three ASCII characters followed by a carriage return and line feed. This is primarlly a test done at the factory.

To test the modem's RAM memory, use the command ATI2. The modem calculates the bytes in the RAM and compares the total against the correct sum, stored in ROM. If the two figures agree, the internal memory is complete, and the OK result code is displayed. If the figures do not agree, the ERROR result code is displayed.

Analog Loopback Self-Test

The analog loopback self-test allows you to test the internal modem. During this test the modem modulates and demodulates its internal test pattern and sends the pattern to the computer for display.

To begin the test, set S-register 16 to 5 followed by the dial command D. These two commands may be combined:

```
ATS16=5D
```

The modem will go "off hook" (the equivalent of picking up a phone receiver), send the message CONNECT 1200 or 2400 to the screen, and then send the test pattern.

To end the test, press any character key on the keyboard. The modem goes back "on hook" (the equivalent of hanging up the phone) and displays the message NO CARRIER.

Reset S-register 16 back to 0 (ATS16 = 0).

During normal operations, the internal modem communicates at two frequencies within the communication channel, one low and the other high. The dial command, D, in the preceding test, caused the modem to test the originate (low) frequency. To test the modem at the answer (high) frequency, substitute the answer command, A, as follows:

```
ATS16=5A
```

Analog Loopback Test

The analog loopback test allows you to test the internal modem by entering data at the keyboard and verifying it on the display. This test can also be at both the low and high frequencies. Enter either of the following commands:

```
ATS16=1D
```

or

```
ATS16=1A
```

The modem goes "off hook" and displays the message CONNECT 1200 or 2400.

Type any message at the keyboard. It is looped through the modem and returned to the display.

End the test by not typing anything for at least one second and then entering the escape code + + + followed by another one-second pause. This causes the modem to go "on hook" and return to the command state. The OK result code is displayed.

Reset S-register 16 to 0 (ATS16 = 0).

The Test Pattern

The test pattern allows you to test the internal modem, the remote modem, and the telephone line. During this test, when a dial command is issued, the modem transmits the test pattern upon connection with the remote system.

To begin the test, set S-register 16 to 4, and issue the dial command followed by a telephone number. These commands may be combined:

ATS16=4D*n* (*n* represents a telephone number)

To end the test, press any character key on the keyboard. The modem goes back "on hook" (the equivalent of hanging up the phone) and displays the message NO CARRIER.

Reset S-register 16 back to 0 (ATS16 = 0).

If S-register 16 is set to 4 and the modem is set for auto answer, the modem will transmit the test pattern when it answers a call.

Dial Test

The dial test allows you to test the frequencies of the touch-tone values. during this test, when a single tone value (such as 7) is dialed, the modem continues to transmit that tone until the RETURN key is pressed.

To begin the test, set S-register 16 to 2 and issue a dial command.

ATS16=2DT*n* (T is the command for touch-tone dialing and *n* is a number from 0 to 9)

The modem will transmit the tone value for that number. Pressing the RETURN key will end the tone transmission. Another dial command can be issued to test other tone values. When finished, reset S-register 16 to 0 (ATS16 = 0).

Peripherals

A peripheral is any device that can be attached to the computer. This includes disk drives, which have already been discussed in this chapter, as well as printers, plotters, and other serial and parallel input and output devices.

This section of the manual will guide you throuçh the information necessary for the MS-DOS CONFIGUR program, which is designed to help you configure your computer's serial and parallel port for operation with almost any peripheral that can be attached to it.

You will probably want to get your peripherals up and running as quickly as possible. Therefore, the amount of information presented in this section is limited. For full details on the CONFIGUR utility, refer to your MS-DOS documentation.

Before you start the CONFIGUR program, you must know the following about your peripheral.

NOTE: Some of the following material will make reference to handshaking on specific pins. These pin numbers refer to the connector at the computer's serial output port. If your device requires handshaking on a different pin, you will have to obtain or make a cable that moves the signals generated by the computer to those pin numbers required by your peripheral device.

- Is it a parallel or serial device? If it is a parallel device, then you do not need any other information; otherwise, the following question needs to be answered.

- Is it one of the following serial devices? If so, then you do not need any other information; otherwise, the remainder of the questions need to be answered.

  — An Epson MX-80 or similar serial device with the following characteristics: 4800 baud, DTR positive handshaking on pin 20.

  — A Heath/Zenith H-25, H-125, Z-25, or Z-125, or similar serial device with the following characteristics: 4800 baud, RTS positive handshaking on pin 4.

  — A Heath H-14 or Zenith WH-24, or similar serial device with the following characteristics: 4800 baud, RTS negative handshaking on pin 4.

  — A Diablo 630 or 1640 or similar serial device with the following characteristics: 1200 baud, ETX/ACK handshaking.

  — A Heath/Zenith WH-23, WH-33, or WH-43 modem, or similar serial device with the following characteristics: 300 baud, no handshaking.

  — A Heath WH-12 Votrax Type-N-Talk or similar serial device with the following characteristics: 4800 baud, RTS positive handshaking on pin 4.

- Do you need parity stripped on input from the device?

- Do you need parity stripped on output to the device?

- Do you need lowercase characters converted to uppercase characters on input from the device?

- Do you need lowercase characters converted to uppercase characters on output to the device?

- What baud rate do you need? The following are available: 110, 150, 300, 600, 1200, 2400, 4800, and 9600 baud.

- How many stop bits (1 or 2) does the device need?

- What parity (odd, even, or none) is required?

- How many bits (7 or 8) make up a word?

- What kind of handshaking is required? The following are available:

  — ETX/ACK.

  — DC1/DC3.

  — Compatibility mode, DTR positive on pin 20 and RTS positive on pin 4.

  — RTS positive on pin 4.

  — RTS negative on pin 4.

  — DTR positive on pin 20.

  — DTR negative on pin 20.

  — No handshaking.

- Is a pad character required after a carriage return and, if so, what is that pad character and how many should be transmitted?

Chapter 8

The Monitor Program and Programming with Interrupts

To maintain a high level of software compatibility with PC-compatible computers, this computer uses firmware-established interrupts for most control and input/output functions. This chapter provides you with an overview of MFM-180 (multifunction Monitor program) and its operation, along with a summary of the interrupts that are set up when the computer is first turned on. The chapters that follow provide you with explanations for each interrupt.

Diagnostic Routines

During the initial powerup sequence, the Monitor program performs a number of tests to make sure the computer is ready to function, including a self-test of all circuits in the system. When the tests are successfully finished, the computer will attempt to load sector 1 of disk track 0 into memory and execute it. This is the automatic boot procedure (autoboot).

When autoboot starts, if you are using the standard 3.5-inch disk drives, a disk must be placed in the drive and the door must be shut within 20 seconds or an error message will be displayed (see Chapter 17 for a list of error messages). If you have a rigid disk system with the MS-DOS operating system installed, the autoboot procedure will load the operating system and display the system prompt (C>).

NOTE: The drive letter in the prompt may differ, depending on the number of drives selected by the configuration switches. If the rigid disk does not have MS-DOS installed, the system will display a message indicating that the drive is not bootable. In this case you will want to access the Monitor program.

To reach the Monitor program, autoboot must be defeated or bypassed. To defeat autoboot, press the **ESC** key after the system is first turned on. The monitor prompt (–>) will be displayed. To reach the Monitor program after the operating system has been booted, press **CTRL-ALT-FN-INS**. The Monitor program opening message will be displayed. It will appear similar to the following.

```
MFM-180Monitor, version 2.0
Memory Size: 640K
Press "?" for help.
->
```

NOTE: The version number may vary from the one printed in this example. Most Laptop computers will contain 640K of RAM; those that do not will display the amount of memory present (128K or 384K).

User Commands

From this point, the Monitor program selections can be displayed on the screen. To display the monitor command summary, enter a question mark and press the **RETURN** key. A display similar to the one shown in Figure 8-1 will be displayed. Each command on the menu is discussed in the material that follows.

```
                    -MFM-180 Command Summary-

CMD:       Explanation            Syntax

?          Help                   ?
B          Boot from disk         B [{0 | 1 | 2 | 3}]
C          Color Bar              C
D          Display memory         D [<range>]
E          Examine memory         E <addr>
F          Fill Memory            F <range>,{<byte> | "<string>"}...
G          Execute (Go)           G [=<addr>] [,<breakpoint>]...
H          Hex math               H <number1>,<number2>
I          Input from port        I <port>
M          Move memory block      M <range>,<dest>
O          Output to port         O <port>,<value>
R          Examine Registers      R [<register>]
S          Search memory          S <range>, [<byte> | "<string>"}...
T          Trace program          T [<count>]
U          Unassemble program     U [<range>]
S          Set Video/Scroll       S [M<mode>] [S<scroll>]
TEST       Extended diagnostics   TEST
Where      <range> is:            <addr>{<addr> | L<length>}

           Copyright (C) 1986, by Zenith Data Systems
```

Figure 8-1. MFM-180 Command Summary Menu

The Monitor program has six basic features: automatic powerup self-tests (diagnostics), user-executed tests, video commands, machine language debugger, basic input/output system, and disk boot routines. This portion of the manual will address the last four topics. Information on the self-tests and the user-executed tests can be found in Chapter 17.

In this example, 16 bytes of memory will be displayed, starting at address 1000:0. The resulting display might appear as follows:

```
1000:0000 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
```

There would, of course, be 16 trailing periods, one for each data byte displayed. A second example entry might appear as follows:

```
D DS:7000
```

In this example, 128 bytes of memory will be displayed, starting at address location offset 7000H from the base address of the data segment, which is stored in CPU segment register DS. If the extension L200 were added to the above command, it would cause 200 hex bytes of data to be displayed following this segment address.

If you display a large range of memory, the information will begin to scroll off the screen. You may use the CTRL-S key combination to stop the display, and any other key to start it again.

Examine Memory

Syntax: E <address>

Example: `E 1000:100` *RETURN*

The examine memory command allows the user to examine or change one byte of memory. When executed, the command will display one byte of data and await user input. If a MINUS (or DASH) is entered the contents of the previous memory location will be displayed. If a SPACE is entered the contents of the next byte will be displayed. Otherwise, the entry of a hexadecimal number from 0 – FF will modify the contents of the current location. To exit from the examine option press the RETURN key.

Fill Memory

Syntax: F <start address>,<range>,<data list>

Example: `F 1800:0,1FF,"TEST",20,54,45,53,54,20` *RETURN*

The fill command allows you to enter data directly into memory. The entry may be made using ASCII characters, delineated by quote marks, or in hexadecimal format. You may enter one full line of data or text using the fill command. Each hexadecimal byte and text word must be separated by commas. Data provided in the list will be reused as often as necessary to fill the specified range. The starting address, range, and data must be specified for the command to function.

If you specify a range for this command, such as 1FF, this value will be interpreted as 1FF hex bytes, and that amount of memory will be written to by the debugger.

The fill command is extremely powerful in that all of system memory is available to the user. It is therefore possible to overwrite vital system files and programs causing a loss of data or actually halting the computer and causing it to cease operation. This is often referred to as a “system crash” or a “lock-up”. If this should happen to you, merely turn the computer off, wait a few seconds, and turn it on again. It is not possible to use the normal CTRL-ALT-FN-INS key sequence to return to the Monitor program from this condition.

GO (Execute)

Syntax: G[=<address>][,<breakpoint>]...

Example: `G 1000:0` *RETURN* or `G 100` *RETURN* or `G=1000 100` *RETURN*

The Go command transfers control from the Monitor program to a user program at a specified address with optional breakpoint settings. A maximum of eight breakpoints may be specified with the Go command.

The equal sign (=) is used to specify the starting address for the user program. This can be a specific offset within the current code segment, or a different segment and offset. For example, G 1000 would specify an offset of 1000 within the current code segment (specified by the CS register value). An entry of G=1000 would reset the base address of the CS register, and start program execution. In a similar fashion, G=1000:100 110 would reset the base address of the CS register and specify a program start offset of 100 within that segment with a breakpoint set at 110.

Breakpoints are programming debugging tools. If you are developing a program and want to check certain values within the program during execution, you use breakpoints. When the debugger encounters a breakpoint the Monitor program will save the processor status and display a register dump. This display gives the current status of the processor registers and flags as well as the current instruction that was executed. Even though a maximum of eight breakpoints may be set, the debugger will halt when it encounters one of them, and will not remember the other breakpoints. This somewhat limits your debugging capabilities to following the flow of the program, but it is still useful. Breakpoints must be set at addresses containing valid processor instruction codes, otherwise the system will crash.

Hex Math

Syntax: H <number1>,<number2>

Example: `H 1000,23FF` *RETURN*

This command provides the user with a very simple means to do base 16 (hexadecimal) computations. When executed the command will return the sum and the difference of the two numbers entered on the command line. This command will only do computations on two numbers in hexadecimal format. In this format, number2 is subtracted from number1.

Input From Port

Syntax: I <port address>

Example: `I 3F8` *RETURN*

The input command provides the user with a means of obtaining information concerning the status of the system hardware. The port addresses and the values returned are in hexadecimal format. Only one byte of information will be read from the specified port.

Move Memory Block

Syntax: M <start address>,<range>,<new address>

Example: `M 0:1000,107F,0:2000` *RETURN*

The move command is actually something of a misnomer. What the command actually does is copy a user specified block of memory to a user specified location. The word "move" implies a physical relocation and this is not what happens using the move command.

In order to use this command, the user must specify the starting address of the memory block, its length, and the starting address of the new location. If this information is not provided, the Monitor program will flag an error.

The move command, like the fill command discussed earlier, makes all system memory accessible to the user. It is possible to overwrite system and program files with the same results as before. When you are moving data within memory, be sure that the locations you have chosen will not cause problems.

Output To Port

Syntax: O <port address>,<data>

Example: `O 2F8,00` *RETURN*

Like the input command discussed earlier, the output command provides the user with some direct control of the system hardware. The port address and the data sent to the port are both in hexadecimal format.

Examine/Modify Registers

Syntax: R [<register>]

Example: `R` *RETURN* or `R CS` *RETURN*

The examine registers command is the only means that the debugger provides to alter the contents of the processor registers directly. The command also allows the user direct access to the processor status flags. This command provides the user with several options.

If the letter R is entered by itself, the resulting display will indicate all registers, the flags and the current instruction. This is the register dump discussed earlier. If instead the letter R is entered followed by a two letter register name, such as CS, the contents of that register alone will be displayed.

A display of a single register's contents is followed by a dash prompt. At this point the debugger is waiting for user input. You may change the register contents by entering a valid hexadecimal value followed by pressing the RETURN key. If you want to leave the register contents alone, merely press the **RETURN** key. When changing the contents of a register, if the hexadecimal value you supply is not valid, the contents will not be changed. If you want to change a processor status flag, the procedure is somewhat different.

The command R FL will display the current status of the processor flags and provide a dash prompt indicating that the debugger is awaiting user input. The processor makes use of eight different status flags, each with a specific two letter abbreviation. Table 8-3 explains the flags and the on and off codes.

The highest priority is assigned to IRQ0 (interrupt request 0) and the lowest to IRQ7. By sending a bit mask to port 021H, the programmer can control whether a particular interrupt can be processed when requested. For more information on the 82C59, refer to Chapter 14.

Table 8-4 describes the usual hardware requests and the software interrupts they trigger. The normal order of precedence is used in the table. Jumpers and/or other hardware factors may change the source of the interrupt request in some computer installations (not recommended).

Table 8-4. Hardware Generated Interrupt Requests

HARDWARE INTERRUPT REQUEST	SOFTWARE INTERRUPT	USUAL SOURCE OF INTERRUPT
0	08H	Time-of-day timer
1	09H	Key pressed
2	0AH	Not used
3	0BH	Modem (COM2)
4	0CH	Communications (COM1)
5	0DH	Hard disk drive
6	0EH	Floppy disk drive
7	0FH	Parallel printer (LPT1)

Using a Software Interrupt

Some of the interrupts discussed in this manual are initialized by the operating system. Since MS-DOS is the standard operating system for these computers and helps maintain PC-compatibility, those interrupts that are considered universal (that is, not specific to a particular version of MS-DOS) are included in the summary presented later in this chapter. Those based on MS-DOS that are subject to change between versions are discussed in the Programmer's Utility Pack for MS-DOS. Other operating systems and MS-DOS for other computers (non-PC-compatible) may or may not initialize the same or similar interrupts.

NOTE: Any software interrupt that has not been initialized will jump to a return (RET) function (no operation).

When you prepare to use or modify an interrupt routine, there are a couple of factors you should keep in mind.

- All parameters that are passed to and from interrupt routines must go through the CPU's registers. Where more than one function may be performed by a routine, or where additional parameters are required by a routine, more than one register will be used by the interrupt.
- Most of the interrupt routines will preserve the values in the CPU registers except when a value will be returned to a specific register. When you write or modify routines for these interrupts, you should plan on preserving, where possible, the values of the CPU registers.

Each interrupt routine is pointed to by a dedicated interrupt vector (address). To execute one of the interrupt routines, you must first load the specified parameters into the registers of the CPU. Then perform an INT *xx*H instruction where *xx*H is the interrupt number. For instance, the code example in Listing 9-1 executes the INT 11H interrupt and then tests for and returns the number of drives connected to the system as established by the configuration switches.

Since no parameters were required by INT 11H, none are loaded. Note however, that the results are returned in register AL, where they are tested by the routine.

Listing 8-1. Sample Code

```
                                ;test for number of drives
        INT     11H             ;retrieve status byte
        PUSH    AL              ;temporarily store results
        AN      DAL,1B          ;isolate bit 1
        TEST    AL,0            ;test for drives present
        JZ      ZERODRIVES      ;if no drives, jump to routine
        POP     AL              ;pop results back off stack
        AND     AL,1100000B     ;isolate bits 5 and 6
        MOV     CL,5            ;set for shift of 5 bit-places
        SHR     AL,CL           ;shift data right 5 bits
        INC     AL              ;increment results by 1 so
                                ;that true number of drives
                                ;is now in register AL
```

Modifying an Interrupt

To modify an interrupt, you must patch the interrupt vector with the address of your routine as outlined in the following procedure (MS-DOS environment). For more information, refer to the Programmer's Utility Pack for your version of MS-DOS.

1. Use the MS-DOS Function Request, Get Interrupt Vector (35H), to read the current value of the interrupt vector. Save this value in your code segment.

2. Use the MS-DOS Function Request, Set Interrupt Vector (25H), to set the address of your interrupt routine.

3. Do not execute an IRET instruction at the end of your routine. Instead, execute a JMP DWORD PTR to the address saved in step 1. This allows multiple background tasks to have a chance at the interrupt. Note that the final routine executed during an interrupt sequence is an IRET, but only after the CPU registers have been restored.

4. Do not modify any CPU registers unless specifically stated in the discussion of that interrupt. Upon entry to an interrupt routine, the ES, DS, BX, CX, DX, AX, BP, SI, and DI registers contain information that needs to be saved. Therefore, it is a good idea to PUSH the information onto the stack and then restore it following the execution of the your routine.

Software Interrupt Summary

Table 8-5 summarizes the interrupts and describes the function, the initializing system (the Monitor program MFM-180, DOS, or application software), and the chapter where you can find a complete discussion. Interrupts that show dashes (—) in each column are either reserved for future use or generally are not used by PC-compatible computers; more specifically, they are not implemented in Zenith Data Systems computers or software.

Following Table 8-5 are tables that organize the interrupts into logical groups: Table 8-6, System and CPU Interrupts; Table 8-7, Keyboard Interrupts; Table 8-8, Input/Output Interrupts; Table 8-9, Data Storage Interrupts; and Table 8-10, Video Interrupts. Each table is repeated in the chapter that discusses those interrupts.

Table 8-5. Interrupt Summary

	INITIALIZED		
INTERRUPT	BY	CHAPTER	FUNCTION
00H	DOS	10	Divide by zero
01H	DOS	10	Single step
02H	MFM-180	10	Nonmaskable interrupt
03H	DOS	10	Software breakpoint
04H	DOS	10	Arithmetic overflow
05H	MFM-180	12	Print screen
06H	—	—	
07H	—	—	
08H	MFM-180	10	Timer (time-of-day)
09H	MFM-180	11	Key pressed
0AH	MFM-180	10	Real-time clock
0BH	Software	12	Communications (COM2)
0CH	Software	12	Communications (COM1)
0DH	Software	12	Alternate parallel printer (LPT2)
0EH	MFM-180	13	Floppy disk drive
0FH	Software	12	Parallel printer (LPT1)
10H	MFM-180	14	Video input/output
11H	MFM-180	10	Equipment configuration
12H	MFM-180	10	Memory size
13H	MFM-180	13	Disk input/output
14H	MFM-180	12	Serial input/output
15H	—	—	
16H	MFM-180	11	Keyboard input/output
17H	MFM-180	12	Printer input/output
18H	MFM-180	12	Parallel/serial configuration
19H	MFM-180	13	Booting an operating system
1AH	MFM-180	10	Set/read the time of day
1BH	MFM-180	11	Keyboard break
1CH	MFM-180	10	Tick timer
1DH	MFM-180	14	Video initialization
1EH	MFM-180	13	Disk parameters
1FH	MFM-180	14	Defining characters

Table 8-6. System and CPU Interrupts

INTERRUPT	FUNCTION
00H	Divide by zero
01H	Single step
02H	Nonmaskable interrupt
03H	Software breakpoint
04H	Arithmetic overflow
08H	Timer (time-of-day)
0AH	Real-time clock
11H	Equipment configuration
12H	Memory size
1AH	Set/read the time of day
1CH	Tick timer

Table 8-7. Keyboard Interrupts

INTERRUPT	FUNCTION
09H	Key pressed
16H	Keyboard input/output
1BH	Keyboard break

Table 8-8. Input/Output Interrupts

INTERRUPT	FUNCTION
05H	Print screen
0BH	Communications (COM2)
0CH	Communications (COM1)
0DH	Alternate parallel printer (LPT2)
0FH	Parallel printer (LPT1)
14H	Serial input/output
17H	Printer input/output
18H	Parallel/serial configuration

Table 8-9. Floppy Disk Interrupts

INTERRUPT	FUNCTION
0EH	Floppy disk drive
13H	Disk input/output
19H	Booting an operating system
1EH	Disk parameters

Table 8-10. Video Interrupts

INTERRUPT	FUNCTION
10H	Video input/output
1DH	Video initialization
1FH	Defining characters

System Organization

Table 8-11 provides an overall map of the system's addressable memory. Table 8-12 is a general map of the system's ports.

NOTE: The information presented in these tables is subject to change as new products are introduced for this family of computers.

NOTE: This computer does not support a light pen port. Some computer games use this feature and may not operate correctly.

Table 8-11. System Memory Map

ADDRESS RANGE	DESCRIPTION
00000H – 9FFFFH	System memory. This is further divided by the Monitor program and DOS as follows.
(00000H – 003FFH)	Interrupt vector table (addresses).
(00400H – 004FFH)	Monitor program compatible data segment.
(00500H – ?????H)	IO.SYS* (part of the DOS)
(?????H – ?????H)	DOS.SYS* (part of the DOS)
(?????H – ?????H)	Resident portion of COMMAND.COM (part of the DOS)
(?????H – ?????H)	User-installed .COM and .EXE files
(?????H – ?????H)	Transient portion of COMMAND.COM
(?????H – 9FFFFH)	Open for general use
A0000H – AFFFFH	Reserved.
B0000H – B3FFFH	Monochrome graphics.
B4000H – B7FFFH	Reserved.
B8000H – BBFFFH	Color graphics.
F0000H – F3FFFH	Monitor program's scratchpad memory.
F4000H – F7FFFH	Rigid drive buffer.
F8000H – FFFFFH	System ROM (Monitor program).

NOTE: IO.SYS and DOS.SYS are representative of the names of hidden files that are part of MS-DOS. The actual names of these files can vary from version to version.

Table 8-12. System Port Map

PORT ADDRESSES	DESCRIPTION
000H – 00FH	DMA processor.
020H – 021H	Interrupt controller.
040H – 043H	System timer.
060H – 063H	Peripheral interface status port.
080H – 083H	DMA page registers.
0A0H – 0AFH	Nonmaskable interrupt registers.
0C0H – 0D7H	Real-time clock.
0E0H – 0EFH	Reserved.
200H – 20FH	Game input/output port (not implemented).
278H – 27FH	Reserved.
2F8H – 2FFH	Modem (COM2).
378H – 37FH	Parallel printer #1 (LPT1).
3B0H – 3BFH	Monochrome TTL video controller and/or parallel printer #2 (LPT2) (neither are implemented on this computer).
3D0H – 3DFH	Color graphics controller.
3F0H – 3F7H	Floppy disk controller.
3F8H – 3FFH	RS-232C serial input/output interface #1 (COM1).

Chapter 9

System and CPU Interrupts

Programming System and CPU Interrupts

Those interrupts that are generally considered to be system interrupts are defined in Table 9-1. For information pertaining to the use and programming of the software interrupts, see Chapter 8.

Table 9-1. System and CPU Interrupts

INTERRUPT	FUNCTION
00H	Divide by zero
01H	Single step
02H	Nonmaskable interrupt
03H	Software breakpoint
04H	Arithmetic overflow
08H	Timer (time-of-day)
0AH	Real-time clock.
11H	Equipment configuration
12H	Memory size
1AH	Set/read the time of day
1CH	Tick timer

Divide by Zero (INT 00H)

INT 00H (divide by zero) will be executed if a divide instruction produces a quotient that is too large to fit in the result register (such as dividing a value by 0). This routine is initialized by the operating system. The routine will print `Divide Overflow` and return control to the operating system.

You must set up a vector to intercept DIV and IDIV instructions if you do not want control returned to the operating system. That way, you can test for the error condition and prevent program control from returning to DOS. For instance, BASIC uses this method to retain control when a divide by zero condition occurs while executing BASIC functions.

Single Step (INT 01H)

INT 01H (single step) is used for executing a single machine instruction at a time. It is called by the CPU when an instruction is executed with the trace flag (TF) set.

It is commonly used by routines such as the MS-DOS DEBUG command and the Monitor program's trace command. Since this routine is initialized by the command (issued by the DOS or Monitor program) calling it, you must also initialize the routines you want executed when you call this interrupt.

Nonmaskable Interrupt (INT 02H)

INT 02H is the nonmaskable interrupt (NMI). It is initiated by hardware external to the 80C88. In most Zenith Data Systems computers, it is used to indicate that a power-down condition has started. Also, the 8087 Numeric Processor Extension uses this interrupt in its normal operation.

Normally, this interrupt is not disabled. One exception, however, is during the powerup sequence while the self-tests are being run. If you need to disable this interrupt (turn it off), send 00H to port A0H. To enable the NMI, send 80H to the same port (A0H).

Software Breakpoint (INT 03H)

INT 03H is the software breakpoint interrupt. When the processor encounters a breakpoint in a program, it will execute an INT 03H instruction, calling the interrupt routine.

The Monitor program's debugging routines and MS-DOS' DEBUG allow you to set breakpoints in machine language code. Then, when a breakpoint is encountered under the Monitor program or MS-DOS DEBUG, control is returned to the last command level.

Arithmetic Overflow (INT 04H)

INT 04H is the arithmetic overflow interrupt. It is executed by an INTO instruction (interrupt on overflow) when the overflow flag (OF) is set. The overflow flag is set by an arithmetic or logic operation.

Timer (Time-of-Day) (INT 08H)

INT 08H is the timer interrupt. The output of counter 1 of the 82C53 Programmable Interval Timer initiates this interrupt 18.2159 times per second, or every .054897095 seconds. This timer is not affected by the CPU clock speed and is used for such functions as keeping track of the time, timing out disk motors, and calling the timer tick interrupt (1CH).

In many computers the timer keeps track of the time-of-day in a 32-bit word (sometimes called a double-word). When the count reaches approximately 1803D8H (1,573,848 decimal), a flag is set to 1 to indicate that the timer has rolled past midnight to a new day. Interrupt 1AH is used to set and/or read the value of this word. This interrupt is established by the hardware as IRQ0.

In the lap top computer, the time of day is kept by a real-time clock IC, which is described in Chapter 14.

Real-Time Clock (INT 0AH)

INT 0AH is the real-time clock alarm interrupt. It is initiated by a hardware interrupt request (IRQ3) or when the alarm from the real-time clock IC takes place. The alarm is set and handled through INT 15H, function 83H (event wait) and function 86H (wait). The hardware interrupt request takes place approximately 1,024 times a second.

In most systems this interrupt is not used. Since this interrupt is essentially controlled by hardware unique to machines such as the Laptop computer, it should not be called by user programs.

Equipment Configuration (INT 11H)

INT 11H is the request equipment configuration interrupt. It is used to report the configuration of the equipment. The report is returned in a 16-bit word (register AX). Since this interrupt is common to PC-compatible equipment, you need to be aware of all possible responses and what they mean. Refer to Tables 9-2, 9-3, and 9-4 for a description of each bit in the data returned in register AX.

Table 9-2. Register AX Report from INT 11H

BIT	DESCRIPTION
0	If this bit is set to 1, floppy disk drives are installed in the system. If this bit is set to 0, then no floppy disk drives are installed. 1If this bit is set to 1, then the 8087 Numeric Data Coprocessor is installed. If this bit is set to 0, then the coprocessor is not installed.
2 and 3	These two bits indicate the device size of the RAM chips installed in the computer. Bits 2 and 3 will always be set to 1, indicating that either 64- or 256-kilobit devices are installed. Some early PC-type computers used 16- and 32-kilobit devices in memory.
4 and 5	These two bits indicate the initial video mode at power-up. See Table 9-3.
6 and 7	These two bits report the number of floppy disk drives installed in the computer according to hardware switch or firmware settings. See Table 9-4.
8	Unused in Zenith Data Systems computers.
9 – 11	These three bits indicate the number of RS-232 ports in the system. The standard input/output on this system emulates the IBM PC input/output, so the minimum will be one port.
12	If this bit is set to 1, then a game card is installed. It is set to 0 in the Laptop computer.
13	Unused by Zenith Data Systems computers.
14 and 15	These two bits report the number of printers installed.

Table 9-3. Bits 4 and 5: Video Initialization

BIT 4	BIT 5	DESCRIPTION
0	0	80 × 25 text mode on an EGA card.
0	1	40 × 25 text mode on a color card.
1	0	80 × 25 text mode on a color card.
1	1	80 × 25 text mode on a monochrome card.

NOTE: The Laptop computer emulates the color card, but does not support PC-compatible cards. Therefore, the only modes recognized are the 40 × 25 text mode and the 80 × 25 text mode on a color card.

Table 9-4. Bits 6 and 7: Disk Drive Count

BIT 6	BIT 7	DRIVE COUNT
0	0	1 Drive
0	1	2 Drives
1	0	3 Drives
1	1	4 Drives

NOTE: The third and fourth drives are external on the Laptop computer. Most PC-compatible computers do not support floppy disk drive 3 and 4. In those cases, drive C and D are usually reserved for Winchester partitions under MS-DOS.

Memory Size (INT 12H)

INT 12H is the memory size interrupt. It returns the number of contiguous 1K blocks of user memory in the system in register AX. For instance, if the value in AX is 256 following this interrupt, then the computer has determined that there is 256K of memory installed in the system.

Set/Read the Time of Day (INT 1AH)

INT 1AH is the set/read the time of day interrupt. It allows the programmer to set and/or read the internal clock's time and date, and to set the alarm on or off.

Register AH establishes the type of operation (read or write). Registers CX and DX are used for the value to be passed to or from the 32-bit counter. Register AL is used as a flag to report that the value has rolled over into a new day since the last time the counter was read.

In most PC-compatible computers, the value that represents the time of day is stored in a 32-bit time-of-day counter, which is updated 18.2159 times a second. Therefore, 1 a.m. would be represented by a count of 65,577 and 12 noon would be 786,926. Register CX passes the high part of the count and would contain 1 for 1 a.m. and 12 (0CH) for 12 noon. Register DX passes the low part of the count and would contain 41 (29H) for 1 a.m. and 494 (1EEH) for 12 noon. The contents of register AL would depend upon whether the count had rolled over into a new day since the last time the counter was read. If AL is zero, then the counter has not advanced past 24 hours. If AL is any value other than zero, then it had advanced past 24 hours.

To read the current time-of-day value from the time-of-day word, place a zero in register AH and execute the interrupt. Upon return from the call, register CX will contain the high part of the count, and register DX will contain the low. If the count has rolled over, register AL will not contain zero. If AL contains a zero, the value has not rolled over to a new day since the last time the counter was read.

To set the time-of-day value, place the high count in the CX register and the low count in the DX register. Place a 1 in register AH and execute the interrupt.

In the Laptop computer, a real-time clock IC keeps track of the date and time. While this IC remains active even when the computer is turned off, the 32-bit time-of-day word is still used for compatibility. When the DOS is first loaded, it (version 3.2 or higher) will automatically read the time of day from the clock IC into the 32-bit word, or a utility program can be run which will do the same thing. The end result is that software that depends upon the 32-bit word will still work, in the case of the Laptop computer, a back-up real-time clock is also present.

Tick Timer (INT 1CH)

INT 1CH is the tick timer interrupt. It provides a means for you to produce CPU-independent timing loops in your programs. At power-up, this interrupt points to an IRET instruction.

The speed of the CPU clock frequency in PC-compatible computers is inconsistent from one machine to another. Furthermore, some models offer switchable-speed CPU clock frequencies. This interrupt provides a convenient means of consistently controlling timing loops regardless of the CPU clock frequency.

This interrupt is called every time INT 08H (the timer interrupt) is executed (18.2159 times a second). Therefore, if you use INT 1CH in a program, you must save the registers at the beginning of the routine and then restore them before returning to the main program.

Programming Sound

This computer does not contain a dedicated sound chip. However, tones may be generated and played through the speaker via the 82C55 programmable peripheral interface chip in three different ways, all of which may be implemented simultaneously.

1. Generate a pulse train by toggling a program control register bit.
2. Program the output of channel 2 of the timer/counter to deliver a sound waveform to the speaker.
3. Modulate the clock input to the timer/counter through a program-controlled input/output register bit.

Programming sound, particularly music, is a specialized application in itself and is beyond the intended scope of this manual. If you want to produce sounds or music, you should use one of the special application programs that are designed for that purpose or GW-BASIC, which contains statements to perform this type of task.

User Memory

In general, programming user memory is straight forward. Most operations can be performed directly from assembly or machine language programs and include the following.

- Reading and writing data to specific memory locations.
- Allocating and/or reserving specific locations in memory for specific purposes, such as for software interrupt routines or storing user-defined character fonts.
- Storing character strings and numeric values that are widely used by a number of applications.
- Rerouting interrupts to user-defined routines.
- CPU stack operations.
- Moving memory contents from one area to another.
- Using the contents of RAM to control video graphics.

NOTE: The Laptop computer family of computers does not contain memory parity checking circuits. If you attempt to disable and/or enable parity generation and/or checking, nothing will happen.

Memory Address Format

The 80C88 uses a 2-part number to designate specific memory locations. The hexadecimal number consists of a 4-digit number to identify the segment address and a 4-digit number to identify the offset address within that segment. The format for this value is:

XXXX:YYYY

The first four digits actually represent a 5-digit hexadecimal memory address, since an imaginary shift left (multiply by 16) is performed on the value to arrive at the RAM bank and row to select. The second value selects the RAM column from the selected bank. For example, the value 3F3F:5B11 would represent memory address 44F01H. 3F3F shifted left equals 3F3F0H. 5B11 is the offset that is added to the segment address. The result of adding 3F3F0H and 5B11H is 44F01H.

Since the least-significant digit of the segment is always 0, more than one combination of segment and offset can point to the same memory address. For instance, the address in the previous example could also be defined by 3F00:5F01.

The highest and lowest usable segment values are determined by the memory location being defined. In the example address, the lowest usable segment value would be 34F1H, since anything lower would result in an offset greater than FFFFH. The highest usable segment value would be 44F0H.

Chapter 10

Keyboard Interrupts and Codes

Programming Keyboard Interrupts

The three interrupts that are applicable to the keyboard are described in Table 10-1. Following a description of the interrupts are tables that describe the codes generated by the keyboard.

Table 10-1. Keyboard Interrupts

INTERRUPT	FUNCTION
09H	Key Pressed
16H	Keyboard Input/Output
1BH	Keyboard Break

Key Pressed (INT 09H)

INT 09H is the key pressed interrupt. It is executed each time a key is pressed or released. The interrupt routine reads the key from the keyboard register and encodes it or notes the action if it is a shift or control key. Valid key codes are placed in the keyboard buffer where they can be acted upon by applications or system software.

You should not change the action of this interrupt routine since it directly affects the action of the keyboard.

Keyboard Input/Output (INT 16H)

INT 16H is the keyboard input/output interrupt. It is used to perform three routine keyboard operations. Three function codes, 0, 1, and 2, perform key code retrieval (receive a character from the keyboard), check the keyboard buffer to see if any codes are in it, and report the status of the keyboard shift and control keys.

Function Code 00H: Get Character — This function code will cause the interrupt to get a character from the keyboard buffer. If a key has been typed and is waiting in the keyboard buffer, it will be retrieved and removed from the buffer. The code (ASCII codes 00H – 7FH or non-ASCII codes 80H – FFH) assigned to that character will be returned in the AL register. The scan code (01H – 84H) for the key pressed will be returned in register AH. If the keyboard buffer is empty, the routine will wait until a key has been pressed on the keyboard, generating a character code. Note that the shift and control keys do not generate codes, but affect the codes generated by other keys.

Function Code 01H: Check Keyboard Buffer — This function code will cause the interrupt to check the status of the keyboard buffer. If the buffer is empty, the zero flag (ZF) will be set. If the buffer contains key codes waiting to be processed, the zero flag will be cleared (not set). The code (ASCII codes 00H – 7FH or non-ASCII codes 80H – FFH) assigned to the first character in the buffer will be returned in the AL register. The scan code (01H – 84H) for that same key will be returned in register AH.

NOTE: This operation does not remove any codes from the keyboard buffer. Therefore, if you execute function code 01H followed by function code 00H, the same key codes will be returned. Only function code 00H removes the key codes from the buffer.

Function Code 02H: Get Keyboard Status — The keyboard status is stored in two bytes in memory: 0040:0017 and 0040:0018. This function code will cause the interrupt to report the status of the keyboard stored in 0040:0017. The value of this byte is returned in register AL and is described in Table 10-2. If the report states that a key is pressed, that key was being held down at the time this function was executed.

Table 10-2. Keyboard Status Report (0040:0017)

BIT (AL)	DESCRIPTION
0	If this bit is set, the right SHIFT key is pressed.
1	If this bit is set, the left SHIFT key is pressed.
2	If this bit is set, the CTRL key is pressed.
3	If this bit is set, the ALT key is pressed.
4	If this bit is set, the SCROLL LOCK mode is active.
5	If this bit is set, the keypad lock mode is active.
6	If this bit is set, the CAPS LOCK mode is active.
7	If this bit is set, the INS (insert) mode is active.

The keyboard status stored in byte 0040:0018 can be read directly and interpreted by your own routines. The value of this byte is described in Table 10-3.

Table 10-3. Keyboard Status (0040:0018)

BIT	DESCRIPTION
0	Not used.
1	Not used.
2	Not used.
3	If this bit is set, the CTRL-NUM LOCK (pause) mode is active.
4	If this bit is set, the SCROLL LOCK key is pressed.
5	If this bit is set, the NUM LOCK key is pressed.
6	If this bit is set, the CAPS LOCK key is pressed.
7	If this bit is set, the INS key is pressed.

Keyboard Break (INT 1BH)

The INT 1BH instruction is executed when the key pressed interrupt (09H) detects the CTRL and BREAK keys (CTRL-BREAK) at the same time (The CTRL key must be pressed first and then held while the SCROLL LOCK/BREAK key is pressed).

The interrupt for this routine must return with an IRET instruction to properly exit the 09H interrupt. When the computer is turned on, this routine provides only the IRET instruction to make sure that nothing will happen if CTRL-BREAK is pressed during the self-tests.

GW-BASIC is an example of software that uses this routine. It uses this interrupt to halt execution of a BASIC program when the CTRL-BREAK key is pressed.

If you allow this interrupt to retain control, you may have to accommodate one or more of the following conditions. For more information, see the discussion on the 82C59 interrupt controller in Chapter 14.

- The break may occur during interrupt processing. You must then send one or more end-of-interrupt commands to the 82C59 interrupt controller.
- If an operation was in process when you caused the interrupt to take place, all input/output devices must be reset.
- Programs that use this interrupt must not chain to the previous owner of this interrupt and the interrupt must be restored when you are finished.

Remember, when servicing an INT 1BH instruction, your routine must perform an IRET to make sure that the interrupt is restored. Do not link your service routine to the next one.

Keyboard Codes

The keyboard on the Laptop computer generates a wide range of keycodes in order to duplicate all the functions of a fully-defined PC-compatible keyboard. Tables 10-4, 10-5, 10-6, 10-7, 10-8, and 10-9 summarize the various key scan codes and functions available from the Lap Top computer keyboard.

Alphabetic Keys

The keyboard interface is defined so system software has the maximum flexibility in defining keyboard operation. This is accomplished by having the keyboard return specific hexadecimal codes rather than ASCII codes. In addition, all keys except control keys can be event-driven and generate both make and break codes, commonly referred to as up and down codes. For example, the ESC key (defined internally as key 1) produces 01 hexadecimal when pressed (down) and 81 hexadecimal when released (up). The keyboard controller can produce code either with or without control keys (SHIFT, CTRL, FN, ALT) pressed, or under event-driven conditions as required by the application. In addition, different codes are produced by placing the keyboard in one of the five operating modes. These operating modes are discussed in detail in Chapter 7 of this manual.

The scan codes generated from the alphabetic keys are listed in Table 10-4. The least-significant byte of the scan code is the value returned as the ASCII code.

Figure 10-1 provides an illustration of the alphabetic keys.

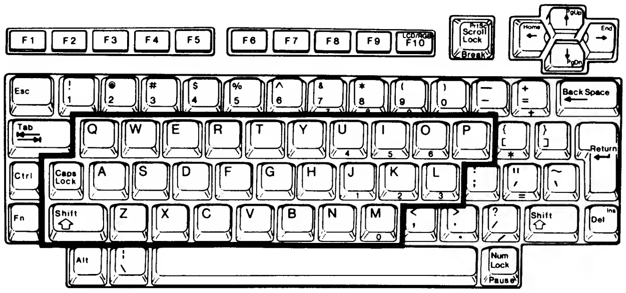

Figure 10-1. Alphabetic Keys

Table 10-4. Alphabetic Key Scan Codes

KEY	NO SHIFT	SHIFT	CTRL	ALT	NUMERIC	CURSOR
A	1E61H	1E41H	1E01H	1E00H	——	
B	3062H	3042H	3002H	3000H	——	
C	2E63H	2E43H	2E03H	2E00H	——	
D	2064H	2044H	2004H	2000H	——	
E	1265H	1245H	1205H	1200H	——	
F	2166H	2146H	2106H	2100H	——	
G	2267H	2247H	2207H	2200H	——	
H	2368H	2348H	2308H	2300H	——	
I/5	1769H	1749H	1709H	1700H	4C35H	
J/1	246AH	244AH	240AH	2400H	4F31H	4F00H
K/2	256BH	254BH	250BH	2500H	5032H	5000H
L/3	266CH	264CH	260CH	2600H	5133H	5100H
M/0	326DH	324DH	320DH	3200H	5230H	5200H
N	316EH	314EH	310EH	3100H	——	
O/6	186FH	184FH	180FH	1800H	4D36H	4D00H
P	1970H	1950H	1910H	1900H	——	
Q	1071H	1051H	1011H	1000H	——	
R	1372H	1352H	1312H	1300H	——	
S	1F73H	1F53H	1F13H	1F00H	——	
T	1474H	1454H	1414H	1400H	——	
U/4	1675H	1655H	1615H	1600H	4B34H	4B00H
V	2F76H	2F56H	2F16H	2F00H	——	
W	1177H	1157H	1117H	1100H	——	
X	2D78H	2D58H	2D18H	2D00H	——	
Y	1579H	1559H	1519H	1500H	——	
Z	2C7AH	2C5AH	2C1AH	2C00H	——	

NOTE: In the above table a dashed entry (——) indicates that the output code will be the same as the unshifted key code. A blank indicates that no output results from this combination.

Numeric and Punctuation Keys

The numeric and punctuation keys include the numbers 0 through 9, the common punctuation marks, and the special programming characters that make up the remainder of the printable ASCII character set. Since each of these keys has two characters, the upper character is generated by pressing either SHIFT key. The CAPS LOCK key does not affect the operation of these keys and is therefore not listed in Table 10-4, which lists the codes produced by each key. Again, the ASCII code is the least-significant byte of the scan codes shown. Note that some keys do not produce a unique scan code when pressed in conjunction with the CTRL or ALT key. In those instances, the code produced will be the same as if the CTRL or ALT key were not pressed and will be dependent upon the state of the SHIFT keys. Table 10-5 summarizes the keycodes for these keys.

Figure 10-2 illustrates the numeric and punctuation keys used on the Laptop computer.

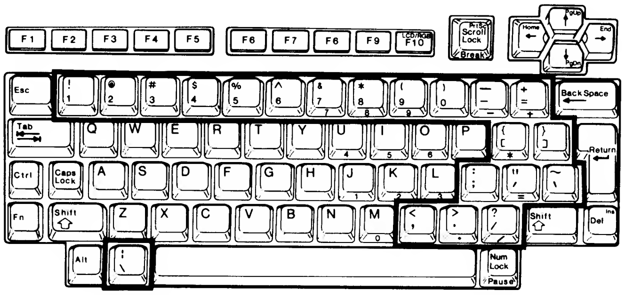

Figure 10-2. Numeric and Punctuation Keys

Table 10-5. Nonalphabetic Key Scan Codes

KEY	NO SHIFT	SHIFT	CTRL	ALT	NUMERIC	CURSOR
1/!	0231H	0221H	——	7800H	——	
2/@	0332H	0340H	0300H	7900H	——	
3/#	0433H	0423H	——	7A00H	——	
4/$	0534H	0524H	——	7B00H	——	
5/%	0635H	0625H	——	7C00H	0635H	
6/ ^	0736H	075EH	071EH	7D00H	——	
7/&/7	0837H	0826H	——	7E00H	4737H	4700H
8/*/8	0938H	092AH	——	7F00H	4838H	4800H
9/(/9	0A39H	0A28H	——	8000H	4939H	4900H
0/)	0B30H	0B29H	——	8100H	——	
-/_/-	0C2DH	0C5FH	0C1FH	8200H	4A2DH	
=/+/+	0D3DH	0D2BH	——	8300H	4E2BH	
`/~	2960H	297EH	——	——		
[/{/*	1A5BH	1A7BH	1A1BH		372AH	372AH
]/}	1B5DH	1B7DH	1B1DH	——	——	
;/:	273BH	273AH	——	——		
'/"/=	2827H	2822H	——	——	0D3DH	0D3DH
,/<	332CH	333CH	——	——	342EH	
./>/.	342EH	343EH	——	——	0D2BH	
//?//	352FH	353FH	——	——	——	
\/:	2B5CH	2B7CH	2B1CH	——	——	
BACKSP	0E08H	——	0E7FH	0E7FH		

NOTE: In the above table a dashed entry (——) indicates that the output code will be the same as the unshifted key code. A blank indicates that no output results from this combination.

Table 10-6. Common Control Key Scan Codes

KEY	NO SHIFT	SHIFT	CTRL	ALT	FUNCTION
BACK SPACE	0E08H	0E08H	0E7FH	——	Moves the cursor one position to the left. Often used to erase data.
TAB	0F09H	0F00H	——	——	Moves the cursor to the next tab column or to the previous tab column with SHIFT key pressed.
RETURN	1C0DH	1C0DH	1C0AH	——	Returns the cursor to the left side of display. Software usually adds a line feed. Indicates completion of operator entry.
space bar	3920H	3920H	3920H	3920H	Enters a blank character (ASCII 20H).

Common Control Keys

The common control key scan codes and the usual software function are listed in Table 10-6. These keys are illustrated in Figure 10-3.

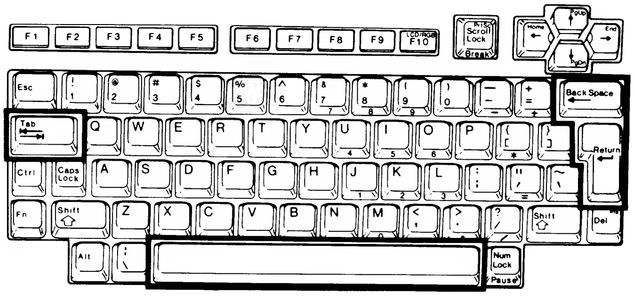
F1 F2 F3 F4 F5 F6 F7 F8 F9 LCD/RGB F10
Scroll Lock Break
Home PgUp End PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — - + =
Back Space
Tab Q W E R T Y U I O P { [}]
Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ \
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt Num Lock Pause

Figure 10-3. Common Control Keys

Control Keys

The control keys do not generate a code but modify the codes and action of the other keys. Some of these keys alter the way the keyboard controller functions, which, in turn, alters operation of various facets of the computer. Table 10-7 describes each key or key combination. Figure 10-4 illustrates these keys.

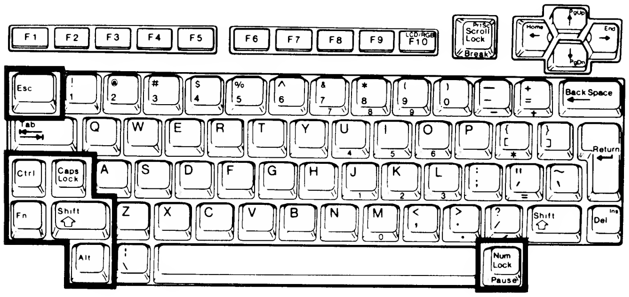
F1 F2 F3 F4 F5 F6 F7 F8 F9 LCD/RGB F10
Scroll Lock Break
Home PgUp End PgDn
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — - + =
Back Space
Tab Q W E R T Y U I O P { [}]
Return
Ctrl Caps Lock A S D F G H J K L : ; " ' ~ \
Fn Shift Z X C V B N M < , > . ? / Shift Ins Del
Alt Num Lock Pause

Figure 10-4. Control Keys

Table 10-7. Control Keys and Combinations

KEY(S)	NAME AND DESCRIPTION
ESC	Escape. This key generates ASCII code 1BH (scan code 011BH). The function is often used by software to interrupt (that is, escape from) a program function. It may also be used in sequence with another key for entering escape codes.
CTRL	Control.
FN	Function.
ALT	Alternate. The ALT key can be used to generate any hexadecimal code from 0 to FFH (0 to 255 decimal). To do so, press and hold the ALT key while entering the decimal equivalent of the hexadecimal code that is to be generated. Then release the ALT key. When the ALT key is released, the conversion takes place and the hexadecimal code is generated.
NUM LOCK	Numbers Lock. The NUM LOCK key is used to lock the keypad into the numeric position. This allows the operator to use the keypad in a manner similar to a 10-key calculator. However, the NUM LOCK key only locks the numeric and arithmetic keys into the keypad state. During the use of the NUM LOCK feature, the keypad keys will generate the same scan codes as the numerals in the top row of the keyboard.
CAPS LOCK	Caps Lock.
SHIFT	Shift.

Special Function Key Combinations

The special function key scan codes and functions (if any) are listed in Table 10-8. Figure 10-5 illustrates the keys in this section.

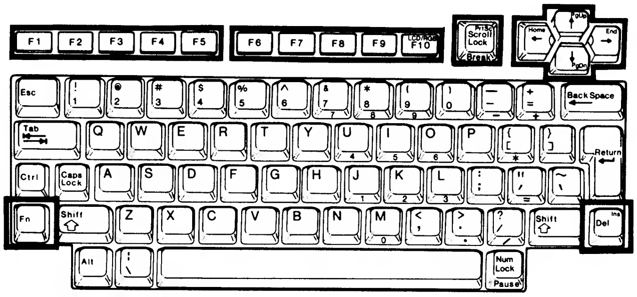

Figure 10-5. Special Function Keys

Table 10-8. Special Function Key Scan Codes

KEY	NO SHIFT	SHIFT	CTRL	ALT	FUNCTION
F1	3B00H	5400H	5E00H	6800H	Function key 1 When pressed with the FN key, equals Function key 11.
F2	3C00H	5500H	5F00H	6900H	Function key 2 When pressed with the FN key, equals Function key 12.
F3	3D00H	5600H	6000H	6A00H	Function key 3
F4	3E00H	5700H	6100H	6B00H	Function key 4
F5	3F00H	5800H	6200H	6C00H	Function key 5
F6	4000H	5900H	6300H	6D00H	Function key 6
F7	4100H	5A00H	6400H	6E00H	Function key 7
F8	4200H	5B00H	6500H	6F00H	Function key 8
F9	4300H	5C00H	6600H	7000H	Function key 9
F10/LCD/ RGB	4400H	5D00H	6700H	7100H	Function key 10 or, when pressed with the FN key, toggles the video output between the LCD display and the RGB output connector on the back of the computer.
PRTSC/ SCROLL LOCK/ BREAK					When pressed with the SCROLL LOCK/FN key, causes the contents of the screen to be printed, locks screen scrolling, or when pressed with the CTRL key, interrupts program execution (see Table 10-5).
HOME/ ←	4B00H (*0004H is produced when ALT released)		7300H	0004H*	Moves the cursor one position to the left or, when pressed with the FN key, "homes" the cursor (see Table 10-5).
PGUP/ ↑	4800H (*0008H is produced when ALT released)	4900H		0008H*	Moves the cursor up one line or, when pressed with the FN key, moves to the previous page (see Table 10-5).
END/ →	4D00H (*0006H is produced when ALT released)	4F00H	7400H	0006H*	Moves the cursor one position to the right, or when pressed with the FN key, moves to the end of the text (see Table 10-5).
PGDN/ ↓	5000H (*0002H is produced when ALT released)	5100H		0002H*	Moves the cursor down one line or, when pressed with the FN key, moves to the next page (see Table 10-5).
FN-DEL	5200H	5100H			Enters the insert mode (word processing). Used to delete characters (word processing).

The use of the special purpose cursor movement keys varies between application packages. Table 10-9 provides describes some of the ways these keys are used in selected application packages. In most cases, these keys are used in other ways by screen-oriented games.

Table 10-9. Special Purpose Keys

KEY	NAME AND DESCRIPTION
PRTSC	Print Screen. This key is used with the operating system to print the contents of the screen. To take advantage of graphics capabilities of most printers, you will have to load a printer driver into the computer from the operating system before you use this key.
SCROLL LOCK	Scroll lock. This key is generally used to control screen scrolling. In the case of GW-BASIC, this key allows programs to be displayed one screen at a time. In the case of Word, this key toggles the cursor up and down keys between normal cursor movement and movement of the text, which causes the text to scroll up or down past the cursor position.
BREAK	Break. This key is generally used to break into program execution. In GW-BASIC, this key is pressed with the CTRL key to interrupt execution of a command or program, similar to the action of CTRL-C.
HOME	Home. This key is generally used to "home" the cursor (that is, move it to the upper left-hand corner of the screen). Microsoft Word uses this key to move the cursor to the beginning of the current line. WordStar uses this key to move to the beginning of the document.
PGUP	Page Up. This key is generally used by word or text processors to move the cursor up through the document (toward the beginning) one screen full of text.
END	End. This key is generally used by word or text processors to move the cursor to the end of the document. Word uses this key to move the cursor to the end of the current line.
PGDN	Page Down. This key is generally used by word or text processor to move the cursor down through the document (toward the end) one screen full of text.

UP/DOWN Key Codes

As mentioned earlier, each key on the keyboard will produce two distinct codes each time it is used. The DOWN code is produced when the key is pressed and the corresponding UP code is produced when the key is released. It should be noted that these codes are distinct from the scan codes resulting from the use of the INT 16H function. In order to access the UP/DOWN codes you must use the INT 09H function. It should also be noted that altering the keyboard processing codes should only be attempted by experienced programmers. Tables 10-10, 10-11 and 10-12 list the UP/DOWN codes for all keyboard keys.

Table 10-10. Alphabetic Key Up/Down Codes

KEY	DOWN	UP	FN-DOWN	FN-UP
A	1EH	9EH		
B	30H	B0H		
C	2EH	AEH		
D	20H	A0H		
E	12H	92H		
F	21H	A1H		
G	22H	A2H		
H	23H	A3H		
I	17H	97H	4CH	CCH
J	24H	A4H	4FH	CFH
K	25H	A5H	50H	D0H
L	26H	A6H	51H	D1H
M	32H	B2H	52H	D2H
N	31H	B1H		
O	18H	98H	4DH	CDH
P	19H	99H		
Q	10H	90H		
R	13H	93H		
S	1FH	9FH		
T	14H	94H		
U	16H	96H	4BH	CBH
V	2FH	AFH		
W	11H	91H		
X	2DH	ADH		
Y	15H	95H		
Z	2CH	ACH		

Table 10-11. Non-Alphabetic Key Up/Down Codes

KEY	DOWN	UP	FN-DOWN	FN-UP
!/1	02H	82H		
@/2	03H	83H		
#/3	04H	84H	04H	84H
$/4	05H	85H	05H	85H
%/5	06H	86H	06H	86H
^/6	07H	87H	07H	87H
&/7	08H	88H	47H	C7H
*/8	09H	89H	48H	C8H
(/9	0AH	8AH	49H	C9H
)/0	0BH	8BH	0BH	8BH
_/-	0CH	8CH	4AH	CAH
+/=	0DH	8DH	4EH	CEH
~/`	29H	A9H		
{/[	1AH	9AH	37H	B7H
}/]	1BH	9BH		
:/;	27H	A7H		
“/'	28H	A8H	0DH	8DH
</,	33H	B3H		
>/.	34H	B4H	53H	D3H
?//	35H	B5H	35H	B5H
¦/\	2BH	ABH		

Table 10-12. Control and Special Function Key Up/Down Codes

KEY	DOWN	UP	FN-DOWN	FN-UP
Esc	01H	81H	55H	D5H
Tab	0FH	8FH		
Lt Shift	2AH	AAH		
Space	39H	B9H		
Back Space	0EH	8EH	E04CH	E0CCH
Return	1CH	9CH		
Num Lock/ Pause	45H	C5H		
F1	3BH	BBH	57H	D7H
F2	3CH	BCH	58H	D8H
F3	3DH	BDH		
F4	3EH	BEH		
F5	3FH	BFH		
F6	40H	C0H		
F7	41H	C1H		
F8	42H	C2H	5EH	DEH
F9	43H	C3H	5FH	DFH
F10	44H	C4H	5DH	DDH
Alt	38H	B8H		
Rt Shift	36H	B6H		
←/Home	E04BH	E0CBH	E047H	E0C7H
Up Arrow/ PgUp	E048H	E0C8H	E049H	E0C9H
→/End	E04DH	E0CDH	E04FH	E0CFH
Dn Arrow/ PgDn	E050H	E0D0H	E051H	E0D1H
PrtSc/ Scroll Lock/ Break	46H	C6H	2A37B7AAH	
Caps Lock	3AH	BAH		
Del/Ins	E053H	E0D3H	E052H	E0D2H

Chapter 11

Input/Output Interrupts

Programming Input/Output Interrupts

The input/output interrupts are defined in Table 11-1 and described in this chapter. Refer to Chapter 8 for information on how to use and program these interrupts.

Table 11-1. Input/Output Interrupts

INTERRUPT	FUNCTION
05H	Print screen
0BH	Communications (COM2)
0CH	Communications (COM1)
0DH	Alternate parallel printer (LPT2)
0FH	Parallel printer (LPT1)
14H	Serial input/output
17H	Printer input/output
18H	Parallel/serial configuration

Print Screen (INT 05H)

This interrupt sends the contents of the screen to the LPT device (usually a printer). Valid characters, or those that match a valid character, will be sent to the printer. All other shapes will be ignored by the default print screen routine. This interrupt routine performs the same function that is performed when you press the FN-PRTSC key.

To aid in using the print screen function, a byte of memory at location 0050:0000 has been reserved as a status byte. While the print screen routine is executing, this byte is set to 01H. This is actually a flag used by the print screen routine to prevent additional print screen requests from being honored while the routine is being executed. Upon completion of the print screen routine, the status byte is changed to 00H if no errors occurred, or FFH if an error occurred. The error is usually caused by a printer time-out.

Communications (INT 0BH and INT 0CH)

The INT 0BH and INT 0CH instructions were written for serial communications through COM1 (INT 0CH) and COM2 (INT 0BH). COM2 is reserved for use with the optional internal modem. These interrupts are reserved for use with applications software, so if you want to use these interrupts for your own communication activities, you will have to supply your own input/output routines.

NOTE: Interrupts INT 14H and INT 18H provide support for Monitor program and MS-DOS serial communications.

Parallel Printer (INT 0DH and INT 0FH)

The INT 0FH (LPT1) the INT 0DH (LPT2) interrupts are used for parallel communications, generally with printers. These interrupts are reserved for use with applications software, so if you want to use these interrupts, you will have to supply your own routines.

NOTE: INT 17H provides complete support for parallel communications.

Serial Input/Output (INT 14H)

The INT 14H interrupt will allow you to perform the serial functions described in Table 11-2 and explained in the following paragraphs. For more information, see "Parallel/Serial Configuration" in this chapter.

Before the interrupt can be successfully executed, register AH must be loaded with a specific function code. Register DX must be loaded with 0 for COM1 or 1 for COM2.

NOTE: The addressed serial device must be connected to the system. Since the Laptop computer has only one serial port, register DX must always be set to 0 for COM1.

Table 11-2. Serial Input/Output Function Codes

CODE	DESCRIPTION
00H	Initialize the serial input/output port.
01H	Send character to the serial port.
02H	Receive character from the from serial port.
03H	Read communications status.

Function Code 00H: Initialize the Serial Input/Output Port — This function code initializes the parameters of the serial port according to the value placed in register AL. This value is defined in Table 11-3 through Table 11-6. Register DX must be loaded with 0 for the COM1 port.

Table 11-3. Mode-Select Byte Breakdown

BIT	DESCRIPTION
0	With bit 1, sets word length (see Table 11-4).
1	With bit 0 sets word length (see Table 11-4).
2	Number of stop bits. Set to 0 for 1 stop bit, or 1 for 2 stop bits.
3	With bit 4, sets parity selection (see Table 11-5).
4	With bit 3, sets parity selection (see Table 11-5).
5	With bits 6 and 7, sets baud rate (see Table 11-6).
6	With bits 5 and 7, sets baud rate (see Table 11-6).
7	With bits 5 and 6, sets baud rate (see Table 11-6).

Table 11-4. Word Length Selection

BIT 1	BIT 0	WORD LENGTH
0	0	5 bits (Not supported in PC computers).
0	1	6 bits (Not supported in PC computers).
1	0	7 bits.
1	1	8 bits.

NOTE: If the word length to be transmitted or received is set to less than 8, the character will be converted from or to a full 8-bit byte by the asynchronous communications element. Extra bits will be ignored or set to 0.

Table 11-5. Parity Selection

BIT 4	BIT 3	SELECTION
0	0	No parity.
0	1	Odd parity.
1	0	No parity.
1	1	Even parity.

Table 11-6. Baud Rate Selection

BIT 7	BIT 6	BIT 5	BAUD RATE
0	0	0	110
0	0	1	150
0	1	0	300
0	1	1	600
1	0	0	1200
1	0	1	2400
1	1	0	4800
1	1	1	9600

NOTE: Although the hardware is capable of handling other baud rates, these are the only rates supported by the interrupt.

Upon return from the routine, the 16-bit register AX will contain the serial port status report. The 8 bits of register AX that make up register AH will contain the line control status (see Table 11-7), and the 8 bits of register AX that make up register AL will contain the modem control status (see Table 11-8). Function code 03H of this interrupt wi!l return the same report.

Function Code 01H: Send Character to the Serial Port — This function code causes the byte placed in register AL to be transmitted out the specified serial port. Register DX must be loaded with 0 for the COM1 port.

Upon return from the routine, the 16-bit register AX will contain the serial port status report. The 8 bits of register AX that make up register AH will contain the line control status (see Table 11-7). The 8 bits of register AX that make up register AL will contain the modem control status (see Table 11-8). Function code 03H of this interrupt will return the same report.

NOTE: In place of the timeout error indicated in Table 11-7, the most-significant bit of register AH will contain the transmit status bit. If the routine could not transmit the character placed in register AL for one reason or another, the transmit status bit will be set (a 1).

Function Code 02H: Receive Character from the Serial Port — This function code causes the byte at the specified serial port to be loaded into register AL. Register DX must be loaded with 0 for the COM1 port.

Upon return from the routine, the 16-bit register AX will contain the serial port status report. The 8 bits of register AX that make up register AH will contain the line control status (see Table 11-7). The 8 bits of register AX that make up register AL will contain the modem control status (see Table 11-8). Function code 03H of this interrupt will return the same report.

NOTE: Bits 7, 4, 3, 2, and 1 of register AH will contain the data transfer status (see Table 11-7). If the contents of register AH is 0, the routine has read the byte properly into AL. If AH is not 0, some type of error occurred. In these cases, time-out errors refer to either the absence of either the data set ready (DSR) signal or the clear to send (CTS) signal.

Function Code 03H: Read Communications Status — This function code will return a report of the status of the specified communications port. Register DX must be loaded with 0 for the COM1 port.

The status report is returned as a bit-mapped value in register AX. The 8 bits of register AX that make up register AH contain the line control status (see Table 11-7). The 8 bits of register AX that make up register AL contain the modem control status (see Table 11-8).

Table 11-7. Line Control Status (Register AH)

BIT	DESCRIPTION
0	The received data is ready.
1	An overrun error has occurred. Another character has arrived and the previously received character has not been read by the system.
2	A parity error has occurred. Parity was enabled, but the parity of the incoming character did not match that which was programmed.
3	A framing error has occurred. Framing is handled by the first bit of the word and the stop bit. Either the system incorrectly received the transmitted character or the total number of characters in the received word did not match that which was programmed.
4	The BREAK key has been detected. Some terminals have a BREAK key that, when pressed, will hold the data line low. This bit detects that action.
5	The transmitter holding register is empty. This indicates that the serial input/output channel is ready to receive another character to be transmitted.
6	The transmitter shift register is empty. This indicates that the specified serial input/output channel is not currently transmitting.
7	Time-out error. The receiving device did not respond within a reasonable period of time. A printer, when off line, will typically return this type of error.

Table 11-8. Modem Control Status (Register AL)

BIT	STATUS
0	The clear to send (CTS) line has changed state.
1	The data set ready (DSR) status has changed.
2	The tailing edge of the ring signal has been detected.
3	The carrier detect (CD) signal has changed state.
4	The current state (high or low) of the clear to send line.
5	The current state (high or low) of the data set ready line.
6	The current state (high or low) of the ring indicator line.
7	The current state (high or low) of the carrier detect line.

Printer Input/Output (INT 17H)

The INT 17H instruction is used to perform input/output functions with the parallel port on the computer. Before the interrupt can be executed, register AH must be loaded with one of the function codes described in Table 11-9. The parallel configuration table, which is set up by interrupt 18H, is used by this interrupt to complete its operation. Refer to "Parallel/Serial Configuration" in this chapter.

Table 11-9. Printer Input/Output Function Codes

FUNCTION CODE	DESCRIPTION
00H	Print the next character. The character in register AL is sent to the port. If the parallel device does not return a ready status within a reasonable amount of time, bit 0 of register AH will be set to 1, otherwise the register will return the status report (see Table 11-10). Register DX must be loaded with the port number to be used (00H for LPT1, 01H for LPT2, etc.).
01H	Initialize the printer port. Register AH will return the status report following execution of the interrupt (see Table 11-10).
02H	Read status of printer port. This function returns the status report (see Table 11-10) in register AH.

Table 11-10. Parallel Printer Status Report

BIT	DESCRIPTION
0	A timeout error has occurred. The parallel device is probably not ready, perhaps off line.
	1 Not used.
	2 Not used.
	3 An input or output error has occurred.
	4 The device is on line.
	5 The out of paper signal is active.
6	The transmitted character is acknowledged; that is, it has been received by the device.
7	The device is busy or is in an error state.

Parallel/Serial Configuration (INT 18H)

The INT 18H instruction is used as a pointer to the parallel channel and serial channel device tables that are used by MS-DOS and many application packages to configure the computer's serial and parallel ports. It is also used to route the parallel printer output to a serial communications channel.

NOTE: In IBM PC computers that contain BASIC in ROM, this interrupt is used to point to BASIC. Programs that make a specific call the BASIC ROM routines will not work on Zenith Data Systems computers, including the Laptop computer.

In order to maintain compatibility between various versions of MS-DOS, the default configuration tables are first loaded by the Monitor program. MS-DOS and application programs can either modify these tables, or install new tables and modify the pointers accordingly.

Chapter 12

Disk Drive Interrupts

Programming Disk Drive Interrupts

The disk drive interrupts are defined in Table 12-1 and described in this chapter. Refer to Chapter 8 for information on how to use and program these interrupts.

NOTE: Even though the Laptop computer does not currently support a rigid (Winchester) disk system, software support is still provided and described in this chapter. Any interrupt that is not initialized will jump to a return (RET) function (no operation).

Table 12-1. Data Storage Interrupts

INTERRUPT	FUNCTION
0EH	Floppy disk drive
13H	Disk input/output
19H	Booting an operating system
1EH	Disk parameters

Floppy Disk Drive (INT 0EH)

The INT 0EH interrupt is used for communication with floppy disk drives. It is normally used by the Monitor program to provide disk input/output at the hardware level and should not be modified by user programs. You will find the INT 13H instruction more useful for most disk communication.

Disk Input/Output (INT 13H)

The INT 13H interrupt is used to communicate with the system's disk drives. Table 12-2 lists the port addresses associated with the disk drives.

NOTE: Even though the hard or rigid disk drive (Winchester) is not supported by the Lap Top Computer, software support is provided for PC compatibility. As noted earlier, any interrupt that is not initialized will jump to a RET instruction which results in a "no operation" function.

Table 12-2. Disk Drive Port Addresses

PORT	REGISTER
3F2H	Floppy disk digital output port.
3F4H	Floppy disk controller status port.
3F5H	Floppy disk controller data port.
320H	Hard disk drive #1 port.
322H	Hard disk drive #2 port.

To access a particular drive, you must use one of the drive codes defined in Table 12-3 to identify it.

Table 12-3. Drive Identification Codes

CODE	DRIVE
00H	Floppy disk drive 1.
01H	Floppy disk drive 2.
02H	Floppy disk drive 3.
03H	Floppy disk drive 4.
80H	Hard disk drive 1.
81H	Hard disk drive 2.
82H	Hard disk drive 3.
83H	Hard disk drive 4.

Function Codes

To use the INT 13H instruction, place one of the function codes described in Table 12-4 in register AH. Function codes 06H and higher are used only for hard disk operation. They are included in this manual for information only, since they are not implemented in the Laptop computer. Each function code is described separately on the following pages.

Table 12-4. Disk Drive Function Codes

CODE	FUNCTION
00H	Reset disk system. Each disk in the system will move its head(s) to track 0.
01H	Read disk status. The status is returned in register AH.
02H	Read specified sector(s) into a buffer.
03H	Write buffer contents to specified sector(s).
04H	Test and verify that the specified sector(s) can be read without error.
05H	Format the specified track.
06H	Flag the specified track as bad.
07H	Format the remainder of the specified drive starting at a specified track.
08H	Return current rigid drive parameters.
09H	Initialize rigid drive characteristics.
0AH	Read the specified sector(s) and ECC bytes.
0BH	Write the specified sector(s) and ECC bytes.
0CH	Seek to the specified track.
0DH	Reset rigid drive controller for the specified drive.
0EH	Read rigid drive controller's sector buffer for the specified drive.
0FH	Write rigid drive controller's sector buffer for the specified drive.
10H	Test the status of the specified drive.
11H	Recalibrate the specified drive. The head(s) will be moved to cylinder 0.
12H	Execute controller diagnostics on the memory (buffer) of the specified drive.
13H	Execute internal diagnostics on the specified drive and drive to controller interface.
14H	Execute the controller's self-tests for the specified drive.

Table 12-9. Controller Parameters

BIT	DESCRIPTION
7	When this bit is set, disable retries on disk operations. 6When this bit is set, disable error correction code (ECC) operation.
3–5	Not used.
0–2	These three bits determine the step rate to be used by the disk drive: 000 = 3 microseconds. 100 = 200 microseconds. 101 = about 60 microseconds. 110 = 3 milliseconds. 111 = 3 milliseconds.

Function Code 0AH: Read Sectors and ECC Bytes — This function code will cause the interrupt routine to read the specified sectors into memory including the error correction code bytes. Refer to Table 12-10 for the parameters required for this operation.

Function Code 0BH: Write Sectors and ECC Bytes — This function code will cause the interrupt routine to write memory into the specified sectors including the error correction code bytes. Refer to Table 12-10 for the parameters required for this operation.

Table 12-10. Register Requirements for Function Codes 0AH and 0BH

REGISTER	CONTENTS
AH	This register must contain the function code (0AH or 0BH).
AL	The register must contain the number of sectors involved in the read or write operation, a value from 01H to 4FH.
ES:BX	These registers must contain the memory address of the disk buffer.
CH	This register must contain the cylinder number, a value from 00H to 3FFH.
CL	This register must contain the starting sector number of the operations. The two most significant bits of the cylinder number are to be placed in the two most significant bits of the register CL.
DH	This register must contain the head number, a value from 00H to 07H.
DL	This register must contain the rigid drive number, a value from 80H (for rigid drive 0) to 87H (for rigid drive 7.

Function Code 0CH: Seek a Track — This function code will cause the interrupt routine to seek (move the heads) to the specified track. Except for register AL, which is empty or may contain any value, refer to Table 12-7 for the CPU register requirements.

Function Code 0DH: Reset Controller — This function code will cause the interrupt routine to reset the rigid drive's controller values to the default values for the specified drive. Register DL must contain the drive number, a value from 80H to 87H.

Function Code 0EH: Read Sector Buffer — This function code will cause the interrupt routine to read the specified rigid drive's controller's sector buffer into memory. Register AL must be set to 1, registers ES and BX must contain the starting address in memory to which the sector buffer data will be transferred, and register DL must contain the drive number, a value from 80H to 87H.

Function Code 0FH: Write Sector Buffer — This function code will cause the interrupt routine to write the contents of memory into the specified rigid drive's controller's sector buffer. Register AL must be set to 1, registers ES and BX must contain the starting address in memory from which the data will be transferred, and register DL must contain the drive number, a value from 80H to 87H.

Function Code 10H: Test Drive Ready — This function code will cause the interrupt routine to check the status of the specified drive. Register DL must contain the drive number to be tested, a value from 80H to 87H. The status is returned in register AH. If the drive is ready, the status will be 00H.

Function Code 11H: Recalibrate Drive — This function code will cause the interrupt routine to recalibrate the specified drive (move the read/write heads to cylinder 0). Register DL must contain the drive number, a value from 80H to 87H.

Function Code 12H: Execute Controller Memory Diagnostic - - This function code will cause the interrupt routine to run a data-pattern test on the controller's internal memory buffer for the drive specified in register DL.

Function Code 13H: Execute Internal Diagnostics — This function code will cause the interrupt routine to run a test of the specified drive and drive-to-controller interface. These tests consist of a recalibrate command, followed by a series of seek commands. Sector 0 of head 0 is tested to see if it can read data successfully. These tests are not destructive; no write commands are performed. Register DL must contain the specified drive number, a value from 80H to 87H.

Function Code 14H: Controller Self-Test — This function code will cause the interrupt routine to run the controller's self tests for the specified drive. These tests include a check of the controller's processor, the data buffer, ECC circuits, and the checksum of the ROM. Register DL must contain the specified drive number, a value from 80H to 87H.

Error Status Codes

All of the rigid drive function calls will return a status code in register AH upon completion of the routine. Register AH and the carry flag CF will contain a 0 if the function was successful. If the function failed, the carry flag will be set (1) and register AH will contain a value representing a specific error. Table 12-11 defines the error codes.

Table 12-11. Disk Drive Error Codes

VALUE	DESCRIPTION
00H	No error has occurred.
01H	A bad (invalid) command was issued.
02H	The rigid disk controller could not find an address mark on the specified disk.
03H	This error indicates that a write (or format) operation was attempted on a write-protected disk.
04H	This error indicates that the specified record (or sector) could not be found.
05H	This error indicates that a hard disk controller reset failed.
07H	Controller would not accept drive parameters.
08H	This error indicates that a DMA overflow took place. This occurs when the DMA controller cannot keep up with the disk data transfer and, as a result, information is lost. It usually results from excessive DMA action from other devices on the data bus or else a hardware failure occurred within the bus interface.
09H	This code indicates that a DMA boundary error has occurred. The hardware is incapable of transferring sector buffer information across 64K memory boundaries. This error can be avoided by reducing the sector count in register AL or by changing buffer pointer so that entire transfer area resides within a single 64K memory segment.
10H	This code indicates that a bad CRC on a disk read operation has occurred. The specified record was found, but the cyclic redundancy check (CRC) for the data did not match the value calculated by the controller. On hard disk drives, this error flags any error that the ECC circuitry could not correct.
11H	On hard disk drives this code indicates that an ECC error has occurred, but the controller was unable to reconstruct the lost data.
20H	This code indicates that the disk controller IC has failed.
40H	This code indicates that the disk controller attempted to move the read/write head to a specified track, but could not find a matching sector header on that track (bad seek).
80H	This code indicates a timeout error. This occurs when a command has been issued to the controller, but was not completed within an amount of time specified by the hardware.
BBH	This code indicates that an undefined error has occurred. Usually, this is the result of a bad controller IC.
FFH	On hard disk drives, this code indicates that a sense drive status operation has failed.

Booting an Operating System (INT 19H)

The INT 19H interrupt will attempt to boot an operating system from the specified disk drive. Register DL must contain the drive number and if it is the hard disk drive, AL must contain a partition number between 0 and 3 expressed in ASCII format (that is, a value from 30H to 33H). If the value is 0, the boot track of the default partition will be selected.

The routine will attempt to read from track 0, sector 1 of the specified device, and execute the code once it is located and successfully loaded into memory. If the boot routine fails because the drive did not exist or because of a hardware failure, an error message is displayed and control is passed to the Monitor program.

Disk Parameters (INT 1EH)

The INT 1EH interrupt points to an area in memory that contains the disk parameters for the system. This area in memory initialized when the system is turned on and provides support for loading the operating system from the disk. The data stored in this area of memory is described in Table 12-12.

NOTE: The parameter tables will have to be modified if you use different disk drives than those supplied by Zenith Data Systems.

Table 12-12. Disk Parameter Table

BYTE	DESCRIPTION
1	Bits 0 – 3 contain the head unload time. The range is 16 to 240 milliseconds in 16-millisecond increments. Bits 4 – 7 contain the head step rate for the drive. The range is 1 to 16 milliseconds in 1-millisecond increments, in reverse order — 0FH = 1 millisecond, 0EH = 2 milliseconds, and so on.
2	Bits 1 – 7 contain the head load time (the range is 2 to 254 milliseconds in 2-millisecond increments). Bit 0 contains the DMA flag. If bit 0 = 0, the DMA mode will be used.
3	This byte contains the motor on timeout value. This is the amount of time the motor will remain on after the last disk operation and is measured in timer ticks.
4	This byte contains the number of data bytes per sector. See Table 12-13.
5	This byte contains the number of sectors per track.
6	This byte contains the gap length of gap 3 for read and write operations. The value of this byte is determined by the value in byte four. See Table 12-13.
7	If byte 4 (sector length) is 0, then this value is the length of the data that will be read from or written to each sector. If byte 4 is not 0, then set this value to FFH; it is ignored by the controller.

Function Code 02H: Set Cursor Position — This function code will cause the interrupt to place the cursor at the specified row and column location on the screen. Load register DH with the cursor row number and register DL with the column number. Rows are numbered from 0 to 24 and columns are numbered from 0 to 79 (80 characters per line) or 39 (40 characters per line). Therefore, when register DH is given a value of 0 and register DL is given a value of 0, the cursor will be placed in the upper left-hand corner of the screen. Also, load register BH with the video page number, which must be consistent with the video mode selected (only page 0 is valid when you are in the graphics mode).

Function Code 03H: Read Cursor Position — This function code will cause the interrupt to return the current cursor information as follows: the row number will be in register DH, the column number will be in register DL, the cursor's starting scan line number will be in register CH, and the cursor's ending scan line number will be in register CL. Prior to executing the interrupt, register BH must contain the page number.

Function Code 04H: Read Light Pen Position — A light pen option is not supported by the Laptop computer, even though this is a valid function code for PC-compatible computers.

The function code will cause the interrupt to attempt to obtain the light pen's position. After the interrupt has been executed, register AH will contain the light pen trigger/switch status (0 or 1). If register AH contains a 0, the switch on the light pen has not been pressed to trigger the light pen's operation. If register AH contains a 1, then the following information concerning the light pen's detected position will be returned: register DH will contain the row number, register DL will contain the column number, register CH will contain the scan line row number (0 to 199), and register BX will contain the pixel column number, which can be between 0 and 319 or 0 and 639, depending upon the graphics mode.

Function Code 05H: Select Active Display Page — This function code will cause the interrupt to display the page specified in register AL. In the text modes, unused portions of video memory can be used for additional video pages. In video modes 0 and 1, the valid page numbers are 0 to 7; in video modes 2 and 3, they are 0 to 3. In the graphics modes, only page 0 is valid.

Function Code 06H: Scroll an Area of the Screen Up — This function code will cause the interrupt to scroll the specified area of the screen up the specified number of lines. Prior to executing this interrupt, the following registers must be loaded: register CX must contain the upper left-hand coordinates (place the row number in register CH and the column number in register CL), register DX must contain the lower right-hand coordinates (place the row number in register DH and the column number in register DL), register BH must contain the attribute byte for the blank lines, and the number of lines to be scrolled must be placed in register AL. If register AL is loaded with a 0, the entire window will be cleared.

NOTE: If a hardware or smooth scrolling mode is currently active when this interrupt function is executed, they will function only if the entire screen is scrolled. Hardware or software scrolling will not work on a portion of the screen.

Function Code 07H: Scroll an Area of the Screen Down — This function code will cause the interrupt to scroll the specified area of the screen down the specified number of lines. Prior to executing this interrupt, the following registers must be loaded: register CX must contain the upper left-hand coordinates (place the row number in register CH and the column number in register CL), register DX must contain the lower right-hand coordinates (place the row number in register DH and the column number in register DL), register BH must contain the attribute byte for the blank lines, and the number of lines to be scrolled must be placed in register AL. If register AL is loaded with a 0, the entire window will be cleared.

Function Code 08H: Read Character and Attribute — This function code will cause the interrupt to return the character and attribute codes for the character that resides at the current cursor position. In text modes, register BH must contain the video page number. Upon completion of the routine, the character code will be in register AL and the attribute code will be in register AH.

Function Code 09H: Write Character and Attribute to Screen — This function code will cause the interrupt to write the specified character and attribute codes to the cursor location. Place the character code in register AL, the attribute code in register BL, the number of times the character is to be repeated in register CX, and in text modes, the page number in register BH.

Function Code 0AH: Write Character to Screen — This function code will cause the interrupt to write the specified character to the screen; but not the attribute byte. Place the character code in register AL, the number of times to repeat the character in register CX, and in text modes, the page number in register BH.

Function Code 0BH: Set Color Palette — This function code is only available in mode 4, the 320 × 200 graphics mode. Place a value from 0 to 127 in register BH (see the following text), and a value from 0 to 4 in register BL.

If the value placed in register BH is even, then the current background color will be used for the foreground color, normally a value between 0 and 31. Values above 15 will select the intensified level of the 16 colors.

If the value placed in register BH is odd, one of the two available palettes will be selected by the value placed in register BL. The palette and pixel color number will determine the foreground color. If the value is 0, then palette 0 will be selected; if the value is 1, then palette 1 will be selected. Refer to Table 13-4 for the palette number and color number matrix.

Table 13-4. Palette and Pixel Colors

COLOR NUMBER	PALETTE 0	PALETTE 1
1	Green	Cyan
2	Red	Magenta
3	Yellow	White

Function Code 0CH: Write Graphics Pixel — This function code will cause the interrupt to light a single pixel at the specified location on the screen. Place the pixel row number in register DX, the pixel column number in register CX, and the color in register AL. Rows are numbered 0 to 199. Columns are numbered 0 to 319 or 0 to 639. Colors are numbered 0 to 3 in medium-resolution (320 × 200) or 0 to 1 in high-resolution (640 × 200). In all cases, 0 is the background color (usually black). In the high-resolution mode, 1 is the foreground color. In the medium-resolution mode, the color is determined by the pixel and the color number, as described in Table 13-4.

If the most significant bit (bit 7) of AL is set, the color will be XORed with the current color permitting simple animation.

Function Code 0DH: Read Graphics Pixel — This function code will cause the interrupt to return the color of the pixel at the specified location in register AL. Place the pixel row number in register DX, the pixel column number in register CX, and the color in register AL.

Function Code 0EH: Dumb Terminal Display — This function code will cause the interrupt to treat the character as if it were sent to a dumb terminal. That is, the back space (08H), carriage return (0DH), line feed (0AH), and bell (07H) will be treated as console commands rather than characters to be used for screen formatting. Furthermore, if a character is to be printed at the end of a screen line, the cursor will be positioned at the start of the next line. If a line feed is performed on the last display line of the screen, or if a character is printed at the last position of the last line, the screen will be scrolled up one line. When scrolling, the attribute for the new row (when in text mode) will be the same as the attribute of the character at the cursor position on the line when the scrolling takes place.

Place the character in register AL, the foreground color in register BL (for graphics modes), and the display page number in register BH.

Function Code 0FH: Return Video State — This function code will cause the interrupt to return the current video state. Register AL will contain the current video mode (see Function Code 0), register AH will contain the screen width in columns, and register BH will contain the active video page number.

Function Code 64H: Set Scrolling Mode — This function code will cause the interrupt to select one of three scrolling modes. Place the scrolling mode value in register AL. Mode 0 is software scrolling, mode 1 is hardware "jump" scrolling, and mode 2 is hardware "smooth" scrolling. Keep in mind the following limitations.

- Hardware scrolling will not work in the 40 × 25 text modes (mode 0 and mode 1).
- Hardware smooth scrolling works only in the high-resolution graphics mode (mode 6).
- Hardware jump scrolling works only in the graphics (mode 4, 5, and 6) and 80 × 25 (mode 3) mode.
- Software scrolling will work in all modes. If you write software that bypasses the Monitor program, use software scrolling.

Video Initialization (1DH)

The INT 1DH instruction, unless otherwise programmed by an application program, will initialize the video section parameters according to the information stored in the Monitor program ROM. This is the same data used to initialize the video section of the computer when the system is first turned on.

Four tables are required to properly define the video section by this interrupt. Keep in mind the video mode 7, the monochrome text (TTL) video mode, is not used by the Laptop computer. The values for each register in the 6845 register set are defined in Table 13-5. Refer to Chapter 14, "Hardware", for a discussion on the 6845 register set in the V6355 video controller.

Table 13-5. Video Initialization Default Values

REGISTER NUMBER	TEXT 40 × 25	TEXT 80 × 25	GRAPHICS	MONOCHROME TEXT (TTL)
R0	38	71	38	61
R1	28	50	28	50
R2	2D	5A	2D	52
R3	0A	0A	0A	0F
R4	1F	1F	7F	19
R5	06	06	06	06
R6	19	19	64	19
R7	1C	1C	70	19
R8	02	02	02	02
R9	07	07	01	0D
R10	06	06	06	0B
R11	07	07	07	0C
R12	00	00	00	xx
R13	00	00	00	xx
R14	xx	xx	xx	xx
R15	xx	xx	xx	xx
R16	xx	xx	xx	xx
R17	xx	xx	xx	xx

NOTE: All values are expressed in hexadecimal. xx represents any value between 00H and FFH.

Defining Characters (1FH)

The INT 1FH instruction allows access to an extended character set for use with the medium- and high-resolution color graphics modes.

Normally the first 128 characters (00H – 7FH) used in the graphics modes are supplied by the character generator ROM. In addition to these characters, you can create a custom character set of 128 additional characters (80H – FFH) using the following procedure.

1. Allocate a 1K section of memory (not video memory) to hold the character set. Eight bytes will be needed for each character your create.

2. Define each character in an 8 × 8 matrix as shown by the 7 in Figure 13-1. Since the top line in the matrix butts up against the bottom line of the character above it, you will need to allow for ascenders and descenders. Also, do not forget to allow for a space between characters.

3. For each line of the matrix, identify which of the 8 pixels will be lit.

4. With the first pixel as the most-significant bit, add the binary-based hexadecimal weight of the lit pixels in each row to produce a hexadecimal value for the byte representing that row.

5. Load all eight bytes that form the defined character into the first eight bytes of the memory allocated for the character set.

6. Repeat this procedure for each of the 128 characters in the set.

7. Once the characters have been defined and placed into memory, set the pointer for interrupt 1FH (memory location 0000:007C) to the start of memory allocated for the defined character set.

8. Then, whenever you use INT 10H to display a character code between 80H and FFH in graphics mode, the character from your set will be used.

	BINARY VALUE								RESULTING
ROW	80H	40H	20H	10H	08H	04H	02H	01H	VALUE
1			•	•	•	•	•		3EH
2							•		02H
3							•		02H
4						•			04H
5					•				08H
6				•					10H
7				•					10H
8									00H

Figure 13-1. Character Design Matrix

Chapter 14

Hardware

This section of the manual describes the theory behind the circuits that make up the Laptop computer. This chapter contains the overall theory of operation supported at block diagram level.

Computer Organization

Refer to Figure 14-1. The computer is made up of four major sections (the power control, processor, memory, and peripheral support sections). Each of the major sections contain a number of major integrated circuits.

Three major integrated circuits control the computer's operation, from the first time the power is turned on, through the operation of the built-in self-tests, during any basic input/output operations, and during the idle states, when "nothing" appears to be happening. These integrated circuits are the ROM, which contains the Monitor program; the CPU gate array, which contains a large portion of the logic of the computer; and the decoder gate array, a programmed logic array that provides most of the chip select signals for the rest of the system. Two other gate arrays, the system bus buffer gate array and the printer gate array, greatly reduce the number of discrete circuits required in the computer.

The processor section consists of the CPU (an 80C88) and optional numeric processor extension (an 8087), and the support integrated circuits. The CPU is often called the heart of the computer, although it couldn't function without the control section to guide its first few steps. Between the CPU and numeric processor extension, all the calculations and sophisticated processing takes place. The support ICs take the signals generated by the CPU and coprocessor and cause other sections of the computer to function. Some of these devices, such as the interrupt controller, will act analogous to a secretary, feeding the CPU only the most important messages, which will cause it to interrupt its normal operation to expedite a more important process.

The memory section contains most of the read/write memory in the computer. This memory is sometimes called user memory, although it is more commonly called RAM. There are two distinct sections of memory: scratchpad memory, used by the CPU for intermediate calculations and for temporary storage space (buffers); and contiguous memory, which is divided into 128K of internal memory on the main board, and 512K of expansion memory, which is plugged into the main board.

The peripheral support section consists of the video controller and memory; the floppy disk controller; and the input/output sections, which include the keyboard, the internal modem, the parallel printer port, and the serial input/output port.

The power control section provides operating voltages to power the logic devices used in this computer. These circuits also develop the required voltage to operate the LCD display.

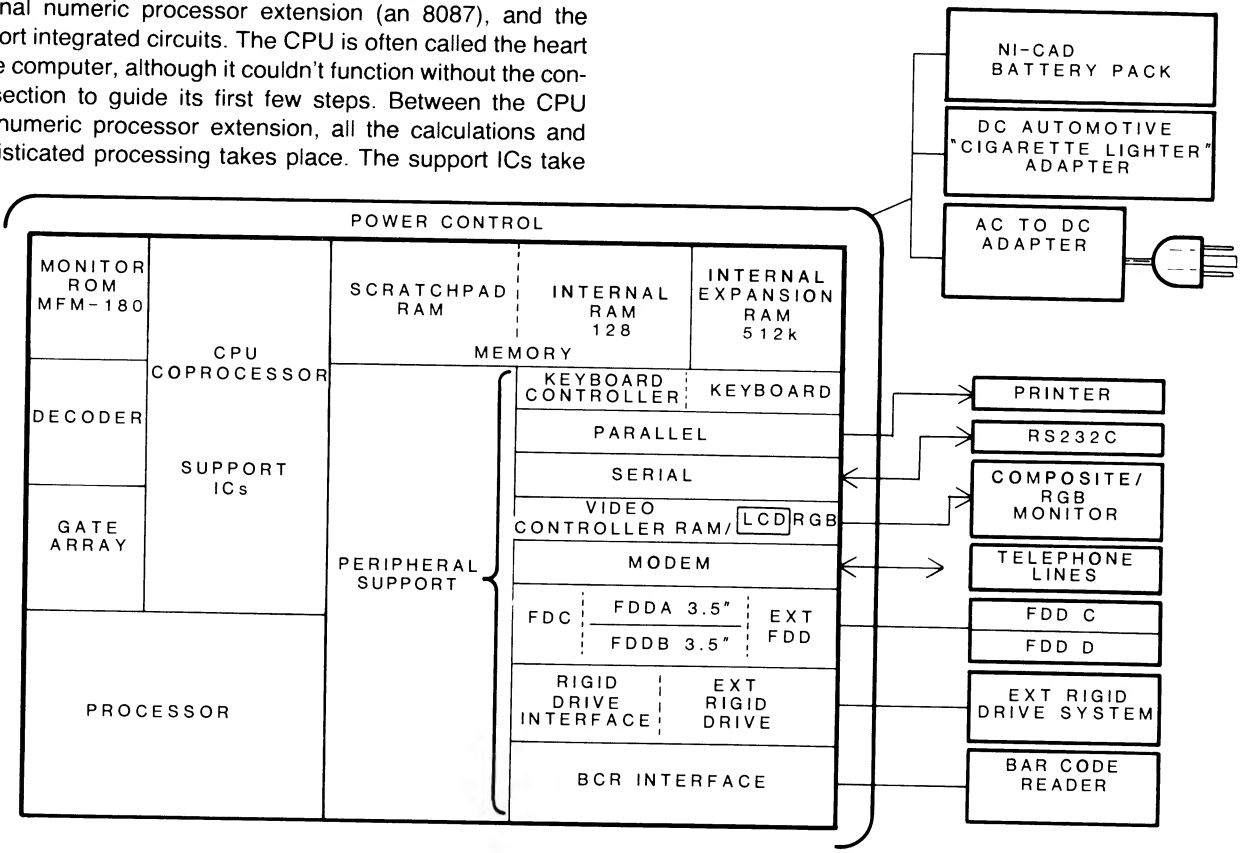

Figure 14-1. Basic Laptop Computer Design

External Connectors

Refer to Figure 14-2. The following external cable connectors are located behind an access panel on the rear of the computer.

- Hard Disk Drive Connector — This connector provides IBM PC-compatible bus signals to an external hard disk drive.

- External Floppy Disk Drive/Bar Code Reader Connector — This connector provides the signals necessary to connect an external disk drive or bar code reader to the computer.

- Video Connector — This connector supplies RGB video output for an external color video monitor.

- Parallel Connector — This connector provides Centronics-type output signals for a parallel printer or peripheral. The operating system contains the necessary software to properly configure this connector for use with most parallel devices.

- Serial Connector — This connector is an RS-232 DTE input/output port for use with a serial printer or peripheral. The operating system contains the necessary software to properly configure the connector for use with most serial devices.

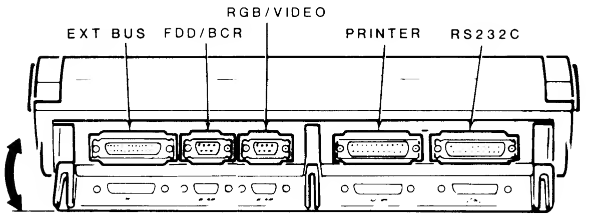

Figure 14-2. Back View

Refer to Figure 14-3. The following connectors are located on the left side of the computer.

- Line Connector — This connector supplies the necessary signals for telephone computer-to-computer communication. It is used to connect the internal modem to the telephone company line.

- Telephone Connector — This connector supplies the necessary signals for a standard telephone when the telephone company line is connected to the Line Connector.

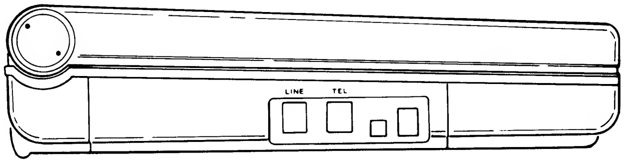

Figure 14-3. Left Side View

Refer to Figure 14-4. The following are located on the bottom of the computer.

- Battery Compartment — The internal battery pack is inside this compartment. The cover is secured with four screws.

- Expansion Memory — Up to 512K of expansion memory can be installed in this area. When combined with the 128K of internal memory, the system has a total of 640K of user memory, the capacity of PC-compatible systems and software.

- Configuration Switch — Behind a small rubber cover is the configuration DIP switch. This switch is described in the configuration section of Chapter 7.

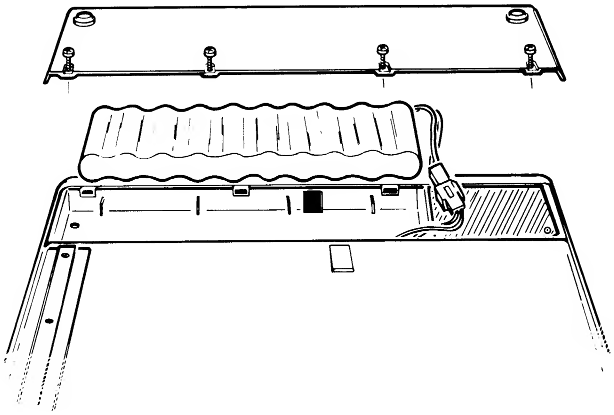

Figure 14-4. Bottom of the Computer

Power Control

The power supply section provides the voltages used by the various circuits in the computer. Power for the supply is received from an external adapter (AC line or DC automotive) or an internal battery pack. The supply is divided into seven circuits that provide battery charging, low voltage detection, LCD drive, LCD backlight, +5 VDC (logic), +5 VDC (floppy interface), −5 VDC (RS-232 port), and −7 VDC.

The battery recharge circuit provides a high current charge to the battery pack depending on the state of discharge of the battery pack. The operation can take place when the computer is turned off and during operation as well. The recommended charge time is 8 hours for a dead battery. Do not exceed 12 hours charging time or the life of the battery pack can be shortened considerably.

The low voltage detection circuit is controlled by the output voltage of the battery. When this voltage drops below a preselected level, the output will trigger the low power warning indicator.

The LCD drive voltage circuits allow information to be displayed on the LCD. The LCD driver circuit supplies power to the LCD driver board. If no power is supplied, no information will be displayed. In this Laptop computer, the user has the option to display on the LCD of the computer or on an external monitor. Control is provided in two ways: by setting a configuration switch and by toggling the LCD/RGB key on the keyboard.

The backlight power supply output is the only AC output of the power supply board. The backlight requires an AC signal of about 750 Hz whose voltage ranges from 20 VAC (dim backlight) to 80 VAC (bright backlight). The backlight circuit consists of three parts: the drive, which includes the brightness control; the on/off control; and the oscillator, which produces the output to the backlight.

Processor and Support Circuits

There are four major circuits in the processor and control section of the computer: Clock, which is necessary to run the CPU and coprocessor; Control, which includes the interrupt signals and circuit selection signals; Addressing, so that memory and various input/output devices and ports may be addressed; and Data, to pass data to and from the various parts of the computer. Because of the complexity of these circuits, they are discussed separately.

Refer to Figure 14-5. This simplified block diagram shows how the four major circuits relate to each other.

The clock circuit part of the CPU gate array, represented here by the CLK signal line, provides the processor block with the basic timing signals for the operation of the computer. The CPU produces the S0 – S2 timing signals, which feeds the gate array. The timer block develops the TRM1 and TRM2 signals which are used with the S0, S1, and S2 signals to produce the control signals for the rest of the system. The control signals are used by the decoder to produce the chip select signals, which make up part of the control bus. The gate array supplies the remainder of the signals that make up the control bus.

The processor block, consisting of the 80C88 CPU and 8087 coprocessor, along with the gate array, is the source for all addressing within the system. The first eight lines are multiplexed with the data lines and are bidirectional. They are split into the appropriate address and data lines by the bidirectional data buffer and unidirectional address buffer. The gate array uses the DMA address buffers to set up DMA addressing over the address bus.

CLOCK is a real-time clock and is selected by address lines (not shown). It feeds or receives data from the address-data lines.

The interrupt controller block is fed one of the outputs of the timer block to initiate an INT 08H instruction in the CPU 18.2159 times per second or every .0549254 seconds (see "Timer (Time-of-Day)" in Chapter 9. The interrupt controller handles all hardware initiated interrupts and routes them to the CPU for processing.

The other interrupt line is the NMI (non-maskable interrupt) line. It is initiated by the gate array and usually indicates that a power-down condition has started.

The major ICs in the processor and support circuits section of the computer include the 80C88 CPU, the optional 8087 numeric processor extension, the RP5C15 real-time clock, the 82C59 interrupt controller, the 82C53 programmable interval timer, the gate array, and the decoder. The following section describes the major purpose(s) for each of these integrated circuits. The various devices used as buffers are also described.

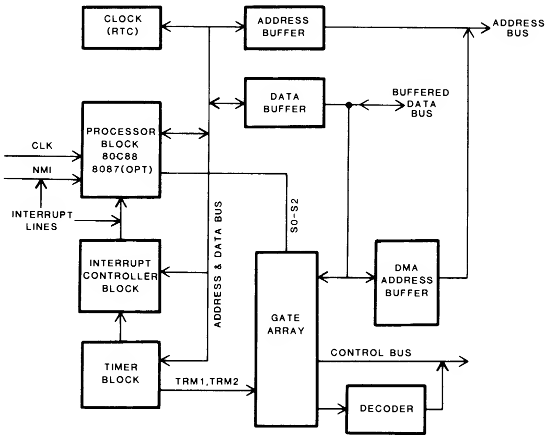

Figure 14-5. Main Logic Board Block Diagram

The 80C88 CPU

The 80C88 is a low-power third-generation microprocessor with a 16-bit internal architecture and an 8-bit external data path to memory and input/output ports. This processor's most outstanding characteristic is that it is suitable for a wide spectrum of computer applications. In addition, the processor is designed to operate with a numeric data (math) coprocessor. The design is substantially more powerful than previous 8-bit microprocessors, and when compared to the older 8080, it represents an improvement of four to six times, depending upon the application. The higher performance is due to a pipeline architecture where instructions may be prefetched during spare bus cycles.

Microprocessors usually execute programs by repeatedly cycling through the following series of steps.

1. Fetch the next instruction from memory.
2. Read the operand, if required by the instruction.
3. Execute the instruction.
4. Write the result of the instruction, if required.

Second-generation CPUs, which do not have the pipeline design, usually perform these steps in a serial manner, or at the most, with a single bus cycle fetch overlap. The 80C88 design contains two separate, independent, internal processing units (the execution unit and the bus interface unit), which share the previously-listed steps. The execution unit executes the instructions, while the bus interface unit fetches instructions, reads operands, and writes the results of the instruction. This mode of operation results in overlapping fetch and execution steps, decreasing the amount of time required to perform a series of instructions.

The execution unit is a 16-bit arithmetic logic unit (ALU) that maintains the CPU status and control flags, and manipulates the general registers and instruction operands. All internal register and data paths are 16 bits wide. Instructions to be executed are transferred from a queue that is maintained by the bus interface unit. Access to memory or an input/output port is requested through the bus interface unit, which retrieves or stores the data.

The bus interface unit performs all bus operations for the execution unit and during periods when the execution unit is busy, the bus interface unit "looks ahead" and fetches (retrieves) instructions from memory, which are stored in an internal RAM queue.

There are eight 16-bit general purpose registers, four segment register, and an instruction pointer inside the CPU. The general purpose registers are divided into two sets of four registers each. One set is identified as the data registers, which are sometimes called the H & L group, for "high" and "low." The other set is called the pointer and index registers, sometimes called the P & I group. Refer to Table 14-1 for a description of each register.

Table 14-1. 80C88 Internal Registers

MNEMONIC	DESCRIPTION
AX	Accumulator register. This register is also divided into the 8-bit AH (high) and AL (low) registers. Specifically used by word multiply, word divide, and word input/output instructions.
AH	The high 8 bits of register AX. Specifically used by byte multiply and byte divide instructions.
AL	The low 8 bits of register AX. Specifically used by byte multiply, byte divide, byte input/output, translate, and decimal arithmetic instructions.
BX	Base register. This register is also divided into the 8-bit BH (high) and BL (low) registers. Specifically used by translate instructions.
BH	The high 8 bits of register BX.
BL	The low 8 bits of register BX.
CX	Count register. This register is also divided into the 8-bit CH (high) and CL (low) registers. Specifically used by string operations and loops.
CH	The high 8 bits of register CX.
CL	The low 8 bits of register CX. Specifically used by variable shift and rotate instructions.
DX	Data register. This register is also divided into the 8-bit DH (high) and DL (low) registers. Specifically used by word multiply, word divide, and indirect input/output instructions.
DH	The high 8 bits of register DX.
DL	The low 8 bits of register DX.
SP	Stack pointer register. This 16-bit register is used specifically by stack operations.
BP	Base pointer register. This 16-bit register is not specifically used by any instruction.
SI	Source index register. This 16-bit register is used specifically by string operations.
DI	Destination index register. This 16-bit register is used specifically by string operations.
CS	Code segment register. This 16-bit register points to the base address of the current code segment. CPU instructions are fetched from this segment.
DS	Data segment register. This 16-bit register points to the base address of the current data segment, which usually contains variables for the program that is being executed.
SS	Stack segment register. This 16-bit register points to the base address of the current stack segment. This segment holds the stack for the CPU.
ES	Extra segment register. This 16-bit register points to the base address of the current extra segment, which is usually used for data storage.
IP	Instruction pointer. This 16-bit register is similar in purpose to the program counter in the 8-bit CPU designs. Its contents point to the next instruction and are updated by the bus interface unit. Normally the next instruction is the one to be fetched by the bus interface unit. However, if the instruction pointer is saved on the stack, it will be adjusted so that its contents point to the next instruction to be executed.

NOTE: All eight general purpose registers fit the definition of accumulator as defined for 8-bit CPUs. Many instructions may be programmed to use registers in addition to those specifically mentioned in this table.

The four 16-bit data registers may be addressed as either a 16-bit registers or as two separate 8-bit registers. They may be freely used in most arithmetic and logic operations, but some instructions use specific registers. The four 16-bit pointer and index registers can also be used by many arithmetic and logic operations. With the exception of the base pointer register, the other three registers are specifically used by some instructions. Refer to Figure 14-6 and Table 14-2.

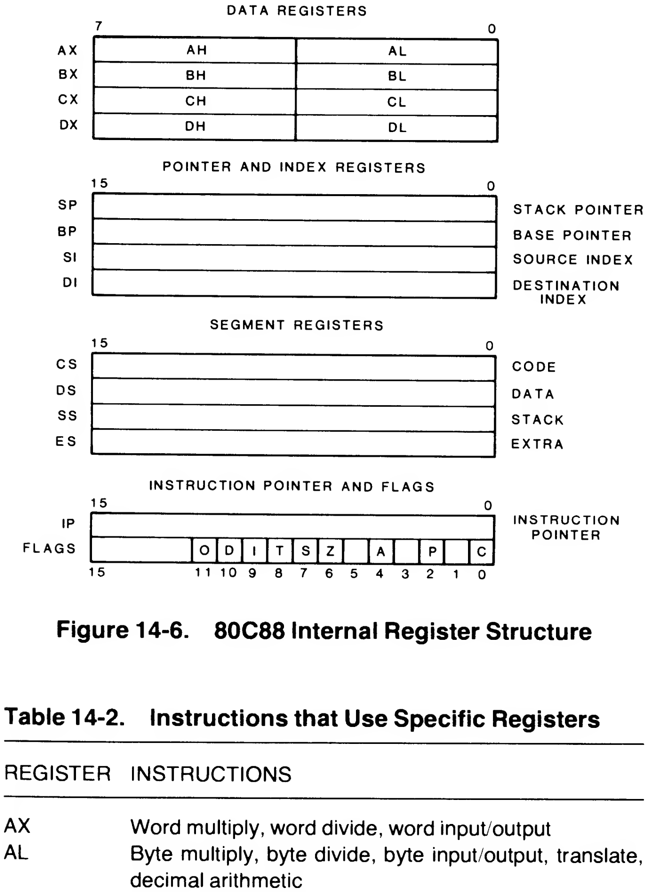

Figure 14-6. 80C88 Internal Register Structure

Table 14-2. Instructions that Use Specific Registers

REGISTER	INSTRUCTIONS
AX	Word multiply, word divide, word input/output
AL	Byte multiply, byte divide, byte input/output, translate, decimal arithmetic
AH	Byte multiply, byte divide
BX	Translate
CX	String operations, loops
CL	Variable shift and rotate
DX	Word multiply, word divide, indirect input/output
SP	Stack operations
SI	String operations
DI	String operations

The 80C88 can address up to a megabyte of memory, but not all at the same time. In order to most effectively use memory, The megabyte of addressable memory is divided into 64K sections, called segments. A segment may start on any 16-byte memory boundary and contain up to 64 kilobytes in it. The CPU has direct access to four segments at any one time. The base (starting) address of each segment is stored in its corresponding segment address register in the CPU (see Figure 14-6). The segments may be adjacent, disjointed, partially overlapping, or fully overlapping. A physical memory location may be addressed by one or more of the segments, each having the same or a different value, depending upon the value of the base address for that segment.

Finally, there are six 1-bit status flags and three 1-bit control flags. These are described in Table 14-3.

Table 14-3. CPU Flags

MNEMONIC	DESCRIPTION
CF	The carry flag, when set, indicates that a carry out of or a borrow into the high-order bit of the result has taken place. Rotate instructions can also isolate a bit from a memory location or a register by rotating it into the carry flag.
AF	The auxiliary carry flag, when set, indicates a carry or a borrow has taken place. This flag is used by decimal arithmetic instructions. The flag can indicate a carry out of the low nibble into the high nibble or a borrow from the high nibble into the low nibble of an 8-bit value.
OF	If the overflow status flag is set, an arithmetic overflow has occurred. The CPU supports an interrupt on overflow instruction so that the user can recognize when an overflow takes place.
SF	The sign status flag indicates the sign of the result. If set, the result is negative (negative binary numbers are represented in standard two's complement notation). If this flag is not set, the result is positive.
PF	The parity flag, when set, indicates that the result has even parity (an even number of set bits). This flag can be used to check for data transmission errors.
ZF	The zero status flag, when set, indicates that the result of the operation is 0.
DF	The direction flag is a control flag. Setting it causes string instructions to automatically decrement so that strings are processed from right to left. If the flag is not set, string instructions are automatically incremented so that strings are processed from left to right.
IF	The interrupt-enable flag is a control flag. Setting it allows the CPU to recognize external maskable interrupts. If this flag is clear (not set), the interrupts will be ignored. This flag does not affect non-maskable interrupts.
TF	The trap flag is a control flag. Setting it puts the processor into a single step mode, where the CPU automatically generates an internal interrupt after each instruction is executed. This allows the user to inspect the program after each instruction on a step-by-step basis.

The 80C88 communicates with the rest of the computer by using the three types of signals described below. Various other signals are multiplexed and require latches to separate the information.

- Address Signals — Twenty address lines make up the address bus. It allows the CPU to address 1 megabyte of memory, sixteen address lines are used to address input/output ports, and eight address lines are multiplexed with the eight data lines.
- Data Signals — Eight data lines make up the data bus. The 80C88 communicates with the rest of the computer over this bus. These signal lines are multiplexed with the 8 low-order address lines before being separated by bi-directional buffers.
- Control Signals — The control signals generated in the 80C88, gate array, and decoder, are numerous. These are the commands that the rest of the computer carries out and includes various read, write, and timing signals.

The CPU carries out its instructions, generally with one or more functions taking place: fetch (obtaining information over the data lines), execute (manipulating the data in some manner), write (sending data to memory or a peripheral device), or read (receiving data from memory or a peripheral device). You may want to draw a parallel between the read and fetch functions. However, their purpose is slightly different: the fetch function is used to obtain the next instruction where the read function is used to obtain data.

The instruction set for this processor consists, in effect, at two levels: assembly language level and machine language level. At the assembly language level, there are about 100 instructions. But, at the machine language level, there are about 300 instructions. The assembly language compiler takes the various instructions and translates them into the necessary machine language instruction to implement correct operation by the CPU

As an example of this, there is one basic MOV instruction in assembly language. It can transfer a byte or word from a register, a memory location, or an immediate value, to either a register or memory location. At the machine language level, there are 28 different MOV instructions that accomplish these various tasks. It is much easier for a programmer to remember one general instruction and how to use it, than to have to remember many smaller, but very specific instructions.

The CPU is a very complex device and as such, it must be started in an orderly manner. The RESET signal does this, as explained in the detailed circuit description, later in this chapter. When the CPU detects a positive-going edge on the RESET line, all activities are terminated. When the line goes low, the CPU is initialized and the internal flags are cleared; the instruction pointer, DS, SS, and ES registers have a value of 0000H; the CS register has a value of FFFFH, and the queue is empty.

Since the instruction pointer contains 0 and the code segment (CS) register contains FFFFH, the processor fetches its first instruction from memory location FFFF0H. This location is in the multifunction monitor ROM and contains a JMP instruction to the start of the initialization routines. For a complete map of the computer's memory and port addresses, see the Chapter on system memory.

The first thing the fetch operation does is to load the contents of the instruction pointer into the address lines through the bus interface unit. As the address output lines become stable, the $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ lines are placed in their correct state ($\overline{S0}$ and $\overline{S1}$ are high and $\overline{S2}$ is active-low). This combination of signals is interpreted by the gate array to produce the AEN1 signal (it goes active-high), which latches the address into the address buffers. The bus interface unit then allows the instruction pointer to be incremented so that it contains the location of the next instruction.

After the address is latched, $\overline{RD}$ and $\overline{S0}$ goes active-low. This new combination of $\overline{S0}$ and $\overline{S2}$ low and $\overline{S1}$ high is translated by the gate array to assert (make active) $\overline{DT/R}$ low. When this happens, the data from the addressed memory is sent through the data buffer back to the CPU where the bus interface unit receives it.

The instruction received is then placed in the instruction queue, where the execution unit can retrieve it to execute, or, if it is data, manipulate it.

The JMP instruction in the multifunction monitor ROM is a JMPF (jump far) instruction to the beginning of the multifunction monitor ROM routines. Therefore, the processor will need to obtain the next four bytes of data to determine the location in memory. Some instructions require two or more bytes to operate, as is the case here. The CPU will repeat the fetch cycle as often as needed to load the remaining parts of the instruction.

The instruction cycle determines what must be done, besides fetching instructions. Operations may involve read from or writing to memory, which are similar to the fetch operation, or they may cause some internal operation to take place.

The $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ lines carry status signals which identify the type of bus cycle that the CPU will require. Table 14-4 describes the types of bus signals as determined by the states of these three lines.

Table 14-4. CPU Bus Cycle Type as Determined by $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$

$\overline{S0}$	$\overline{S1}$	$\overline{S2}$	BUS CYCLE TYPE
L	L	L	Interrupt acknowledge
L	L	H	Read input/output
L	H	L	Write input/output
L	H	H	Processor Halt
H	L	L	Instruction fetch
H	L	H	Read memory
H	H	L	Write memory
H	H	H	Passive, no bus cycle required

The Optional 8087 Numeric Processor Extension

An optional 8087 numeric processor extension may be added to the computer by a qualified service center. This device is installed on the main board next to the CPU. The 8087 is an arithmetic logic unit (ALU) processor that contains expanded 80-bit registers, an expanded instruction set, and internal logic to perform complex mathematical operations with less programming than required to perform the same task with only the 80C88 CPU.

Because the 80C88 requires more program steps to perform the same functions as the 8087, more time is required. In some real-time operations the 80C88 alone may not be able to keep up with the application. The 8087, assuming the software uses it, dramatically increases system throughput where mathematical functions are required by the program.

The 8087 shares many of the input, output, and bidirectional signals of the 80C88. This allows it to monitor the same instructions fetched by the CPU. Since the two integrated circuits are designed to work side by side, the CPU ignores 8087 instructions and the 8087 ignores CPU instructions; each processes only its own instructions.

Like the 80C88, the 8087 is internally divided into to two processing units. These are the control unit and the numeric execution unit. The numeric execution unit performs all numeric instructions while the control unit is responsible for fetching instructions, reading values from and writing values to memory, and executing control instructions. The control unit also maintains synchronization with the 80C88 CPU. Because the 8087 is considered an extension of the 80C88, it does not have registers in the sense that the 80C88 does. Table 14-5 details the bus cycles that occur with respect to the status of the $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ status signals.

Table 14-5. 8087 Bus Cycle as Determined by $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$

$\overline{S0}$	$\overline{S1}$	$\overline{S2}$	BUS CYCLE
H	L	H	Read memory
L	H	H	Write memory
H	H	H	Passive; no bus cycle

The RP5C15 Real-Time Clock

The Laptop computer has a real-time clock IC that is connected directly to the address and unbuffered data bus of the system. Data can be written to and read from the IC in the same manner as RAM is written to and read from. A 1 farad "super-capacitor" maintains the time and 26, 4-bit segments of data in non-volatile memory over several days while power is turned off.

Internal counters track time in hours, minutes, and seconds, and the date in days, months, and years. The day of the week is calculated, and leap years, occurring every four years are taken into account. There is no logic to handle the omission of leap years every 100 years (except that the year 2000 is a leap year). Therefore, the calendar chip will be accurate up through the year 2099. Since the year 2100 is not a leap year, the clock chip will not be accurate in that year (it will produce a February 29th).

The chip contains 51, 4-bit internal registers that are used as counters, control registers, and non-volatile memory. Four modes control the access to these registers. All data is stored in BCD (binary coded decimal) format. The clock is set up to function as the clock for the computer system, therefore reading and writing to the clock should be handled through the normal BIOS interrupts.

Registers 0 – D are read/write registers. Register E and F are write-only registers and will return a zero when you read either one. Registers D – F are common to all four modes. The function of each register is described in the following four tables. See Table 14-6 for mode 0; Table 14-7 for mode 1; Table 14-8 for mode 2; and Table 14-9 for mode 3.

An “x” in the bit column of a table means that the bit may be set or not set; it determines the value in the counter or register when taken in conjunction with the other similarly-marked bits for the same register. A “ – ” means that the value may be written to, but will always produce a 0 (not set) when read by the system.

Table 14-6. Real-Time Clock Registers in Mode 0

REG.		BIT			
NO.	D3	D2	D1	D0	DESCRIPTION
0	x	x	x	x	Ones of seconds counter. The range is 0 to 9.
1	–	x	x	x	Tens of seconds counter. The range is 0 to 5.
2	x	x	x	x	Ones of minutes counter. The range is 0 to 9.
3	–	x	x	x	Tens of minutes counter. The range is 0 to 5.
4	x	x	x	x	Ones of hours counter. The range is 0 to 9.
5	–	–	x	x	Tens of hours counter. When the clock is operating in the twelve-hour mode, the range is 0 to 1 and if bit D1 is set, then the time is PM; if D1 is clear, then the time is AM. When the clock is operating in the twenty-four hour mode, the range is 0 to 2.
6	–	x	x	x	Day of the week counter. The range is 0 to 6.
7	x	x	x	x	Ones of days counter. The range is 0 to 9.
8	–	–	x	x	Tens of days counter. The range is 0 to 3.
9	x	x	x	x	Ones of months counter. The range is 0 to 9.
A	–	–	–	x	Tens of months counter. The range is 0 to 1.
B	x	x	x	x	Ones of years counter. The range is 0 to 9.
C	x	x	x	x	Tens of years counter. The range is 0 to 9.
D	x	x	x	x	Clock enable, alarm enable, and mode selection register. If bit D3 is set, the clock is allowed to run; if clear, the clock will not run. If bit D2 is set, the alarm is enabled; if clear, the alarm is disabled. If bits D1 and D0 are clear, mode 0 will be selected; if bit D1 is clear and bit D0 is set, mode 1 will be selected; if bit D1 is set and bit D0 is clear, mode 2 will be selected; and if bits D1 and D0 are both set, mode 3 will be selected. For normal operation in this computer, the clock is enabled and the alarm is disabled.
E	x	x	x	x	Test register. This register is used during testing by the chip manufacturer for high-speed function testing. If this register is not clear, the clock will not operate properly.
F	x	x	x	x	Reset controller. Bit D3 must be clear to enable the 1 Hz signal out the $\overline{\text{ALARM}}$ pin. Bit D2 must be clear to enable a 16 Hz signal. Bit D1, when set will reset all the clock's internal counters. Bit D0, when set will reset the alarm's internal counters (see Table 14-7). For normal operation in this computer, only bit D2 of this register is set.

Table 14-7. Real-Time Clock Registers in Mode 1

REG.		BIT			
NO.	3	2	1	0	DESCRIPTION
0	–	–	–	–	Not used.
1	–	–	–	–	Not used.
2	x	x	x	x	Alarm ones of minutes counter. The range is 0 to 9.
3	–	x	x	x	Alarm tens of minutes counter. The range is 0 to 5.
4	x	x	x	x	Alarm ones of hours counter. The range is 0 to 9.
5	–	–	x	x	Alarm tens of hours counter. The range is 0 to 1 when the clock is in the twelve hour mode and 0 to 2 when the clock is in the twenty-four hour mode.
6	–	x	x	x	Alarm day of the week register. The range is 0 to 6.
7	x	x	x	x	Alarm ones of days register. The range is 0 to 9.
8	–	–	x	x	Alarm tens of days register. The range is 0 to 3.
9	–	–	–	–	Not used.
A	–	–	–	x	Twelve/twenty-four hour mode. If bit D0 is set, the clock is in the twenty-four hour mode. If bit D0 is clear, the alarm is in the twelve hour mode.

Table 14-7 (continued). Real-Time Clock Registers in Mode 1

REG. NO.	BIT 3	2	1	0	DESCRIPTION
B	–	–	x	x	Leap year counter. This counter follows the ones of years counter and when bits D1 and D0 are both zero, the clock logic will generate a 9 in the tens of days counter following an 8, when the ones of days and the ones of months are both set to 2, indicating February 29 is to follow February 28. Note that because there is no century counter, the clock cannot determine when the century year is not to be a leap year.
C	–	–	–	–	Not used.
D	x	x	x	x	Clock enable, alarm enable, and mode selection register. If bit D3 is set, the clock is allowed to run; if clear, the clock will not run. If bit D2 is set, the alarm is enabled; if clear, the alarm is disabled. If bits D1 and D0 are clear, mode 0 will be selected; if bit D1 is clear and bit D0 is set, mode 1 will be selected; if bit D1 is set and bit D0 is clear, mode 2 will be selected; and if bits D1 and D0 are both set, mode 3 will be selected. For normal operation in this computer, the clock is enabled and the alarm is disabled.
E	x	x	x	x	Test register. This register is used during testing by the chip manufacturer for high-speed function testing. If this register is not clear, the clock will not operate properly.
F	x	x	x	x	Reset controller. Bit D3 must be clear to enable the 1 Hz signal out the $\overline{\text{ALARM}}$ pin. Bit D2 must be clear to enable a 16 Hz signal. Bit D1, when set will reset all the clock's internal counters (see Table 14-6). Bit D0, when set will reset the alarm's internal counters. For normal operation in this computer, on bit D2 of this register is set.

Table 14-8. Real-Time Clock Registers in Mode 2

REG. NO.	BIT 3	2	1	0	DESCRIPTION
0	x	x	x	x	Four bits of non-volatile RAM.
1	x	x	x	x	Four bits of non-volatile RAM.
2	x	x	x	x	Four bits of non-volatile RAM.
3	x	x	x	x	Four bits of non-volatile RAM.
4	x	x	x	x	Four bits of non-volatile RAM.
5	x	x	x	x	Four bits of non-volatile RAM.
6	x	x	x	x	Four bits of non-volatile RAM.
7	x	x	x	x	Four bits of non-volatile RAM.
8	x	x	x	x	Four bits of non-volatile RAM.
9	x	x	x	x	Four bits of non-volatile RAM.
A	x	x	x	x	Four bits of non-volatile RAM.
B	x	x	x	x	Four bits of non-volatile RAM.
C	x	x	x	x	Four bits of non-volatile RAM.
D	x	x	x	x	Clock enable, alarm enable, and mode selection register. If bit D3 is set, the clock is allowed to run; if clear, the clock will not run. If bit D2 is set, the alarm is enabled; if clear, the alarm is disabled. If bits D1 and D0 are clear, mode 0 will be selected; if bit D1 is clear and bit D0 is set, mode 1 will be selected; if bit D1 is set and bit D0 is clear, mode 2 will be selected; and if bits D1 and D0 are both set, mode 3 will be selected. For normal operation in this computer, the clock is enabled and the alarm is disabled.
E	x	x	x	x	Test register. This register is used during testing by the chip manufacturer for high-speed function testing. If this register is not clear, the clock will not operate properly.
F	x	x	x	x	Reset controller. Bit D3 must be clear to enable the 1 Hz signal out the $\overline{\text{ALARM}}$ pin. Bit D2 must be clear to enable a 16 Hz signal. Bit D1, when set will reset all the clock's internal counters (see Table 14-6). Bit D0, when set will reset the alarm's internal counters (see Table 14-7). For normal operation in this computer, on bit D2 of this register is set.

Table 14-9. Real-Time Clock Registers in Mode 3

REG.		BIT			
NO.	3	2	1	0	DESCRIPTION
0	x	x	x	x	Four bits of non-volatile RAM.
1	x	x	x	x	Four bits of non-volatile RAM.
2	x	x	x	x	Four bits of non-volatile RAM.
3	x	x	x	x	Four bits of non-volatile RAM.
4	x	x	x	x	Four bits of non-volatile RAM.
5	x	x	x	x	Four bits of non-volatile RAM.
6	x	x	x	x	Four bits of non-volatile RAM.
7	x	x	x	x	Four bits of non-volatile RAM.
8	x	x	x	x	Four bits of non-volatile RAM.
9	x	x	x	x	Four bits of non-volatile RAM.
A	x	x	x	x	Four bits of non-volatile RAM.
B	x	x	x	x	Four bits of non-volatile RAM.
C	x	x	x	x	Four bits of non-volatile RAM.
D	x	x	x	x	Clock enable, alarm enable, and mode selection register. If bit D3 is set, the clock is allowed to run; if clear, the clock will not run. If bit D2 is set, the alarm is enabled; if clear, the alarm is disabled. If bits D1 and D0 are clear, mode 0 will be selected; if bit D1 is clear and bit D0 is set, mode 1 will be selected; if bit D1 is set and bit D0 is clear, mode 2 will be selected; and if bits D1 and D0 are both set, mode 3 will be selected. For normal operation in this computer, the clock is enabled and the alarm is disabled.
E	x	x	x	x	Test register. This register is used during testing by the chip manufacturer for high-speed function testing. If this register is not clear, the clock will not operate properly.
F	x	x	x	x	Reset controller. Bit D3 must be clear to enable the 1 Hz signal out the $\overline{\text{ALARM}}$ pin. Bit D2 must be clear to enable a 16 Hz signal. Bit D1, when set will reset all the clock's internal counters (see Table 14-6). Bit D0, when set will reset the alarm's internal counters (see Table 14-7). For normal operation in this computer, on bit D2 of this register is set.

The 82C59 Interrupt Controller

The interrupt controller monitors various devices in the system and notifies the CPU if any of them require immediate attention. These include the timer, keyboard, floppy disk controller, and the input/output ports.

There are two types of interrupts in this computer: maskable and non-maskable interrupts (NMI). Maskable interrupts are handled by the interrupt controller while the non-maskable interrupt goes directly to the CPU and is reserved for serious error conditions and those circuits that have critical (tight) timing requirements.

The 82C59 is designed to be used in a interrupt driven environment, such as the one used in this computer. It manages up eight request lines and is programmed as an input/output device. The user may select from several priority modes which can be used to configure the 82C59 to match the requirements of the system. Furthermore, this configuration may be changed "on the fly", while the system is running. That allows the system to be configured as various requirements for interrupts arise during operation of the system.

Individual interrupt lines can be masked without affecting those with either higher or lower priority. Of the eight maskable interrupts, three are used by this computer: the system timer (IRQ0 – interrupt request 0) and the keyboard (IRQ1 – interrupt request 1), and the floppy disk controller (IRQ6 – interrupt request 6).

The system timer interrupt, which has the highest priority of the maskable interrupts, is a clock signal generated by the 82C53 and is used by the computer for disk drive timing and to generate certain clock "ticks" for the operating system. The keyboard interrupt has the next highest priority and is generated whenever a key is pressed on the keyboard. The floppy disk interrupt is from the floppy disk controller board, which generates the interrupt at the end of a data transfer cycle.

Internal Structure of the Controller

Refer to the block diagram in Figure 14-7 while reading the following material. There are eight major blocks in the chip, seven of which are used in the computer: the data buffer, the control logic, the read/write logic, the in-service register, the priority resolver, the interrupt request register, the cascade/buffer comparator (not used in this computer), and the interrupt mask register. Lines and arrows determine the interrelationship between these blocks.

The data buffer is an 8-line, bidirectional, three-state buffer that interfaces the address/data bus to the CPU. Instructions, status information, and interrupt vector data are transferred through this buffer.

The control logic is responsible for sending the interrupt signal to the CPU and receiving the interrupt acknowledge back from the CPU. It communicates with the priority resolver and receives information from the in-service register. During initialization and programming, it receives instructions from the read/write logic.

The read/write logic is responsible for handling the programming of the chip, both during initialization and during operation. It is also responsible for releasing status information onto the data bus when requested by the CPU. The controlling lines, which are treated by the CPU as an input/output port, are the read, write, chip select, and address line 0 lines.

The in-service register is one of three internal registers in the IC. This register is used to store all the interrupt requests that have been acknowledged by the CPU and are currently being serviced. The priority resolver looks at this data to see if an incoming interrupt request is already being serviced by the CPU.

The priority resolver determines the priorities of the interrupt requests in the interrupt request register. It does this by first determining if the interrupt request line has been masked by the interrupt mask register and then if the interrupt is in the in-service register, which tells the resolver that the interrupt is already being serviced.

The interrupt request register receives all the incoming interrupt request lines and latches them until they can be processed by the CPU. This register stores all interrupts which are requesting service. Once the interrupt request has been ac-

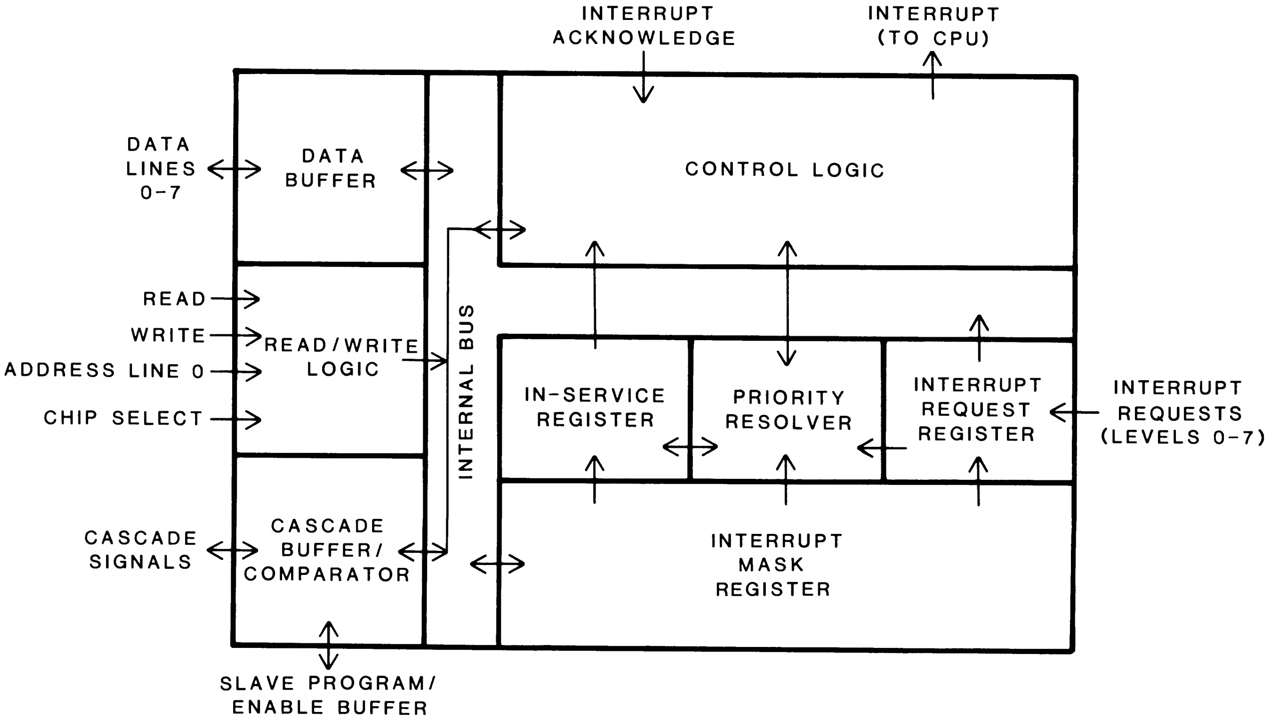

Figure 14-7. Interrupt Controller Block Diagram

knowledged by the CPU, it is transferred to the in-service register. However, before the interrupt request register will recognize another interrupt request on the same line, that line must go inactive. If the request has been transferred to the in-service register, the interrupt request register will clear (which will also clear the in-service register) and will recognize a subsequent interrupt request that is received on the same line. This action prevents an interrupt request that has been serviced by the CPU from reactivating the service routine by going through the system a second or subsequent time.

The cascade/buffer comparator is not used in this computer. It is used only in those installations where more than one 8259 is used for a wider interrupt level structure. The PC-compatible design does not require or recognize more than eight interrupt levels.

The interrupt mask register is used to disable selected interrupt request lines. It operates in conjunction with the priority resolver to mask out incoming interrupt requests. Note that masking one priority level will not affect the operation of higher or lower priority interrupts.

Initialization

As part of initializing the computer system, the CPU must initialize certain internal registers and this programs the interrupt controller by sending it a series of commands. Here is the procedure.

The system will first initialize the SS (stack segment) register of the CPU. This register points to the start of the stack, which is located near the top of scratchpad memory between addresses F0000H and F3FFFH. The exact address depends on the version of the MFM monitor program installed in your computer. Note that this stack location is only used by the MFM monitor program. Under DOS operation, the stack segment register may point elsewhere.

Next, the interrupt vector table in low memory will be initialized. This table contains the addresses of the service routines used by each interrupt. Each interrupt vector address is four bytes long: two bytes are used for the base address for the CS (code segment) register, while the other two are used for the offset address for the IP (instruction pointer).

Finally, the CPU will initialize the interrupt controller with a series of initialization commands. These commands set up the controller for normal operation. Table 14-10 through Table 14-13 describes the contents of the four command words that may be sent to the controller. Note that three or four commands may be sent in sequence. The A0 line, when low, identifies the first command word. Then the A0 line is made high and the following command words are sent to the controller in sequence.

Table 14-10. Initialization Command Word 1

BIT	DESCRIPTION
0	This bit informs the controller whether there will be three or four command words in the sequence. If the bit is high, there will be four command words in the sequence. If the bit is low, there will be three command words in the sequence.
1	This bit tells the controller whether it is the only controller in the system or not. If this bit is set to one, only one controller is in the system and no command word 3 will be issued.
2	This bit tells the controller whether the vector addresses are four bytes long or eight bytes long. The computer uses four byte vector addresses, so this bit will be set.
3	This bit tells the controller to recognize interrupt requests on the leading edge of the interrupt line transition or when the level is high. If this bit is set, the interrupt request will be recognized on the edge of the transition.
4	Always set.
5 – 7	These three bits are not used in this computer. If the 8259 is used in a computer with an 8-bit processor, such as the 8080 or 8085, then these three bits are programmed with address bits A5 – A7 of the vector table address. If the vector addresses are four bytes long, the interrupt controller will automatically supply address bits A0 – A4, depending upon the interrupt requested. If the vector addresses are eight bytes long, the interrupt controller will automatically supply address bits A0 – A5 and bit 5 in this command word will be ignored.

Table 14-11. Initialization Command Word 2

BIT	DESCRIPTION
0 – 2	Not used in this computer. See the explanation for the bit 0 – 7 configuration.
3 – 7	These five bits are programmed with address bits A3 – A7 in this computer. In computers with 8-bit CPUs, such as the 8080 or 8085, these bits are programmed with address bits A11 through A15. See the explanation for the bit 0 – 7 configuration.
0 – 7	This configuration is not used in this computer. In computers with an 8-bit CPU, such as the 8080 or 8085, these bits are programmed with address bits A8 – A15.

Table 14-12. Initialization Command Word 3

BIT	DESCRIPTION
0 – 7	If this interrupt controller is the master device, these bits let the controller know which interrupt request line has a slave interrupt controller attached to it. Each bit corresponds to an interrupt request line.
0 – 7	If this interrupt controller is a slave device, thee three low-order bits are programmed in binary with the value that corresponds to the interrupt line of the master interrupt controller to which this controller is attached. For instance, if this controller is programmed with a six (bits 1 and 2 are set), then this controller would be connected to interrupt request line 6 of the master interrupt controller and bit six in command word 3 for the master interrupt controller would be set.

Table 14-13. Initialization Command Word 4

BIT	DESCRIPTION
0	This bit tells the interrupt controller whether it is in the 8088/8086 (16-bit CPU) mode or the 8080/8085 (8-bit CPU) mode. If this bit is set, the controller is in the 8086/8088 mode.
1	This bit tells the controller whether to program for automatic end-of-interrupt mode or not. If this bit is set, then the controller will be in the automatic end-of-interrupt mode. In the automatic end-of-interrupt mode and in 16-bit mode the interrupt controller will perform a non-specific end-of-interrupt at the trailing edge of the second interrupt acknowledge received from the CPU. In 8-bit mode, this operation is performed at the end of the third interrupt acknowledge received from the CPU.
2	If bit 3 of this word is set, then this bit determines whether the interrupt controller is a master or slave controller. If this bit is set, the controller is a master controller. If bit 3 of this word is not set, then this bit will be ignored.
3	This bit determines whether the controller is in the buffered mode or not and controls the output of the $\overline{\text{SP}}/\overline{\text{EN}}$ line and the status of bit 2 of this word. If set, buffered mode is selected and the $\overline{\text{SP}}/\overline{\text{EN}}$ line becomes an enable output line; bit 2 of this command word will be checked for the master/slave option.
4	This bit determines whether the interrupt controller works in the fully nested mode where the controller supports more than one level of interrupt and the priority order of the eight lines is arranged from highest (IRQ0) to lowest (IRQ7). If the bit is not set, then the controller will not operate in the fully nested mode of operation, then one interrupt will not interrupt the operation of another, but will operate in a sequential manner. Note that the priority order may be changed by software.
5 – 7	Not used.

In this computer, the interrupt controller will be set as follows.

- The interrupt request inputs, IRQ0 – IRQ7, are set to be edge-sensitive. The controller will send an interrupt to the CPU on the positive-going transition of one of these lines.
- Interrupt priority is in reverse numerical order. IRQ0 has the highest priority, IRQ7 the lowest.
- All interrupt masks are cleared.
- If the interrupt controller sends an interrupt to the CPU, the controller will not send any additional interrupts to the CPU until the CPU executes a specific end-of-interrupt instruction.
- The interrupt controller uses vector addresses that are spaced every four bytes in a 32-byte table in memory.

Programming

Once programmed, the interrupt controller is ready to accept interrupt requests through its input lines. However, three commands may be sent to it that cause the controller to operate in different modes. These three commands, which, unlike the initialization command words, do not have to be in sequential order, but may be sent to the controller as needed.

Command word 1 is used to place a mask into the controller's interrupt mask register. Each of the data bits sent to the controller during the write operation correspond to one of the interrupt request lines: bit 0 corresponds to interrupt request line 0, bit 1 to line 1, and so on. If a bit is set, the line is masked. If a bit is not set, the line is not masked. Address line 0 must be set prior to executing this write operation to the controller.

Command word 2 is used for end of interrupt and rotation commands. It is summarized in Table 14-14. The address line 0 must be clear for the write operation of this command word.

Table 14-14. **Operation Command Word 2**

BIT	DESCRIPTION
0 – 2	These three bits, coded in a binary format, determine the interrupt level (request line) that is to be acted upon by the remainder of this word. If all three bits are clear, interrupt request 0 is programmed; bit 0 set and bits 1 and 2 clear will program interrupt request 1; and so on. Note that bit 6 of the command word can disable the function of these three bits.
3 – 4	These two bits are clear, signifying that is command word 2.
5	This bit determines if an end-of-interrupt command will be issued by the controller. If the bit is not set, no end-of-interrupt command will be executed. If the bit is set, bit 6 and bit 7 determine the type of end-of-command operation that is executed.
6	This bit enables or disables the use of bits 0 – 2. If this bit is set, the first three bits of the command word are enabled. If this bit is clear, they are disabled.
7	This bit controls all interrupt controller rotation operations. If this bit is set, rotation will be performed, depending upon the state of bits 6 and 7 of this command word. If bit 5 and bit 6 are both set, then the interrupt request line specified by bits 0 – 2 will be set to the lowest priority and the remainder of the interrupt request levels will conform to the nested mode based upon this interrupt. If bit 5 is set and bit 6 is clear, then bits 0 – 2 will be ignored. If bit 5 and bit 6 are both clear, then no end-of-interrupt command will be issued by the controller and bits 0 – 2 are ignored. If bit 7 is clear, rotation will not be performed.

Command Word 3 is used for a number of purposes. The functions performed by command word 3 is summarized in Table 14-15.

Table 14-15. **Operation Command Word 3**

BIT	DESCRIPTION
0	This bit determines which register (the in-service register or the interrupt request register) is to be read by a read register status operation. If this bit is set, the in-service register will be read. If this bit is clear, the interrupt request register will be read.
1	This bit determines whether a read register command has been issued or not. If this bit is set, then a read register command has been issued. Unless bit 2 is set, bit 0 will determine which register is to be read.
2	This bit determines whether a poll command has been issued or not. If this bit is set, it overrides bit 1 and and poll command has been issued (bit 1 will be ignored). If this bit is clear, a poll command has not been issued.
3 – 4	Bit 3 is set and bit 4 is clear, signifying that is command word 3.
5	This bit determines whether the special mask mode is to be enabled or disabled. Bit 6 controls the validity of this bit.
6	If this bit is set, then bit 5 will be honored and the special mask mode will be enabled or disabled, depending upon the status of bit 5. If this bit is clear, bit 5 will be ignored.
7	Not used.

Operation

Operation of the interrupt controller falls into five areas: CPU mode, priority handling, interrupt trigger, status, and cascading.

CPU Mode — The controller is designed to operate with one of two types of CPUs: eight-bit devices, such as the 8080 and 8085 CPUs; and sixteen-bit devices, such as the 8088 and 8086 designs. The CPU mode determines which type of device the controller operates with and is programmed during the initialization command word sequence. In this computer, the interrupt controller is always operated in the mode for sixteen-bit CPUs.

When an interrupt takes place in the sixteen-bit mode, the controller will place a single interrupt-vector address byte on the data bus in response to two interrupt acknowledge signals from the CPU.

The first signal is used by the controller to resolve interrupt priorities. The second signal times the placing of the interrupt address byte on the data bus. Bits 0 – 2 are determined by the interrupt request being serviced and bits 3 through 7 are determined by the pattern programmed into the controller during initialization command word 2.

The eight bits placed on the data bus are not used as a direct address, but rather, as the call number for the INT command, which is executed by the CPU. These INT commands are the same commands that are used in software.

When the CPU receives the eight bits from the interrupt controller, the processor multiples the value by four to obtain the address of the interrupt vector address. Program execution is then transferred to the address of the routine that is pointed to by the values in the interrupt vector address location.

Priority handling — Priorities fall into two categories, fully nested and masked, and may be handled several different ways by the controller. These modes are dynamically programmable; that is, the way priorities are handled may be changed during the operation of a program.

The fully nested mode is a general purpose mode that supports multiple levels of interrupt priority. The eight interrupt request lines are handled in a highest-to-lowest manner, usually with interrupt request line 0 as the highest priority. This is the default mode that is set up during the initialization of the controller.

Programming can change which interrupt request line is the highest priority. For instance, you could make interrupt request line 4 the highest priority. In that case, interrupt request line 5 would be the next highest, interrupt request line 6 the next, and so on until you reached interrupt request line 3, which would be the lowest.

Once an interrupt is acknowledged by the CPU, the highest priority request is determined from the interrupt request register. The vector value is then placed on the data bus and the corresponding bit in the in-service register is set. The in-service register bit will remain set until an end-of-interrupt command is executed and sent to the interrupt controller.

In the fully nested mode, once the in-service register bit has been set, all subsequent requests by the same or lower-priority interrupt request line will not generate an interrupt to the CPU. However, a higher-priority interrupt request will be honored and will interrupt the execution of the service routine for the lower-priority interrupt. Since the interrupt request pin on the CPU is disabled when the interrupt acknowledge is sent back to the interrupt controller, the CPU must have an enable interrupt instruction executed before a higher-priority interrupt request can be acknowledged. Consider the following example.

While the main program is being executed, the in-service register will be clear, since no interrupts are being serviced.

Suppose interrupt request line 3 becomes asserted. The interrupt controller notifies the CPU of this and is sent and interrupt acknowledge signal. The controller places the vector byte on the data bus and receives the second interrupt acknowledge signal. At that point bit three of the in-service register is set and the CPU starts executing the service routine for interrupt line 3. One of the first instructions to be executed in this routine is the enable interrupts instruction, which allows the CPU to receive further interrupt requests from the controller.

Now suppose that while the CPU is executing this service routine, that interrupt line 1 is asserted. Just as it happened with interrupt request 3, the interrupt controller notifies the CPU of this and is sent and interrupt acknowledge signal. The controller places the vector byte on the data bus and receives the second interrupt acknowledge signal. At that point bit one of the in-service register is set and the CPU starts executing the service routine for interrupt line 1. At this point bit 1 and bit 3 are both set in the in-service register because the service routine for interrupt line 3 has not been completed.

Also, at this point, and only after the CPU once again receives an enable interrupt command, the only interrupt that the controller will act on will be one coming in from the interrupt request line 0.

When the service routine for interrupt line 1 has finished, it must inform the controller by executing an end-of-interrupt command. This will clear bit 1 in the in-service register. Next a return instruction must be executed. This will return control to the service routine for interrupt line 3, which will allow any interrupt line from 0 to 2 to be serviced by the system.

When the service routine for interrupt line 3 has finished, the end-of-interrupt command will reset bit 3 in the in-service register and the return will transfer control back to the main program.

The interrupt controller in this computer is almost always in the fully nested mode of operation. Only two programming conditions can disturb this mode: the automatic end-of-interrupt mode and the special mask mode.

Three different end-of-interrupt formats may be programmed: the non-specific end-of-interrupt command, the specific end-of-interrupt command, and the automatic end-of-interrupt command.

The non-specific end-of-interrupt command, while letting the controller know that an interrupt service routine has been completed, does not inform the controller which level of interrupt is involved. By being in a mode where the controller can determine service routine levels, it can determine that the interrupt level that applies to the routine that just completed corresponds to the highest interrupt level bit that is set in the in-service register. The non-specific end-of-interrupt command will reset this bit.

There are two conditions that may cause the non-specific end-of-interrupt routine to fail: when the service routine reset interrupt priorities and when the special mask mode is in use. In both cases, the controller may not be able to determine the routine's interrupt level.

The specific end-of-interrupt command must include the in-service bit to be reset. This allows the programmer the latitude to change interrupt priorities with the servicing routine or perform other functions that might make it vague to the controller which interrupt routine was being serviced, particularly if other service routines were being executed at the same time.

The automatic end-of-interrupt mode eliminates the need for the CPU to issue an end-of-interrupt command to notify the controller that an interrupt service routine has been completed. While in this mode, the controller will perform a non-specific end-of-interrupt at the trailing edge of the second interrupt acknowledge signal from the CPU.

This mode disturbs the fully nested mode because the in-service register bit is reset right after it was acknowledged, leaving no sign that the service routine is being executed. Therefore, any interrupt request (when interrupts are enabled) will get serviced, regardless of its priority, making it possible for an interrupt request to interrupt its own service routine.

It is considered good programming practice not to use the automatic end-of-interrupt mode unless the CPU's interrupt input will be kept disabled while interrupt routines are being serviced.

Rotation of the interrupt priorities is available under several conditions and is provided in two modes: automatic rotation and specific rotation.

Automatic rotation is desirable when the interrupts are equal in nature, for instance, with a series of communication channels handling peripheral devices. The concept is that once a peripheral is serviced, all other equal priority peripherals should be given the opportunity of being serviced before the first is again serviced. To accomplish this, the automatic rotation mode assigns the just-serviced line the lowest priority. Automatic rotation may be implemented with the rotate on non-specific end-of-interrupt command or in the rotate in automatic end-of-interrupt mode.

When the rotate on non-specific end-of-interrupt command is given, the in-service register bit being serviced is reset and its corresponding interrupt request line is assigned lowest priority. The other lines' priorities are then rotated to conform to the fully nested format, based on the bit that has been assigned the lowest priority. For instance, if bit 3 was just serviced, it is assigned lowest priority, bit 4 the highest, bit 5 the second highest, and so on around to bit 2 which is second lowest.

The automatic end-of-interrupt mode works much the same way, except that the reassignment of priorities takes place at the falling edge of the second interrupt acknowledge received from the CPU.

Specific rotation is completely controllable by the programmer. Through this operation, the specific interrupt request line is selected to receive the highest or lowest priority. Two commands allow this: the set priority command and the rotate on specific end-of-interrupt command.

The set priority command is used to assign an interrupt request line to the lowest priority. The other lines' priorities are then rotated to conform to the fully nested format, based on the bit that has been assigned the lowest priority.

If the set priority command is used during a service routine then you must use either a specific end-of-interrupt command or the automatic end-of-interrupt mode to end the routine. The non-specific end-of-interrupt resets the highest in-service register bit, which may not represent the service routine that issued the set priority command. If the automatic mode is used, there is no problem because it performs the non-specific end-of-interrupt before the set priority command can be issued. It is still the best practice to use the specific end-of-interrupt command to eliminate any possible confusion.

The rotate on specific end-of-interrupt command is a combination of the set priority command and the specific end-of-interrupt command. With this command you specify which interrupt request line is assigned to the lowest priority and issue the end-of-interrupt command at the same time. The other lines' priorities are rotated to conform to the fully nested format, based on the bit that has been assigned the lowest priority.

Masking interrupts allows the programmer to enable interrupt request lines that are at a lower priority than the one being serviced.

As an example, suppose interrupt request line 4 has triggered its interrupt service routine, but you want to allow lower-level interrupt request lines to interrupt the service routine. Inside the service routine for interrupt request line 4, you would first mask interrupt request line 4 and then issue a special mask mode command. This disables normal nested mode priority operation and enables all interrupt request lines except those being masked. To leave this mode, the sequence is executed in reverse order.

There is one problem, however, using the mode. You cannot use a non-specific end-of-interrupt command because all masked interrupt request line bits are hidden and are not clearable from the in-service register. Only if the special mask mode has been exited can you use the non-specific end-of-interrupt command to clear masked bits from the in-service register. Therefore, it is the best policy to issue a specific end-of-interrupt command when using this mode.

Interrupt trigger — Two traditional means are used to sense an interrupt: level-sensitive and edge-sensitive.

In the level-sensitive interrupt mode, the interrupt controller will recognize an active high on any of its interrupt request lines. If the interrupt request line remains active after the end-of-interrupt command is issued, another interrupt will be generated if the CPU has been told to recognize interrupts. In this mode the interrupt must remain active until the first interrupt acknowledge from the CPU has been received. Otherwise, the controller will act as if interrupt request line 7 had been active.

The edge-sensitive mode is used as the default in this computer. In this mode, the interrupt controller recognizes the interrupt on the rising edge as a interrupt request line goes active. If the interrupt request line remains active after the service routine has been completed and the processor set to recognize interrupts, it will not trigger a subsequent interrupt. In this mode the interrupt must also remain active until the first interrupt acknowledge from the CPU has been received. Otherwise, the controller will act as if interrupt request line 7 had been active.

Status — There may be occasions where the status of the three internal registers needs to be known, particularly by an interrupt service routine. Polling of interrupts is also possible, though it is not needed in this computer.

The three internal registers of the interrupt controller, the in-service register, the interrupt request register, and the interrupt mask register, can be read by software. The interrupt request register specifies all interrupt request lines that are currently requesting service. The in-service register specifies all interrupt request lines that are currently being serviced by routines. The interrupt mask register specifies all interrupt request lines that are currently masked.

Cascading — Interrupt cascading is not possible in this computer because there is only one interrupt controller installed. Although this is a feature of the controller, it is not discussed here because it is not applicable in this installation.

Interrupt Controller Port Address

The interrupt controller is treated as an input/output device for programming purposes. Its addresses are 020H (A0 = 0) and 021H (A0 = 1). In those instances where it has been indicated that A0 needs to be cleared, you would use OUT or IN commands to port address 020H. If A0 needs to be set, you would use OUT or IN commands to port address 021H.

Sequence of Operation

Once the interrupt controller has been programmed, the CPU must be instructed to set a bit in its flag register to enable interrupts. If one or more circuits should generate an interrupt, the system will respond as follows.

- One or more of the interrupt request lines, which are labeled IRQ0 to IRQ7 for interrupt request, will go high.
- The controller will check the priorities of the incoming interrupts and compare them to others that may also be waiting to be serviced.
- The controller will assert the the INTRQ line to the CPU.
- CPU recognizes the interrupt request, completes its current instruction, and places the interrupt-acknowledge code on the status bus.
- The interrupt controller will resolve the priority of the interrupts that are requesting service and prepare the vector address value of the selected interrupt line for placement on the data bus during the next interrupt acknowledge cycle.
- The CPU next generates the second interrupt acknowledge signal.
- Upon receipt of the second interrupt acknowledge signal from the CPU, The controller will place the 8-bit interrupt vector address value on the address/data bus.
- The CPU will then multiply the 8-bit value by four to determine the actual address in the interrupt pointer (vector) table for the interrupt service routine.
- The CPU also disables interrupts by clearing the flag bit in its flag register. Then it pushes the flag register, IP (instruction pointer), and CS (code segment register) values onto the CPU stack.
- The new IP and CS values are fetched from the interrupt pointer table and control is jumped to the routine at that address.
- The interrupt-handling subroutine should push any other CPU registers that are to be used during the service routine onto the stack.
- Typically, at the end of the service routine, an end-of-interrupt command is sent to the controller. This resets the in-service register bit for the service routine, indicating that the routine has finished.

- Before the routine is exited, the appropriate CPU registers are popped off the stack, interrupts are enabled, and the CPU returns control to the instruction following the one it completed when the service routine was called.

There will be variations in this sequence depending upon a number of factors, the least of which is the possibility that alternate modes can be used by the interrupt controller under program control. However, this should provide you an idea of how a typical operation would take place.

The 82C53 Programmable Interval Timer

The programmable interval timer is used to generate time delays of varying lengths, which eliminates the need for software timing loops which can be subject to such variables as CPU clock speed and interrupt service routines.

Instead, the programmer can configure the timer to match his requirements and initialize one or more of the three counters in the timer with the desired values. Then, upon command, the timer will count to the value and issue an interrupt request upon completion of the task.

The timer can also be used as a programmable rate generator, and event counter, a binary rate multiplier, a real-time clock, a digital "one-shot", or a complex motor controller. Not all of these functions are used in this computer.

The timer can operate in one of six modes: interrupt on terminal count, programmable one-shot, rate generator, square-wave rate generator, software triggered strobe, and hardware triggered strobe.

Refer to Figure 14-8, the block diagram of the timer, for the following discussion.

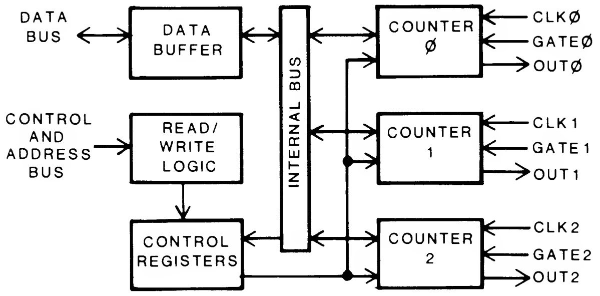

Figure 14-8. Programmable Interval Timer Block Diagram

The timer is divided into six logical sections: the data buffer, the read/write logic, the control register, and the three counters.

The data buffer is a three-state bidirectional buffer used to interface the timer with the address/data bus. Data is set to or received from the bus via IN or OUT commands to ports 040H – 043H.

The read/write logic accepts the $\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{CS53}}$, A0, and A1 signal lines as inputs to determine the type of operation to be performed inside the timer. $\overline{\text{CS53}}$ controls chip access while address lines 0 and 1 determine the type of read or write operation requested.

The control registers receive the instructions that program the counters. The control word for each counter is selected by the value in bits 5 and 6. Table 14-16 describes the contents of the registers. Since all three registers (one for each counter) are identical, only one register is described.

Table 14-16. Timer Control Registers

BIT	DESCRIPTION
0	This bit establishes the counter format: 16-bit binary or 4-digit binary-coded decimal. If the bit is set, the counter will be in the 4-digit binary-coded decimal format. If the bit is clear, the counter will be placed in the 16-bit binary format.
1 – 3	These three bits establish the mode for the selected counter. If they equal zero, mode 0 is selected; a 1 selects mode 1, a 2 or a 6 selects mode 2, a 3 or a 7 selects mode 3, a 4 selects mode 4, and a 5 selects mode 5. Refer to the text for a description of each mode.
4 – 5	These two bits control the loading (writing) of the count into the specified counter. If the value placed in these bits is 0, the counters are latched; if the value is 1, only the most significant byte will be read or loaded; if the value is 2, only the least significant byte will be read or loaded; if the value is 3, the least significant byte will be read or loaded first, followed by the most significant byte. In all cases the number of bytes (1 or 2) must be read or loaded before a different operation can be applied to the specified counter.
6 – 7	These two bits specify the counter to receive the operation. If the value is 0, counter 0 is specified; a value of 1 specifies counter 1; and a value of 2 specifies counter 2. If the value is three, no counter is selected; the value is illegal.

Each of the three counters contain two bytes and may be configured as either a 16-bit binary counter or a 4-digit binary-coded decimal counter. The input, gate, and output lines are configured by the modes programmed through the control register during programming. In addition, each counter may be selectively read without first inhibiting the clock input for the counter being read.

Mode Definitions

Many of the modes require that specific hardware conditions do not exist. For instance, the gates, clocks, and outputs must be free, not tied to specific circuits, which is not the case in this computer. While the programming is fully explained in this section, many of the functions are not available, since the timer is more or less dedicated to specific tasks, as proscribed by PC compatibility. For instance, gates 0 and 1 are tied high in this computer, so they cannot be used to control counters 0 and 1. Gate 2 is tied to the PB0 signal, which is generated by the gate array whenever a key is pressed. Counter 2's output is sent back to the gate array where it is processed and eventually feeds the speaker to produce the audible key click heard whenever a key is pressed. The other two counters are equally committed to specific tasks. Counter 0 is used to provide the CPU with memory refresh timing. Counter 1 is used with DMA requests. So in reality, none of the counters can be fully controlled to produce special timing signals beyond that which is allowed within the confines of the computer's design. Refer to Table 14-17 that indicates the function of each mode and to the descriptions that follow.

Table 14-17. Programmable Interval Timer Modes

MODE	FUNCTION
0	Interrupt on terminal count.
1	Programmable one-shot.
2	Rate generator.
3	Square wave rate generator
4	Software triggered strobe.
5	Hardware triggered strobe.

Mode 0 — This straight-forward mode will decrement the counter until it reaches 0 and then assert its output line high. Refer to Figure 14-9.

The upper set of lines illustrate the counter without receiving any low signals on the gate input line. Once the counter has been loaded, in this case, with a value of 4, the counting (decrementing) of the count starts. When the count reaches 0 the output will go high. Counting will not resume until the counter is reloaded.

The lower set of lines illustrate the counter being affected by the gate line. In this case the counter is loaded with a value of 5. As long as the gate line remains high, the counter will decrement. When the gate line goes low, decrementing the counter will halt, but not reset to the original value. When the gate line goes high again, decrementing the count will resume. When the count reaches 0 the output will go high. Counting will not resume until the counter is reloaded.

In this mode, the output of the counter should be tied to an interrupt line of the CPU. The gate signal will disable counting when low and enable counting when high.

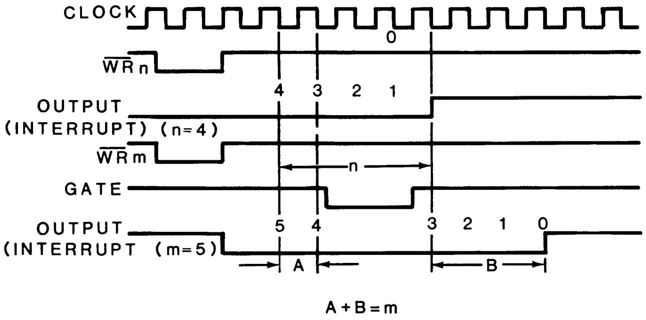

Figure 14-9. Timing Mode 0

Mode 1 — In this mode, the counter acts like a programmable one-shot. The output will go low the following clock cycle when the gate is asserted high (acting as a trigger). The output will remain low until the counter has decremented to 0. Refer to Figure 14-10.

However, if the trigger (the gate input line) goes low and then high again, the counter will be reset and will be held low until the counter decrements to zero.

The counter, when acting as a one-shot, is retriggerable. Therefore, the output will remain low until the counter has decremented from the original value to zero following the rising edge of the gate. The rising edge of the gate reloads and initiates counting.

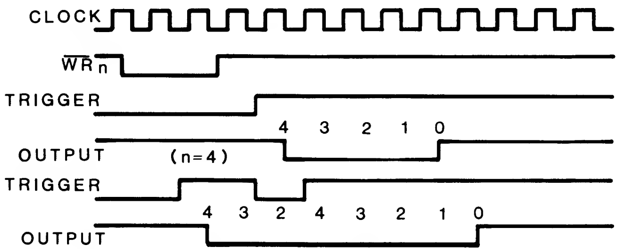

Figure 14-10. Timing Mode 1

Mode 2 — In this mode, the counter will produce a low for one clock cycle on its output pin every *n* clock cycles. As long as the gate remains high, the counter will repeatedly decrement the count to zero and start over. If the counter's value is changed, the new value will take effect the next counting cycle and not affect the current cycle. This action is illustrated in the upper set of lines. Refer to Figure 14-11.

The gate can be used to synchronize the counter. While the gate input is low, the counter will not decrement. When the gate goes high, the counter will reloaded with its programmed value and start decrementing. This action is illustrated in the lower set of lines.

When the gate is low, counting is disabled and the output is made high. The rising edge of the gate reloads and initiates counting. When the gate is high, counting is enabled.

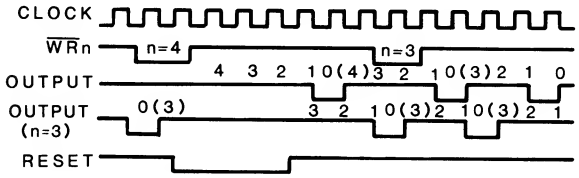

Figure 14-11. Timing Mode 2

Mode 3 — In this mode, the counter is decremented by two each clock cycle, rather than one, effectively halving the time it takes to reach zero. Refer to Figure 14-12.

If the programmed value is even, the counter is always decremented by two. When the counter reaches zero, the state of the output will change (for instance, from high to low), the counter is reloaded, and the decrementing by two starts over. When the counter reaches zero, the output state again changes (for instance, back to high), the counter is reloaded, and the process starts over again. This produces an even square wave.

If the value is odd and the output is high, the first clock cycle will decrement the counter by one and then by twos until it reaches zero. Then the clock will change state to low, the counter will be reloaded with the programmed value, and the first clock pulse will decrement it by three, then by twos until it reaches zero. This produces a signal where the output is high for $(n + 1)/2$ counts and low for $(n-1)/2$ counts.

When the gate is low, counting is disabled and the output is high. The rising edge of the gate initiates counting. When the gate is high, counting is enabled.

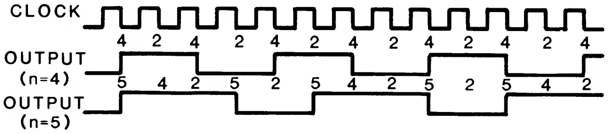

Figure 14-12. Timing Mode 3

Mode 4 — In this mode, the counter, when it reaches zero, will place a single pulse that lasts for one clock cycle on its output. Counting will begin when the counter is programmed with a value. Refer to Figure 14-13.

When the gate is low, counting is disabled. When the gate is high, counting is enabled.

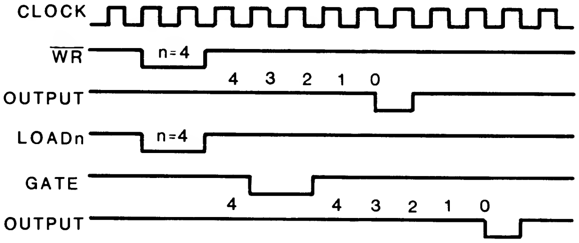

Figure 14-13. Timing Mode 4

Mode 5 — The counter, when it reaches zero in this mode, will place a single pulse that lasts for one clock cycle on its output. The counter will not start decrementing its value until it senses the rising edge of the gate input. The counter is retriggerable and will reload after it reaches zero. Refer to Figure 14-14.

The rising edge of the gate signal initiates counting.

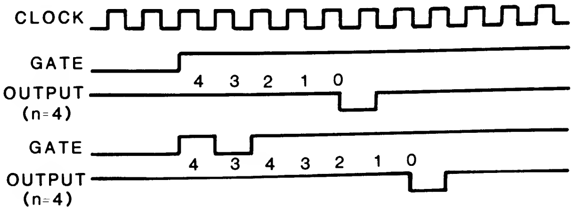

Figure 14-14. Timing Mode 5

Programming Considerations

The monitor ROM is used to program the mode and initial value in each of the three counters in the timer. In each case a control word is placed in the selected counter's register and then the programmed number of bytes (1 or 2) before the counter is started.

There is no special sequence in which the timer must be programmed. For instance, you can write a command word to any of the three counters without affecting the operation or programming of the other two. Suppose you want to load counter 0 with the least significant byte only, counter 1 with both bytes, and counter 2 with the most significant byte only. The following are two possible sequences that accomplish this task.

Load Sequence 1

1. Load command word 0, 1, and 2.
2. Load the most significant byte in counter 2.
3. Load the least significant byte in counter 0.
4. Load the least significant byte in counter 1.
5. Load the most significant byte in counter 1.

Load Sequence 2

1. Load command word 2.
2. Load the most significant byte in counter 2.
3. Load command word 0.
4. Load command word 1.
5. Load the least significant byte in counter 1.
6. Load the least significant byte in counter 0.
7. Load the most significant byte in counter 1.

NOTE: The only sequence that has to be followed is in loading two bytes into a counter. In the both examples, the least significant byte is followed by the most significant byte, before the counter starts operating. In the last example, the least significant byte and most significant byte for counter 1 are separated by the least significant byte for counter 0.

All counters decrement only. Therefore, if you load a counter with a value of zero, the actual count will equal the maximum value (10,000 in BCD; 65,536 in binary).

The values in the counters may be read. The address lines determine which counter is read. Normally you would read the contents of the counter with normal read operations. However, you may wish to read the value in the counter while it is operating. This presents problems because you may not get a valid result because the counter is continuing to decrement. The timer offers a method of latching the output of the counter so that it is stable while it is being read. The counter continues to decrement during the read operations but the data remains as it was when it was latched.

The timer contents may be read or values may be written through IN (read) and OUT (write) programming instructions. Table 14-18 describes each read and write operation that may be performed on the timer.

Table 14-18. Timer Read and Write Operations

PORT	OPERATION	DESCRIPTION
040H	OUT	Write (load) counter 0.
040H	IN	Read the contents of counter 0.
041H	OUT	Write (load) counter 1.
041H	IN	Read the contents of counter 1.
042H	OUT	Write (load) counter 2.
042H	IN	Read the contents of counter 2.
043H	OUT	If either bit 4 or bit 5 are set, write a control word to the specified counter's control register. Refer to Table 14-18 for a description of the contents of the control word. If bits 4 and 5 are both clear, latch the specified counter's value for reading. **NOTE:** The specified counter is identified in bit 6 and bit 7 of the control word.
043H	IN	Place the data buffer in the high-impedance state; do not perform any operation.

The CPU Gate Array

Most of the logic in the computer is contained in three devices: the CPU gate array, the decoder, and the monitor ROM. The decoder is described in the next section and the monitor ROM is described in Chapter 8.

The CPU gate array is a custom device that replaces many of the buffers, latches, and other devices normally required to support the CPU. It generates a number of signals, including many of those associated with the control bus. DMA control and clock circuits are now handled in this device.

As a signal generator, the CPU gate array provides a 4.77 MHz or 8 MHz (depending on the configuration switch setting), 33% duty cycle clock signal to the CPU. It also provides synchronization for the READY signal and provides the RESET signal at power-up. Additional timing signals (TCLK and CLK) are sent to the timer (previously described) and to the control bus to provide synchronized operation with other devices, by way of a buffer/driver.

The CPU gate array decodes the $\overline{\text{S0}}$, $\overline{\text{S1}}$, and $\overline{\text{S2}}$ status signals from the processor and coprocessor to enable the data and address buffers and latches. Read and write signals are generated for memory and peripheral control.

The CPU gate array handles the PC DMA functions, transferring blocks of data between memory and peripherals more quickly than similar CPU-controlled functions.

Except for the DMA functions, there are no special programming functions for the CPU gate array since it replaces components that do not have registers.

The CPU gate array package is a sixty-eight pin device. The socket contains two sets of pins on each of the four sides, making the device a bit difficult to use, look at, or measure signals.

Direct Memory Access

DMA (direct memory access) is used to speed the transfer of data from one memory location, device, or peripheral to another memory location, device, or peripheral. The primary advantage to DMA is in block moves of data. The DMA has its built-in instructions and can operate much faster than the CPU, which must decode a number of instructions in order to transfer data from one device to another. This computer uses a modified 8237 design in the CPU gate array as the DMA controller.

The DMA controller can operate in one of three modes. A single byte of data can be transferred, a block of data can be transferred with both the start and end of the block defined by the CPU, or a block of data can be transferred with the end of the block defined by an external signal on the EOP line.

The DMA controller operates in one of two types of cycles: the idle cycle and the active cycle. The controller is in an idle cycle when the CPU has control of the system bus. During this time, the CPU can program the controller or read the status registers. The active cycle occurs when the controller has control of the system bus. Some of the input lines become output lines so the controller can send address and control signals to the system bus.

When a trasfer of data is desired, the CPU tells the controller where to get the data and where to put it in memory. The controller then handles the data transfer. The controller generates up to 16 address lines through address outputs A0-A7. The high-order address is first placed on the address bus and latched by a high on the LD1 line. The controller only latches the high-order address when it changes, speeding the process. The address latches and buffers are enabled by a high on the AEN line. The CPU generates all of the memory and I/O read and write signals ($\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, and $\overline{\text{IOW}}$) from the S0, S1, and S2 lines through the inverter.

When the controller is to control the bus, a hold request (HOLD RQ) is issued to the CPU. The CPU responds with a hold acknowledge signal (HLDA) and goes into a hold state until the transfer is completed.

A peripheral can request DMA service from the controller on one of the DRQ lines The controller responds to the input by requesting a CPU hold, and then sending a DMA acknowledge output to the peripheral when the CPU enters a hold state.

Refer to Figure 14-15 for the DMA controller bus timing illustrations.

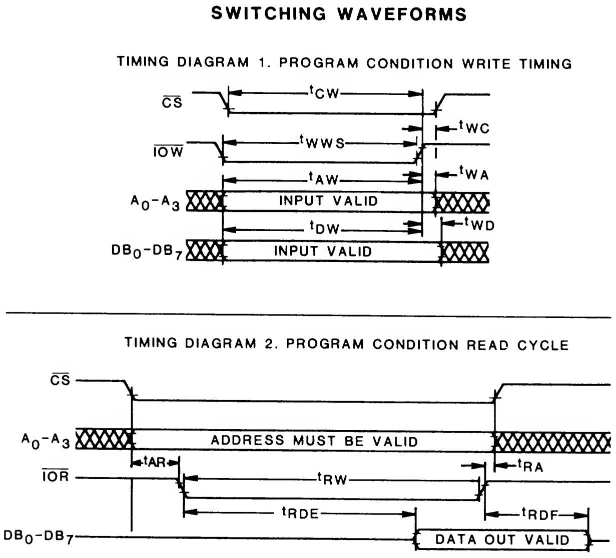

Figure 14-15. DMA Controller Bus Timing

The Decoder Gate Array

Most of the logic in the computer is contained in three devices: the CPU gate array, the decoder, and the monitor ROM. The CPU gate array is described in the previous section and the monitor ROM is described in the "Memory Circuits" section of this Chapter.

The decoder gate array is a custom device that replaces many discrete components normally required to support the CPU. It generates most of the chip select signals, including those required for access to video memory. There are no programmable functions on this chip.

System Bus Buffer Gate Array

The system bus buffer gate array provides in one package the necessary interface circuitry to replace numerous buffers and latches required in the computer. It provides functions such as data bus buffering, address latching, DMA addressing, and address decoding. This greatly reduces the number of discrete circuits required in the computer and reduces the overall system power requirements.

Memory Circuits

This section describes the memory circuits, including DMA, of the Laptop computer. A simplified block diagram of the direct memory access IC and memory circuits is illustrated in Figure 14-16.

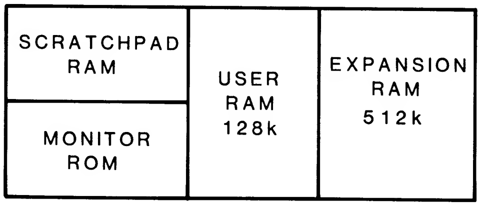

Figure 14-16. Memory Block Diagram

Memory Access Operations

The following description of memory circuit operation explains how memory is accessed and manipulated by the system. Refer to the memory block diagram in Figure 14-17.

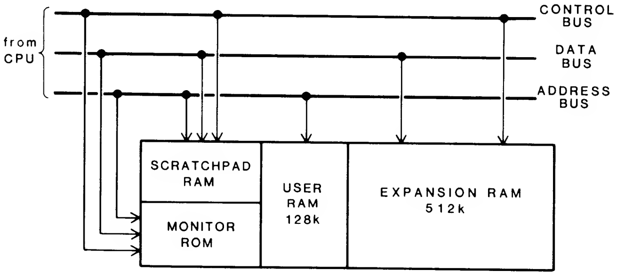

Figure 14-17. Memory Access Block Diagram

A memory access cycle may be initiated by one of three requests: memory read, memory write, or refresh. Contention between these requests, timing, and row and column address selection is handled by the address logic.

The user memory consists of four parts: two banks of 64K and two banks of 256K. The 64K banks are supported by two 64 kilobit × 4 devices each. The 256K banks consist of eight 256 kilobit × 1 devices each. This provides a total of 640 kilobytes of user memory.

Refresh is handled on-chip and is automatically initiated whenever a location is read. During an actual refresh cycle, all rows in all banks are read simultaneously, automatically refreshing them.

Address Multiplexing and Buffering

The address multiplexers provide the means to address all the memory using only eight address bits (nine, if 256 kilobit chips are used) and to buffer the system bus from the memory bus.

When a row is being addressed, A0 through A7 and A16 are present on the multiplex bus as MA0 through MA8. During a column address, A16 through A19 are present as MA16 through MA19. The row address signal on these devices toggles between a logical 1 and a logical 0 to place one group of bits or the other on the multiplex bus.

The memory banks first latch off the low-order group of bits, then the high-order group, and combine them to form an 18-bit address word.

The ninth memory address bit, MA8, connects to pin 1 on each of the RAM sockets. Since pin 1 has no internal connection on 6665 ICs, MA8 is only used when 256 kilobit chips are installed.

When a specific memory location is accessed, every chip in the bank in which that address resides is affected. For instance, if access to the contents of an address in bank 3 were desired, one bit from each of eight chips would contribute either a logical 1 or a logical 0 to form the 8-bit data byte.

To understand how memory is accessed, an understanding of how the memory chip is structured must be developed.

Each dynamic RAM IC is capable of storing 65,536 1-bit words. In order to address 64 kilobits with only eight address lines, it is necessary to multiplex the row address and column address strobe signals.

First, the row address signal is applied. This allows the RAM to latch the eight least-significant address bits off the system bus. Then, a few hundred nanoseconds later, the column address signal is applied and the eight most-significant bits are retrieved to obtain the full 16-bit (or 18-bit for 256K chips) address word.

From a functional standpoint, each 64-kilobit memory chip can be considered a 256- by 256-bit matrix. The address byte is gated by the row address signal which selects the row and the column address signal which selects the column. During a read cycle, one data byte is formed when all the bits from the same row and columns of the eight data chips in a bank are combined.

Table 14-19 describes the bank select logic.

Table 14-19. RAM Bank Select Logic

BANK	FIRST	A19	A18	A17	A16	RASTIM	MEMORY RANGE
0	1	0	0	0	0	1	00000 – 0FFFF
1	1	0	0	0	1	1	10000 – 1FFFF
2	1	0	0	1	0	1	20000 – 2FFFF
3	1	0	0	1	1	1	30000 – 3FFFF
4	1	0	1	0	0	1	40000 – 4FFFF
0	0	0	1	0	1	1	50000 – 5FFFF
1	0	0	1	1	0	1	60000 – 6FFFF
2	0	0	1	1	1	1	70000 – 7FFFF
3	0	1	0	0	0	1	80000 – 8FFFF
4	0	1	0	0	1	1	90000 – 9FFFF

Data Buffer

The memory boards also contain a three-state 8-bit data buffer interfacing the system data bus with the memory banks. When a read cycle is in progress, the buffer is in an active read state and takes the data word from the selected memory address and places it on the system data bus.

When a write cycle is in progress, this buffer takes the data word off the data bus and places it on the memory bus for placement in the selected memory location. If neither a read nor a write operation is requested, the buffer is inactive (data does not flow in either direction). This is also the case when a memory refresh cycle is in progress.

System Memory Map

Memory addresses can range from 0 to 9FFFFH (0 to 655359 decimal), depending on the configuration of the memory board. For instance, the addressable range for a basic system is 0 to 1FFFFH (0 to 131071 decimal) or 128K. The memory map is described in Table 14-20

Table 14-20. System Memory Map

TYPE OF MEMORY	HEXADECIMAL ADDRESS	DECIMAL ADDRESS
1st RAM board	00000 - 9FFFF	0000000 - 0655359
Reserved	A0000 - AFFFF	0655360 - 0720895
Monochrome video	B0000 - B3FFF	0720896 - 0737279
Reserved	B4000 - B7FFF	0737280 - 0753663
Color graphics	B8000 - BBFFF	0753664 - 0770047
Reserved	BC000 - BFFFF	0770048 - 0786431
Reserved: expansion ROM area	C0000 - EFFFF	0786432 - 0983039
MFM-180 scratchpad RAM	F0000 - F3FFF	0983040 - 0999423
Alternate drive buffer	F4000 - F7FFF	0999424 - 1015807
System ROM	F8000 - FFFFF	1015808 - 1048575

Peripheral Support Circuits

The peripheral support circuits used in the Laptop computer include video interface for either a raster-scan CRT or a liquid crystal display, serial and parallel port control, mass storage control, and keyboard interface. The following sections describe of these circuits.

The 6355 LCDC Video Controller

The one major IC in the video section is a low power CMOS display interface IC that has the unique capability of providing control for two different types of displays. The 6355 liquid crystal display and cathode-ray tube controller (LCDC) provides the signals necessary to produce video on the resident LCD in the computer, and standard R, G, B, and I video signals for use with an external raster-scan CRT video display. The display format implemented through the 6355 controller is compatible with the IBM Color Graphics Adapter format.

All of the capabilities of the popular 6845 CRT controller have been incorporated in the design of the 6355 LCDC. However, many additional features have been provided to allow a high level of system flexibility. The most unique feature is the ability to control a liquid crystal display. Other features include a light-pen interface port, 16 × 16 dot size cursor displayable as either a logical AND or EXCLUSIVE OR symbol, 16 level LCD intensity control, and various resolutions of screen modes.

There are many useful modes and functions available to the user that are selectable through registers in this device. These function calls are software-compatible with the IBM PC format. However, the 6355 contains expanded function calls that allow the user to create a more colorful and flexible screen.

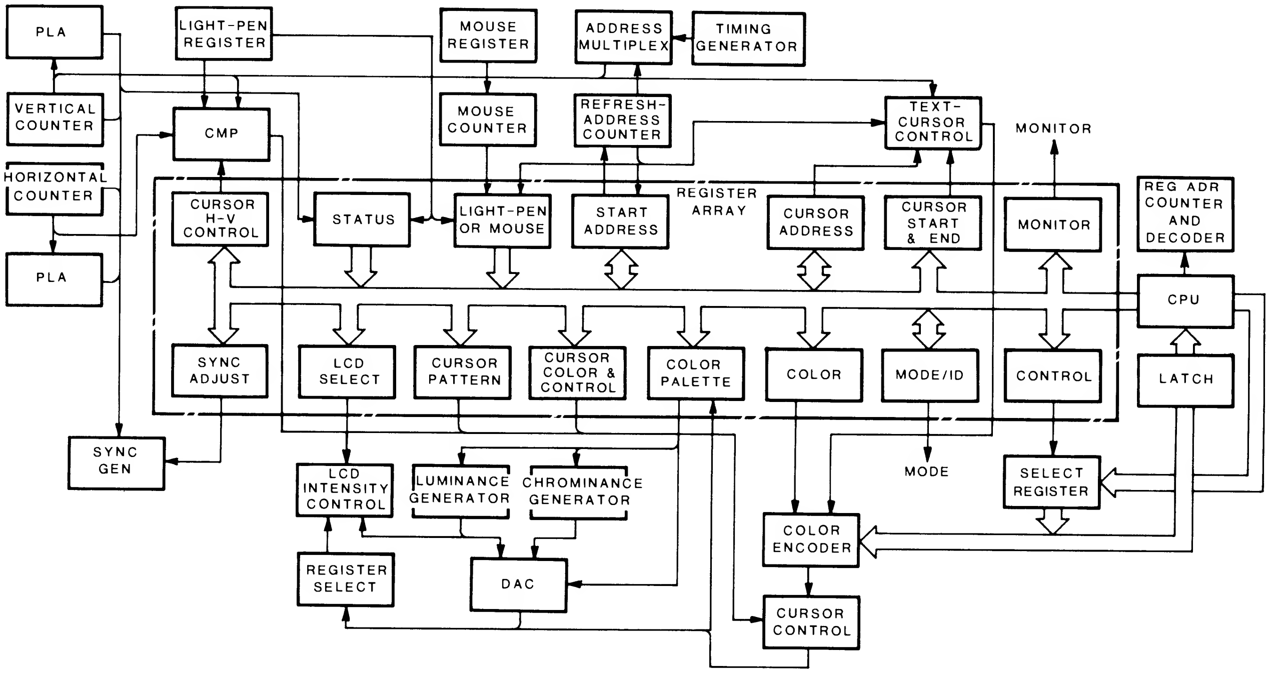

Figure 14-18. Video Controller Block Diagram

I/O Registers

There are numerous registers within this device that may be accessed by the user to establish modes and function calls. Ten of these registers are general purpose, 8-bit I/O devices designed to set up the video interface for proper operation. The purpose of these registers is detailed in Table 14-21.

Table 14-21. 6355 LCDC I/O Registers

REGISTER	ADDRESS (HEX)	R/W
6845 Address register.	4	W
6845 Data register.	5	R/W
Mode control/ID register.	8	R/W
Color select register.	9	W
Status register.	A	R
Clear light-pen register.	B	W
Preset light-pen register.	C	W
Register bank address register.	D	W
Register bank data register.	E	W
Display page register.	F	W

Note that the mode control and status registers have been partially expanded over that of the 6845 CRTC and that the register bank address register, register bank data register, and display page register are new sections that allow additional performance to be gained. A description of each of the I/O registers detailed in Table 14-21 is given in the following section.

6845 Address Register

This register is a write-only register that acts as a pointer to other registers within the device. When the four least significant bits of the system address bus equate the Hexadecimal value of 04, and the I/O select signal is active low, this register points to one of the restricted registers indicated in Table 14-22.

Table 14-22. Restricted Registers

REGISTER	ADDRESS POINTER (HEX)	R/W	FUNCTION
R10	10	W	Cursor start line.
R11	11	W	Cursor end line.
R12	12	R/W	Start address (H).
R13	13	R/W	Start address (L).
R14	14	R/W	Cursor address (H)
R15	15	R/W	Cursor address (L).
R16	16	R	Light-pen (H).
R17	17	R	Light-pen (L).

NOTE: Registers R16 and R17 may also be used as the mouse X-Y counter registers respectively, if that function is desired. When used as the X-Y counter registers, R16 and R17 may only be read and must be cleared after the read operation so that the next set of coordinates provided by the mouse can be placed in the registers.

Note that there are other register values (R0 – R9) that are not detailed in the preceding register summary. These registers are automatically loaded with predetermined values provided by the mode control register. These values provide proper control to allow the video interface to display IBM-compatible 40 × 25 alphanumeric, 80 × 25 alphanumeric, and 80 × 25 graphics video modes.

6845 Data Register

This register works in conjunction with the 6845 address register to transfer data into one of the registers (see Table 14-22) pointed to by the address register. Note that the data register also is used to access registers R0 – R9 that are not detailed in Table 14-22.

Mode Select/ID Register

The mode selection and identification register is an 8-bit register that can be read from or written to. This register is addressed at hex 8 when the I/O select signal is active low, in conjunction with I/O OUT or IN command as determined by the operation.

During a read operation, the identification code (Hex CO) of the 6355 controller is made available to the system CPU. During a write operation, eight bits of data are placed in the register to establish the video mode of operation. These modes of operation, relative to the bit position of the register, are detailed in Table 14-23.

Table 14-23. Mode Select Register Description

BIT	MODE SELECTION DESCRIPTION
0	When this bit is set to a logical zero, the video mode will be 40 × 25 alphanumeric, and when set to a logical 1, the mode will be 80 × 25 alphanumeric.
1	When this bit is set to a logical zero, the alphanumerics mode is activated. When this bit is set to a logical 1, the 320 × 200 graphics mode is activated. Note that bits 4 and 6 provide other graphics mode resolutions.
2	When this bit is set to a logical zero, the color burst and chrominance signals are enabled to provide color video. When set to a logical 1 state, the composite intensity level signals are activated to provide monochrome video.

Table 14-23 (continued). Mode Select Register Description

BIT	MODE SELECTION DESCRIPTION
3	This bit is used to enable the selected video signal when its status is logical 1. A logical zero disables video and should be set whenever a video mode change is to occur.
4	When this bit is set (1) it provides 640 × 200 pixel resolution in the graphics mode. Bit 6 is used in conjunction with this signal to provide color or monochrome video.
5	When this bit is set ot a logical zero, sixteen background colors are available for the video mode. When set to a 1, the BLINK attribute is activated for the alphanumeric modes.
6	When this bit is set (1), sixteen colors are available to the graphics modes with the pixel resolution determined by the status of bit 4. When this bit is set to a logical zero, it determines the pixel resolution for the graphics modes (160 × 200 or 640 × 200) and bit 4 determines whether the color mode is activated.
7	When this bit is set (1) the LCDC is placed in a low-power consumption stand-by mode. All of the data in the internal register structure is maintained until this bit is reset (0).

Color Select Register

The color select register is comprised of six bits that may only be written to. The status of the bits are used to establish color selection for the various applicable modes. Table 14-24 details the functions accomplished by each of the bits in this register.

Table 14-24. Color Select Register Description

BIT	COLOR SELECTION DESCRIPTION
0–3	These four bits are used to select the screen border color for each of the video modes available in the computer. The screen background color is determined by these bit settings in the 320 × 200 graphics mode of operation.
4	When this bit is set to a logical 1, an intensified color set is activated in the 320 × 200 graphics mode of operation.
5	When this bit is set (1), it selects the active set of screen colors for the display. The status of this bit is only used in the medium resolution (320 × 200) color graphics mode of operation.

Status Register

The status register is a 5-bit read-only register that is addressed at Hexadecimal A. The purpose of this register is to provide the system CPU with video status information when polled. Table 14-25 details the status information provided by each of the bits.

Table 14-25. Status Register Description

BIT	STATUS DESCRIPTION
0	When this bit is at a logical zero state, it indicates the “display enable” timing to the system CPU. When this bit is set (1), it indicates that video retrace is occurring and that the display is being held in a disabled state.
1	If an optional light-pen is interfaced to the video controller, a positive-going signal from the device will cause this bit to set (1). If a mouse is being interfaced to the controller, the status of this bit indicates the status of the mouse “Switch 1”. If the status of this bit is zero when using a mouse, it indicates that the mouse “Switch 1” is active.
2	If a light-pen is interfaced to the controller this bit indicates the status of the light-pen switch. If a mouse is used, this bit indicates the status of mouse “Switch 0”. When this bit is set to a logical zero state, the applicable switch is active.
3	During an active vertical retrace period, the status of this bit will be set (1). Note that this bit functions in the same manner as bit 4 if the monochrome video mode is selected.
4	If this bit is set (1) during the monochrome video mode, it indicates that video “dot” information is available. Bit 0 and bit 1 of the register bank address register (not yet discussed) enable the R, G, B, and I signals to function as video “dot” bits. When bits 0 and 1 are both zero, the BLUE dot bit is available. When bit 0 is set and bit 1 is zero, the GREEN dot bit is available. When bit 0 is zero and bit 1 is set, the RED dot bit is available. When bit 0 and bit 1 are both set, the INTENSITY dot bit is available. Note that this status register bit is primarily used for testing purposes to make sure that video is occurring properly, and that the mode select, color select, and other write-only registers are operating properly.

Clear/Preset Light-Pen Register

This register set is comprised of two different registers that are addressed at Hexadecimal B and C. These registers may only be written to and affect only light-pen operation. Whenever the register that resides at hex address B is written to, it causes the light-pen flip-flop to be cleared or reset. Any write operation to the register at hex C causes the light-pen flip-flop to become preset.

These two registers are used primarily for testing purposes to make sure that the light-pen flip-flop and status read functions are operating properly.

Register Bank Address and Data Registers

The register bank is comprised of various registers that are used to establish many of the functions that are available to the user. Theses registers are unique to the 6355 video controller and are accessed through the register bank address and data registers. The required 6-bit address is supplied to the address register which allows access to one of the other internal registers. Data can then be move into and out of the selected register to establish different functions.

Table 14-26 indicates the registers that reside within the register bank. The function of these internal registers is detailed in the section that follows.

Table 14-26. Register Bank Internal Registers

ADDRESS RANGE	REGISTER
0000000-0111111	Cursor pattern data.
1000000-1011111	Color palettes 0-15.
1100000-1100001	Cursor horizontal location.
1100010-1100011	Cursor vertical location.
1100100	Test/Cursor control.
1100101	Monitor Control.
1100110	MONO/LCD control.
1100111	Configuration mode.
1101000	Cursor color select.
1101001	Control data.

Cursor Pattern Data Register — A very unique feature of this controller is that it can display a screen cursor as a logical AND or EXCLUSIVE OR symbol. This 16 × 16 dot pattern is provided by this register and the displayed location is determined by the horizontal and vertical location registers. The BLINK attribute and the display enable/disable data for the cursor is contained in the test/cursor control register.

Color Palette Registers — There are 16 different 8-bit color palette registers (0-15) contained within the register bank structure. Each of these registers are responsible for the colors available to the user in each of the video modes. Sixteen colors out of 512 possibilities may be displayed through use of these registers.

Pixel information from video RAM is used to determine the color to display. Each of the registers are write-only registers that can be written to without causing the display to flicker.

This color palette structure is preset to specific values to be compatible with the IBM PC video format. This occurs whenever a system "reset" occurs. The preset color structure is as follows: black, blue, green, cyan, red, magenta, brown, white, gray, light blue, light green, light cyan, light red, light magenta, yellow, and white.

Cursor Horizontal and Vertical Location Registers — These registers contain the data that allows the system CPU to accurately track the position of the cursor on the CRT screen. Without this X-Y coordinate structure, it would be extremely difficult to control the display.

Test/Cursor Control Register — This is an 8-bit write only register that provides the functions detailed in Table 14-27.

Table 14-27. Test/Cursor Control Register Description

BIT	DESCRIPTION
0	This bit enables cursor blinking when it is set (1). When the status of this bit is zero, blinking is disabled.
1	When this bit is set (1), it enables the cursor to be displayed as a logical AND symbol.
2	When this bit is set (1), it enables the cursor to be displayed as a logical EXCLUSIVE OR symbol.
3–5	These three bits define an offset value that causes the screen raster to be shifted upward in text modes.
6–7	These bits are used for specific controller test purposes and are not considered to user accessible.

Monitor Control Register — The monitor control register is used to place data that sets operational requirements for the video interface. These adjustable parameters are detailed in Table 14-28.

Table 14-28. Monitor Control Register Description

BIT	DESCRIPTION
0–1	The values of these bits are used to determine the vertical line number of the screen. When bit 0 and bit 1 are both zero, there will be 192 vertical lines. When bit 0 is set and bit 1 is zero, there will be 200 vertical lines. When bit 0 is zero and bit 1 is set, there will be 204 vertical lines. When bit 0 and bit 1 are both set, there will be 64 vertical lines.
2	This bit is used to determine the number of horizontal pixels that will be displayed on the screen. If the status of bit 2 is zero, either 640 or 320 horizontal pixels may be displayed. If bit 2 is set, either 512 or 256 horizontal pixels may be displayed.
3	This bit is used to select the type of television standard that is to be used. When bit 3 is set, the PAL/SECAM standard is assumed. When this bit is zero, the NTSC standard is assumed.
4	The status of this bit selects between IBM PC-compatible color or monochrome monitor formats.
5	When the status of this bit is zero, a raster-scan CRT may be used as the video monitor and when this bit is set, an LCD display may be used.
6	The status of this bit allows you to use either static or dynamic RAM devices as video memory. When this bit is zero, dynamic memory may be used and when this bit is set, static memory may be used.
7	When the status of this bit is zero, in indicates that a light-pen is being used as a pointing device. When this bit is set, it indicates that a mouse is being used.

Monochrome/Liquid Crystal Display Control Register — This register is used to select between monochrome and LCD video formats and to provide the proper type of signal for interfacing to each type of device. Table 14-29 details the functions established by each bit in this register.

Table 14-29. Display Control Register Description

BIT	DESCRIPTION
0-1	These bits select an offset number for the vertical display position of the upper half of an LCD display. When bit 0 and bit 1 are both zero, the offset is zero. When bit 0 is set and bit 1 is zero, the offset is 2. When bit 0 is zero and bit 1 is set, the offset is 4. When bit 0 and bit 1 are both set, the offset is 6.
2-3	These bits determine the LCD driver type as follows: when bit 2 and bit 3 are both zero, a dual, one bit, serial type of LCD driver is used; when bit 2 is set and bit 3 is zero, a dual, four bit, parallel type of LCD driver is used; whenever bit 3 is set, a dual, four bit, intensity type of LCD driver is used.
4-5	These bits select the LCD driver shift clock frequency. Eight different frequencies can be established by these bit pairs when used with a dual, four bit, parallel driver interface.
6	When this bit is set it disables the intensity signal to affect the available gray scales that can be displayed. Without the intensity signal, only eight shades of gray may be produced. When this bit is zero, 16 shades are possible.
7	When this bit is set it determines the format of the monochrome video interface. This function occurs in conjunction with bit 3 of the status register which indicates video dot information, and bit 0 which indicates the horizontal sync to make the underline function available. When this bit is zero, the format is determined by the color adapter in use.

Configuration Mode Register — The configuration mode register is used to establish various operating parameters necessary for proper control of either an LCD or raster-scan CRT type of device. Table 14-30 details each bits specific function.

Table 14-30. Configuration Mode Register Description

BIT	DESCRIPTION
0-4	These bits are used to provide different frequencies of the "enable clock" and "write clock" signals that are used for the LCD driver in the LCD mode. In the CRT mode, these bits are used to adjust the horizontal position of the cursor.
5	This bit adjusts the control signal period for the type of LCD drive control circuit.
6	This bit allows the "page mode" function to be active when set. 64K DRAM devices may be used that are divided into four display pages.
7	When this bit is set, it enables 16-bit CPU bus operation and allows the 6355 controller to operate as a slave to the system processor.

Cursor Color Select Register — The cursor color select register is an 8-bit write only register that allows different combinations of the R, G, B, and I signals to produce various colors of the screen cursor. Bits 0-3 affect the logical AND cursor symbol foreground color and bits 4-7 affect the logical EXCLUSIVE OR cursor symbol foreground color.

Control Data Register — This register simply provides various control signals to the system CPU.

Display Page Register

During the "page mode" this 3-bit write-only register selects one of four 16K pages within a 64K byte space. The page mode is not supported on the static RAM system configuration.

Modes Of Operation

There are normally two basic modes of operation. These are the alphanumeric and graphics modes. The 6355 controller supports six different submodes of operation. One of these modes is selected by the mode select register. Table 14-31 indicates the various modes and how they relate to the mode selection register. The following section describes each mode of operation.

Table 14-31. Mode Selection Summary

	MODE REGISTER BIT						
MODE	6	5	4	3	2	1	0
40 × 25 A/N	0		0			0	0
80 × 25 A/N	0		0			0	1
160 × 200 GR	1	X	0			1	0
320 × 200 GR	0	X	0			1	0
640 × 200 GR	0	X	0			1	0
640 × 200 B&W	0	X	1			1	X
640 × 200 CLR	1	X	1			1	X

X = Dont care.

NOTE: Any value for bit 2 will cause the mode to be monochrome, any value for bit 3 will enable video, and any value for bit 5 will cause the BLINK attribute to be active in the alphanumeric modes.

Alphanumeric Mode

In the alphanumeric mode, the LCDC controller uses a two-byte character/attribute format to define each display character. The display formats in either a 40-column by 25-line submode or in an 80-column by 25-line submode. In either submode characters are formed in an 8-dot by 8-dot space. Data in the restricted registers R10 and R11 of the controller determines the cursor pattern width and the blinking period (16 frames).

On a monochrome display, several features are available: reverse video, blinking, highlighting, and 16 levels of gray for foreground and background on an individual character basis. On an RGB monitor, 16 foreground, background, or border colors, blinking, and highlighting are available on a per-character basis.

Using 40 × 25 resolution requires 1000 bytes for character information storage and 1000 bytes for attribute information for one screen-page of information. With 16K of video memory, up to eight pages of screens can be stored. Additionally, since all the video memory is directly accessible by the 6355, extreme flexibility in manipulation of screen contents is obtainable. Using 80 × 25 resolution requires more video memory to be used so not as many screens can be stored.

Whenever bit 5 of the mode register is a logical zero, it causes bit 7 of the attribute byte to become the intensity (I) signal. When bit 7 is set (1), 16 colors are available to the screen background. When bit 5 of the mode register is set, bit 7 of the attribute byte enables blinking of the foreground color for a period of 32 frames.

Note that the logical AND or EXCLUSIVE OR cursor symbols are available in the text modes if so desired. Displaying priority is determined by the contents of the appropriate registers within the register bank and restricted registers of the controller.

The 2-byte character/attribute format is used to define each character display position. These two bytes are mapped into assigned locations in the screen video RAM (this buffer is part of the video interface, not system memory).

Graphics Modes

There are four different graphic modes available to the user. Either the text cursor or the logic symbol cursors can be displayed in any graphic mode. The values placed in registers within the register bank and restricted registers of the 6355 LCDC controller determine which graphic mode of operation is enabled.

160 × 200 Graphics — In this mode of operation, resolution is determined by exciting twice as many picture elements or "pixels" than in the standard 320 × 200 medium-resolution graphics mode. Each redefined pixel size is capable of displaying 16 different colors out of 512 available colors. Color selection is determined by values placed in the color select, mode select, and various other registers within the register bank.

The cursor will exist within a 16-dot by 16-dot area and it will be displayed with medium resolution quality (320 × 200).

320 × 200 Graphics — In the 320 × 200 resolution, each pixel may be one of four colors selected from a palette of 512. Background color is selected through bits 0, 1, 2, and 3 of the color select register. When the status of bit 5 of the color select register is zero, color palette set number 1 is selected for display. When bit 5 is set (1), color palette set number 2 is selected for display. The palette sets are structured as follows:

SET 1	SET 2
Color 1 – Cyan	Color 1 – Green
Color 2 – Magenta	Color 2 – Red
Color 3 – White	Color 3 – Brown

The possible background colors are the same basic eight colors used in low-resolution graphics, plus the lighter shades of these colors which result when intensified, for a total of 16 colors including black and white. In order to obtain the lighter (intensified) shades, the monitor being used must be capable of recognizing the "I" bit.

Each video memory byte contains color information for four pixels in 2-bit pairs as in the following format:

7	6	5	4	3	2	1	0
C1	C0	C1	C0	C1	C0	C1	C0

For a given pixel bit pair composed of bits C0 and C1, Table 14-32 lists the logic that determines which color is selected from one of four colors: the current background color or one of the colors from the current palette.

Table 14-32. Color Select Logic

C1	C0	COLOR SELECT CODE
0	0	Pixel becomes the current background color.
0	1	Pixel becomes color #1 of current palette.
1	0	Pixel becomes color #2 of current palette.
1	1	Pixel becomes color #3 of current palette.

640 × 200 Monochrome Graphics — This high-resolution graphics submode is implemented in a monochrome format with each pixel capable of displaying one of two different colors. Addressing and mapping is the same as medium resolution, but formatting of the data is different. In this submode, each bit, as opposed to a bit-pair, represents a pixel on the screen as shown:

Video memory byte

7	6	5	4	3	2	1	0
1st	2nd	3rd	4th	5th	6th	7th	8th

pixels

640 × 200 Color Graphics — In this graphics submode each pixel may be selected to be one of sixteen different colors. A four-bit color code is supplied by external shift registers that correspond to the R, G, B, and I video signals. Additional memory and support logic can also be used in this mode.

Display Interface

Several different kinds of interfaces are supported for raster-scan CRT displays and several types of interfaces are supported for LCD display drivers. Each of these interfaces are implemented through values placed in registers within the register bank. Table 14-33 indicates the supported interfaces.

Table 14-33. Display Interface Support

CRT INTERFACE	
1	Analog RGB interface for high-resolution, linar intensified color displays.
2	Digital RGB interface for high-resolution color display with separate input.
3	Composite brightness and chrominance interface for a monochrome monitor or color television.
4	IBM PC-compatible digital signal for the IBM PC monochrome monitor.
LCD Interface	
1	Dual, one-bit serial driver interface.
2	Dual, four-bit parallel driver interface.
3	Dual, four-bit, intensity driver interface.

Programming

The operating system in your computer provides the necessary commands for accessing the registers within the 6355 LCDC controller. Note that it is not good programming practice to bypass the operating system by programming specific devices.

The Monitor ROM provides various functions for using the video controller in your computer. With these functions you can cover a full range of capabilities for the device. Additional programming information may be found in the vendor's data sheets, and the Programmers Utility Pack for your version of MS-DOS.

Serial and Parallel Communications

This section describes the major IC's that support the serial and parallel connectors on the Laptop computer.

8250 Serial Port Asynchronous Communications Element

The 8250 Asynchronous Communications Element (ACE) controls the serial communications channel. A block diagram of this, device is shown in Figure 14-19.

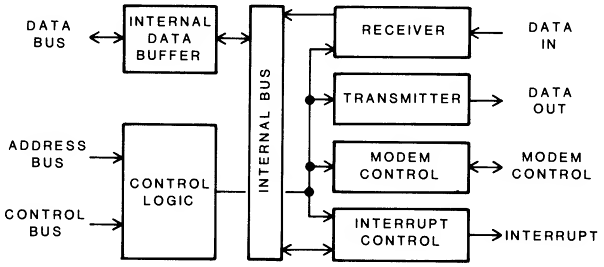

Figure 14-19. 8250 ACE Block Diagram

The 8250 ACE is an NMOS logic device designed to interface a system processor with a serial communications channel. The 8250 receives information through an internal data buffer from the system data bus. Internally, different sections of the device are connected by an internal data bus. This allows the various registers within the 8250 to be programmed to meet the users needs. Input data may be information used to program a register or data meant for output to the serial communications device.

Using system address lines A0 through A2 and various control and timing signals, the control logic monitors the operation of the 8250. The control logic determines whether a register is being programmed or if there is incoming serial data ready to be transferred to the system bus. The control logic section of the device communicates directly with the transmitter section, the receiver section, the modem section, and the interrupt control section, and coordinates the activities of these sections within the 8250.

The receiver section uses the RCLK signal to correctly time the incoming serial data on the SIN line. This information is converted to parallel format and passed to the system data bus by the internal data buffer.

The transmitter section converts parallel formatted data into serial format for transmission. Serial transmission format consists of a start bit, the data bits, a parity bit (if used), and a stop bit sequence. The 8250 adds the start bit, parity bit and stop bits to the transmitted data based on the information programmed into the device. Timing (baud) for serial transmission can be determined by the user.

The interrupt control section of the device signals the host processor whenever an enabled interrupt is active. This allows the 8250 to signal when it requires service from the processor to continue processing data. The modem control section of the 8250 provides the modem control signals used for serial communications with another computer over standard telephone lines.

The 8250 allows a number of programmable options, including interrupts, modem controls, parity, stop bits, and baud rate. Although the 8250 supports programmable baud rates from 50 to 19200 baud, the supported range for PC compatibility is 110 to 9600 baud.

The 8250 ACE operates under the control of the CPU. This device converts 8-bit parallel data into serial form for transmission by the line drivers. The device also converts serial information received from the line receivers into 8-bit parallel data the system can use. The 8250 ACE is programmable and supports a wide range of communications protocols. Programming instructions received by the 8250 establish the baud rate, parity, and handshaking protocols.

The 8250 ACE controls serial communications for the computer. The device can function in either an interrupt-driven or polled configuration with the interrupt-driven configuration being the most commonly used. Separate interrupts are provided for both primary and secondary port address ranges.

Control information for the 8250 is received from the system bus on address lines A0 through A9. This information is decoded and synchronized by two decoders to generate the 8250 ACE chip selection signal. Address lines A0 through A2 of the address bus are connected directly to the 8250 to provide control signals. This information can be used to set programming or communications parameters for the 8250.

Data from the system bus passes through a data buffer before going to the 8250 or from the 8250 to the bus. This data can be programming information for the 8250, data being sent to external devices, or data being received from external devices. The data is converted by the 8250 into serial format for transmission and from serial to parallel when received.

Transmitted data and received data are sent through a set of receive and transmit buffers/drivers. These buffers handle the data transfer between the 8250 and peripheral devices.

Register Selection

In order to access information that is being received or transmitted by the 8250 ACE, the correct registers must be accessed by the system. To accomplish this, 8250 register selection is handled similarly to chip select functions. Register selection is determined by address information placed on address lines A0 through A2. The registers within the 8250 correspond to the address range assigned to this device. Refer to Table 14-34.

Table 14-34. 8250 ACE Register Addressing

ADDRESS	REGISTER
3F8	Receiver buffer (read), transmitter)
3F9	Interrupt enable
3FA	Interrupt ID (read only)
3FB	Line control
3FC	Modem control
3FD	Line status
3FE	Modem status
3FF	Not used

When the 8250 is selected, the information on the address lines is latched to select one of the registers. Changing to the secondary address range (2F8-2FF), shifts the addresses listed in Table 14-35. Once the chip has been selected and a register address has been placed on the address lines, the data in the buffer can be transferred to the selected register within the 8250.

Handshaking

Six handshake lines are available to the 8250 ACE: two output and four input. This makes it possible for the 8250 to function in a wide range of asynchronous or synchronous communications modes.

The input handshake lines are RLSD, DSR, CTS and RI. The RLSD (receive line signal detect) signal is asserted whenever an externally connected modem detects incoming data. The DSR (data set ready) signal is asserted when a modem or other device is ready to establish communications with the computer. The CTS (clear to send) signal, asserts when the external device is ready to accept data. The RI (ring indicator) asserts when an attached modem detects a ring signal on the telephone line.

The two output handshake lines are DTR and RTS. DTR is the data terminal ready signal. When asserted, it will tell the external device that the computer is ready to communicate. The RTS (request to send) signal is asserted when the computer is ready to transmit data. All of the handshake lines are routed through buffer/driver circuits.

Programming

The operating system in your computer provides all the necessary commands for accessing the serial communications channel. It is not a good programming practice to bypass the operating system by programming specific devices.

One aspect of programming the 8250 ACE is important to remember. It has become commonplace to program the 8250 by placing it into a loop back mode. This procedure was originally recommended by chip manfacturers after a bug was discovered in the 8250. Present generation devices do not have this problem and no longer require this procedure.

The Monitor ROM provides four functions, using interrupt 14H, for using the 8250 within your computer. With these functions you can initialize the serial port, send a character, receive a character, or read the serial port status. These functions cover the full range of capabilities for the device. Additional programming information may be found in the vendor's data sheets, and the Programmer's Utility Pack for your version of MS-DOS. If you need detailed information concerning this device, refer to the Western Digital data sheet for the WD8250 Asynchronous Communications Element.

The Programmable Peripheral Interface

The PPI (programmable peripheral interface) in this computer is integrated into the CPU gate array. The PPI provides data communications to the keyboard circuits, sends status information to the CPU, and monitors and controls the sound generator timer.

The PPI consists of a control register and 24 programmable I/O lines, PA0 — PA7, PB0 – PB7, and PC0 – PC7. The control register is a write-only register and selects the operating mode of the array. The operating mode determines whether PA, PB, and PC are inputs or outputs. There are three possible operating modes, 0 through 2. In this computer however, only mode 0 is used, so it will be the only one described.

Data communication between the CPU and the PPI takes place on IO data lines. A0 and A1 select the port (PA, PB, or PC) or the control register. $\overline{\text{IORD}}$ or $\overline{\text{IOWR}}$ assert if the operation is a read or a write, respectively. $\overline{\text{KYSW}}$ asserts whenever the array port is addressed (060H – 063H). The RESET line initializes the array at power up.

To program the array for mode zero, the CPU sends 099H to the control register. The array will then have the following status indicated in Table 14-35.

Table 14-35. Printer Gate Array Status in Mode 0

ADDRESS	REGISTER	DESCRIPTION
060H	PA0 – PA7	Input port
061H	PB0 – PB7	Output port
062H	PC0 – PC7	Input port
063H	Control	Write-only register

PB0, PB1, PC4, and PC5 control and monitor the activities of the sound generator. (See the 8253 System Timer description for more details.) PA0 – PA7 interface with the keyboard processor. PB7 is used to clear the keyboard processor.

The Printer Gate Array

The printer gate array is a custom device that replaces many discrete components that would otherwise be required in this computer. It provdes the LCD/CRT toggle signal, power control to the LCD backlight, and control signals to enable/disable the internal modem. It also provides the interface to the parallel port.

Parallel Port

The parallel port is designed to attach printers with a parallel port interface; specifically, printers that are compatible with the electrical standards established by the Centronics Data Products Corporation, a major printer manufacturer during the first years of the microcomputer era. However, other printers can be used with this port. Simply run the MS-DOS CONFIGUR.COM program.

The parallel port has eight data output lines and nine handshake lines. Four of the handshake lines can be programmed as either output or input; the remaining five are input only. In addition, one handshake input line can be used as a processor interrupt. This interrupt can be enabled and disabled under program control.

Finally, the parallel port has loop-back circuits on the output lines. Diagnostic programs can use these circuits to determine fault isolation between the parallel port and the attached printer.

Parallel Port Interface

The parallel port interfaces parallel data from the computer to an external device that recognizes parallel data transmission. The external device usually sends status information back to the computer. The entire parallel port is actually comprised of three individual ports. Port B is a read-only port that drives status signal data from the parallel device onto the parallel data bus of the computer. Ports A and C are read and write ports. Port A is the parallel data port and Port C is the device command port. The read ports associated with A and C are used to perform a check of the data sent out to the parallel device, to make sure, for example, that false data is not present on the data lines as a result of malfunctions in the port.

All data transfers to and from the parallel port and parity disable port are through a bidirectional decoder. The $\overline{\text{WPA}}$ and $\overline{\text{RPA}}$ signals control the direction of data flow.

A flip-flop generates four handshake lines to the printer and a control line for the parallel port interrupt. The handshake lines have the functions detailed in Table 14-36.

Table 14-36. Parallel Port Handshaking Output Lines

LINE	DESCRIPTION
$\overline{\text{STROBE}}$	After the CPU latches data on PDATA0 – PDATA7, it asserts the strobe line to tell the printer that a new byte is present at the printer input.
$\overline{\text{AUTOFDXT}}$	This line goes low to tell the printer to generate an automatic line feed when the print head reaches the end of the line.
$\overline{\text{INIT}}$	The CPU uses this line to initialize some printers. It does this by sending a short pulse (typically 50 uS) to the printer.
$\overline{\text{SLCTIN}}$	A logic zero on this line selects the printer.

The remaining output controls the printer interrupt, IRQ7, by placing the appropriate logic level on ACK.

At power-up, the system bus reset ($\overline{\text{RESET}}$) sends a logic zero to the flip-flop. This forces the four handshake lines high, putting them in their inactive states.

A three-state buffer forms the loop-back port. A diagnostic program can read this port to help determine if a problem is in the computer or in the printer. In addition, two inputs are status lines. The CPU can check pin 13 to determine if IRQ7 is enabled or disabled.

An octal buffer handles the handshake lines from the printer. These lines perform the functions described in Table 14-37.

Table 14-37. Parallel Port Handshaking Input Lines

LINE	DESCRIPTION
$\overline{\text{ERROR}}$	This line asserts when a printer failure occurs. This can be something simple, such as the ribbon needs changing; or, if the printer has built-in diagnostics, a faulty circuit indication. The CPU stops sending characters until the error is fixed.
SLCT	The printer raises this line (select) to acknowledge the $\overline{\text{SLCTIN}}$ signal from the computer.
PE	The printer asserts this line (paper end) when it runs out of paper. The CPU stops sending characters until PE goes low.
$\overline{\text{ACK}}$	The printer asserts this line to acknowledge that it processed the byte on PDATA0 – PDATA7 and is ready to receive another character.
BUSY	This line asserts if the printer is busy and cannot accept a data byte at the time $\overline{\text{STROBE}}$ occurs. This can happen if the print head is moving (such as during a carriage return) or if the printer is in the off-line mode. The CPU stops sending characters until the BUSY line goes to its inactive state.

Mass Storage

Mass storage is referred to as the ability of some device to permanently retain large amounts of data within some type of media format. There are various types of formats commonly used such as magnetic disks and magnetic tapes. Some new and innovative mass storage formats are becoming popular such as compact laser disk storage. The devices used in this computer are floppy disk drives that use a magnetic disk structure to retain data.

Disk Drives

The disk drives are storage devices designed to transfer information into and out of the computer's memory. The computer will support one or two built-in disk drives, which are identified by most operating systems, such as MS-DOS, as drive A and drive B. In dual drive systems the left drive is drive A and the right drive is drive B. If only one drive is installed, it is drive A.

This computer will also support an optional external drive. Under MS-DOS, this drive is identified as drive C.

The drives used in this computer are double-sided (each containing one read/write head), double-density 3.5-inch floppy disk drives capable of recording up to 720 kilobits of data. One index hole is recognized for soft-sector recording and playback. The disk rotates at 300 rpm.

The read/write heads have three ferrite cores (a core is shaped similar to a split ring), one of which is the read/write core. Erase cores on both sides of the read/write core erase the space between the tracks, providing tunnel erase and a consistently low signal-to-noise level. You should note that these floppy disk drives are very similar to conventional 48-tpi drives except that the width of the read/write core is narrower, resulting in a correspondingly narrower track width. The narrower width of the tracks on disks that have had data recorded on them produce a weaker signal than tracks that have been recorded on 48-tpi drives.

The heads are spring-mounted and face each other. The disk rotates between them. If the computer is moved, cardboard shipping inserts should be inserted into the drives and the door closed.

During power up, a minimum of 200 ms must pass between the time DC power is applied to the drives and any operation begins. The computer's Monitor ROM performs the power-up self-tests during this period. After the self-tests are complete, the drive motors are turned on and the heads are stepped in and then out until the track 00 signal is detected. This is done to synchronize the physical position of the read/write head with the electronics after the computer has been turned on. This action produces the slight noise heard from the disk drives during the power-up sequence.

Floppy Disk Controller Gate Array

The floppy disk controller gate array provides an interface between the CPU and the disk drives. When reading from a disk, the controller converts the serial data bit stream from the disk drive to parallel data and sends it to the RAM. When writing to a disk, the opposite occurs, the controller receives parallel data from the RAM, converts it to serial, adds timing signals, and sends it to the selected drive. Refer to Figure 14-20 for a block diagram of the FDC.

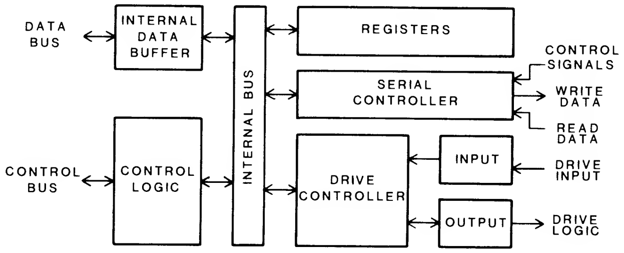

Figure 14-20. Floppy Disk Control Block Diagram

CPU/FDC Logic

On the CPU/controller side, the FDC provides I/O address decoding and an 8-bit parallel data path to the system data bus. In addition to the disk read/write data, the computer uses the data path to monitor the status of the disk drive and controller. The CPU also uses the data path to program the floppy disk controller and send control signals to the disk drive. The computer uses DMA instead of routing data through the CPU. This ensures a fast data transfer.

There are five status registers in the FDC. However, the main status register is the only one that can be directly addressed by the CPU at any time. The remaining four are available only at the end of a data transfer. How many of these four are read depends on how the FDC is programmed.

Hard Reset — The signal on bit DB2 of the data bus is set low by the CPU at power up. The data bus signals are latched to the FDC data bus through a data transceiver. These signals hold the floppy disk controller in a "reset" condition until a software controlled command changes the status of bit DB2.

Chip Selection — The signal on bit A1 of the main address bus is buffered to provide the chip select signal to floppy disk controller.

Register Selection — The signal on bit A0 of the main address bus is buffered to enable the internal data register (A0 = 1) or internal status register (A0 = 0) to transfer its contents to the system data bus. When a read cycle is active, the data register can be written to.

Direct Memory Access (DMA) — The floppy disk controller originates a request for a direct memory access to the DMA controller function of the CPU gate array. The direct memory access request (DRQ) signal is delayed to provide 2 MHz clock signals to the delay flip-flop. The delayed signal (now DRQ2) is buffered and driven to the DMA controller.

When the CPU recognizes the controller request (DRQ2) signal, an acknowledge signal (DACK2) is provided to the floppy disk controller The DACK2 signal enables the internal registers of the controller to be read from or written to.

When a data transfer is completed, the floppy disk controller issues an interrupt request (DISK INT) to cause the controller operation to stop. The interrupt signal is buffered and sent to the programmed interrupt controller.

FDC/Disk Drive Logic

On the drive side of the controller, the FDC has the necessary circuitry to shape the data read/write signals. There are handshake circuits to indicate track zero, disk write-protect status, and the presence of the disk index hole. The FDC also provides drive control lines to position the read/write head, control the spindle motor, and select the disk side to be accessed.

Disk Drive Selection — Disk drive selection is made through the decoded outputs of various devices. Two separate internal disk drives can be connected to the floppy disk controller. The CPU sends data to the system bus through a data transceiver. The status of data bits FD0 and FD1 are latched by a flip-flop and provide the drive select (DS0) and drive select (DS1) information.

Beginning of Track Indication — The INDEX signal from the selected disk drive is sent to the floppy disk controller. The resulting IDX signal indicates the beginning of a track on the disk.

Read/Write Head Selection — A high head select output signal providing the SIDE signal to the system. When SIDE is high, the head on track 0 is selected.

Disk Track Seek and Read/Write Modes — The floppy disk controller specifies either a seek or a read/write mode. In the seek mode, the head in the disk drive can be moved to a different track of the disk. The read/write mode allows data to be read from or written to a track on the floppy disk.

Head Stepping — A stepping pulse is used to move the read/write heads track to track. The SEEK signal enables the stepping pulse signal (STEP). The STEP signal is sent to the disk drive to cause the heads to move. When the head is stepped to track 0, the TRACK0 signal becomes active-low indicating to the floppy disk controller that track 0 of the disk has been found.

Write Data — Write data is converted from parallel to serial form by the Floppy Disk Controller, buffered, and synchronized with the WGATE signal by external logic.

The Write Enable (WE) signal from the floppy disk controller is is sent to the disk drive and enables the drive to accept data. Data is written onto the disk at a rate determined by the WCK frequency. The WCK signal that governs the write-data transmission rate.

Write Protection — During the write cycle, if the disk has a write-protect tab on it, an active low signal will be generated indicating to the floppy disk controller that the disk in use cannot be written to.

During the write cycle, the direction (DIR) and stepping pulse (STEP) signals will not move the read/write heads in the disk drive.

Read Data — During the read cycle, serial data is received from the disk drive through a floppy disk data separator. This chip separates the serial clock signals from the data signals as they are received from the disk drive. A reference clock input of approximately 4 MHz is derived from a crystal oscillator circuit. An internal division ratio of the reference clock is automatically implemented to provide the correct frequency of the signal. The FDC converts the serial data into a parallel format and sends it to the D0 – D7 data bus through an octal data transceiver.

While the read cycle is active, DIR (direction) and STEP signals are disabled so that the read/write heads are not affected.

Timing — Y1 is a 4 MHz oscillator that provides timing signals to a binary counter, floppy disk data separator, and the floppy disk controller. The counter provides divided frequency outputs to the floppy disk controller and the delay flip-flop.

Power — Power for the disk drives is supplied by the power control board but is controlled by the CPU and the keyboard processor. The drives require 12V and 5V sources in addition to the control signals to operate. The 5V power source is controlled by the keyboard processor that monitors the keyboard for activity. If there is no activity, various circuits throughout the computer are disabled, including the FDC. This is a built-in feature designed to conserve power. The appropriate instruction is output from the keyboard processor to a D-type flip-flop. The outputs of the device control the +5 VDC supplies. The +12 VDC supply control signals are latched and output to the power control board.

Keyboard

This section provides information on the keyboard. A brief description of keyboard operation is provided in addition to a description of the keyboard processor. For detailed information on how to operate the keyboard refer to the Keyboard section in Chapter 2.

Operation

The keyboard processor allows maximum flexibility in defining keyboard operation. It returns specific hexadecimal codes rather than ASCII codes when keys are pressed. In addition, all keys except the control keys are event-driven and generate both make and break codes (for example, key number 1 in the matrix causes the processor to send 01H when the key is pressed and 81H when it is released. This allows the system to generate a code with or without one or more control keys (SHIFT, CTRL, ALT, FN) pressed, or as the key is pressed or released.

The codes generated by the keyboard processor are modified by the BIOS before being sent to the rest of the system. Because you can program the BIOS to produce any desired code from any key, the code produced by the keyboard for a particular applications is independent of the computer. If you suspect that the keyboard is not responding correctly, use the built-in diagnostic tests to check its output.

Keyboard Processor HD63705

The keyboard circuit is found on the main board, which contains the CPU gate array and initial support logic and the keyboard processor. The keyboard itself contains the key assemblies.

The initial support logic consists of the buffers for the keyboard reset (RESET), keyboard clock (KYCLK), and keyboard data (KYDATA) lines. The processor is a custom programmed device that takes these signals and derives the strobe scanning signals for the keyswitches. It also contains the logic to decode the results of any key closures and convert them into corresponding key codes. The keyboard is arranged in a 5 × 15 matrix, with diodes arranged to prevent incorrect signals from being sent when two or more keys are pressed at the same time. The BIOS will then convert these keycodes into the appropriate control signals and/or ascii codes for use by the computer.

Part IV — Service

Chapter 15
General Maintenance

The remaining chapters deal with service-oriented matters and should not be used by the user, except to help identify problems. Do not perform any of the disassembly or installation procedures unless you are a qualified service technician.

This chapter discusses general maintenance, including battery procedures. If trouble is encountered, refer to Chapter 16 to determine if the problem is user-, software-, or hardware-oriented. Chapter 17 describes the diagnostic tests that are built into the computer and how to run them. Chapter 18 provides installation instructions for the add-on options. Chapter 19 provides disassembly procedures and a parts list of major components.

Maintenance

The computer is designed to work well and last a long time. As such, no regular maintenance schedule needs to be maintained. Use the following instructions to clean the computer.

- Before cleaning the computer, disconnect external power sources and turn the computer off.

- Cleaning should be limited to wiping off the cabinet with a slightly dampened cloth. A mild detergent can be used, but do not spray any liquids or foam cleaning solution directly on the computer or its cabinet parts.

- Make sure the computer is completely dry before reconnecting any external power sources or turning it back on.

Battery Care

The nickel-cadmium battery pack is designed to provide full operating voltage and current for a maximum length of time. If battery charging is not correctly handled, you may experience shortened battery life. This section will describe how to remedy that problem.

All ni-cad batteries have an operating ledge, which is the point at which the ni-cad will no longer supply an operating current. This ledge is normally located at the point where the ni-cad reaches its lowest level of charge before it must be recharged. Under certain conditions this ledge will rise to the point that the battery can be operated only a short time (sometimes, as short as ten minutes) before it fails to deliver adequate voltage and current and must be recharged.

As long as you remember the following points, you should not experience high operating ledge levels in the battery pack.

- When you use the battery pack as a source of power for the computer, always use it as long as possible before plugging in one of the external power adapters or recharging the batteries.

- Charge a discharged battery pack from eight to twelve hours. Eight hours will fully charge the battery pack. Twelve hours will start overcharging the battery pack. Overcharging the battery pack will raise the operating ledge and shorten the overall life of the batteries.

  CAUTION: Severely overcharging the battery pack can destroy it entirely and damage the computer. **Never** charge the battery pack for more than twelve hours.

- You may operate the computer from an external power source without overcharging the battery pack. However, you should disconnect the adapter whenever you turn the computer off, unless you are charging the battery.

- If the computer is not going to be used for portable operation, remove the battery pack entirely and operate the computer only from an external voltage adapter.

Once the operating ledge has moved to a high level, the battery pack must be fully discharged, sometimes several times. The following procedure will cause the ledge to move back to its normal position, allowing you to use the battery to its fullest potential.

1. To fully discharge a ni-cad battery pack, operate the computer until the system shuts down. To provide the highest amount of battery drain, use the read boot track test from the built-in diagnostic test menu (refer to Chapter 17).

2. When the drive stops operating, continue to operate the computer until the low-power indicator stops glowing.

3. Recharge the battery for a minimum of 8 hours but not more than twelve hours.

At this point, you can attempt to use the battery pack for portable operation. However, if the battery pack continues to fail to deliver a normal operating life on a single charge, repeat the full discharge-recharge process three times.

If the battery continues to deliver an operating voltage for a very short time, typically less than ten minutes on a full charge, one of the cells of the battery pack is probably shorted. If this is the case, the battery pack must be replaced.

Before replacing the battery pack, test the output voltage when it is not connected to the computer. A fully charged battery pack will supply more than thirteen volts DC under no load.

If the battery pack tests ok, then check how the computer is being operated. Excessive disk input/output will greatly reduce battery life. To reduce the amount of disk input/output operations, consider using memory as a temporary disk. The MS-DOS documentation for version 3.2 describes how to set up a ramdrive device driver. Other versions and operating systems have similar features.

To replace a defective battery, refer to Chapter 19, which describes how to remove the battery.

Main Board Controls

There are two controls on the main board that adjust the output signals for an external monochrome monitor. They require an oscilloscope to make the adjustment. Normally, you would make the black level and contrast adjustments with the monitor's controls. If you cannot obtain a usable display with the monitor's controls, try a different monitor before disassembling the computer and attempting to adjust these two potentiometers.

Disk Drive Configuration

The 3.5-inch disk drives used in the computer have only one switch that needs to be configured. It is used to determine if the drive is used in the disk drive A position or disk drive B position. As with the two potentiometers, the computer must be disassembled before you can reach these switches. Since they are correctly set at the factory, you would normally never need to access them, except when you replace a defective drive.

Refer to Figure 15-1 for the location of the switch on the drive. Select DS0 if the drive will be used in the disk drive A position. Select DS1 if the drive will be used in the disk drive B position. The drives cannot be configured with other drive designators; they are reserved for the external 5.25-inch floppy disk drives and the rigid disk drive partitions.

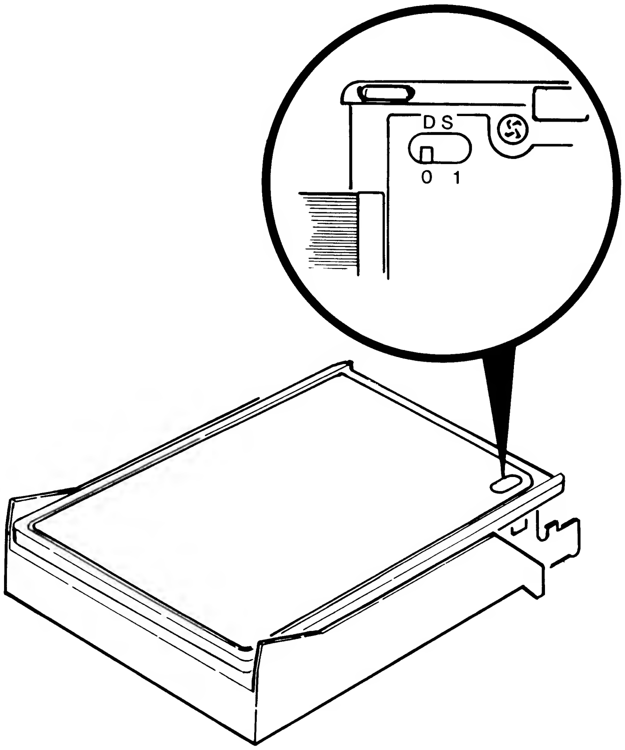

Figure 15-1. Disk Drive Configuration

Chapter 16
Troubleshooting

Concepts of Troubleshooting

Troubleshooting is the logical approach to problem solving. While some problems are obvious and may be resolved very quickly, others can be intermittent and very difficult to locate.

Approximately 90 percent of the time a technician takes in repairing faulty equipment is spent identifying the problem while the remaining 10 percent is actually spent repairing the fault. This chapter is intended to help the technician quickly identify and repair any problem that could occur with this computer.

Problems encountered with any computer can be attributed to one of three areas: operator, software, or hardware (firmware is considered hardware where troubleshooting is concerned).

Troubleshooting Procedure

The following procedure is in a question-and-answer form, with a yes or no answer provided for most steps.

When the response to the question is yes, the YES column will direct you to the next step. Likewise, if the answer is no, the NO column will direct you to the next step. See step 1 for an example.

When a "—" is shown in either or both columns, the fault has been identified and/or corrected, or no action is required. In these cases, the instructions in the step will direct you back to the step that is used to retest the fault area that has been corrected. See step 1a for an example.

Step 1: General

STEP	PROCEDURE	YES	NO
1.	Is there a customer complaint?	1a.	1b.
1a.	Is the complaint, "I can't get the computer to come on," or, "I can't boot a disk"?	1b.	1c.
1b.	Perform a "sense" test of the computer. Check for physical damage to the cabinet, disk drives, keyboard, and LCD screen. Check for mechanical operation of the disk drives and for disks jammed in the disk drive(s). Also smell the computer for indications of overheated or burned components. Do you detect any possible damage?	1d.	1e.
1c.	Proceed to "Common Customer Complaints" in this chapter.	—	—
1d.	Locate and replace damaged parts or free any jammed disk. Return to step 1.	—	—
1e.	Turn on the computer. Normally, the disk drives will be turned on and synchronized with the system, making a relatively quiet buzzing sound that lasts for only a moment. Do you hear anything unusual?	1f.	1g.
1f.	Locate the source of the noise and correct the problem by either fixing a mechanical failure (usually the disk drive) or replacing the faulty component. Return to step 1.	—	—
1g.	Did the computer come on?	2.	1h.
1h.	Proceed to "Power Problems" in this chapter.	—	—

Step 2: Operational

STEP	PROCEDURE	YES	NO
2.	Can you boot a bootable disk?	2a.	2b.
2a.	Run the disk-based diagnostics fast test. If no errors are found, have the customer check the software and/or operating procedures. If errors are found, repair them and return to step 1.	—	—
2b.	Run the user-executed disk read test (refer to Chapter 17 for instructions). Did the test pass?	2c.	2d.
2c.	The computer can read the boot track of a disk placed in boot drive A. Try step 2 with a different disk. Does step 2 still fail?	2d.	2e.
2d.	Run the user-executed memory test. Did it pass?	2f.	2g.
2e.	Step 2 was originally attempted with a bad disk. Since the test now passes, run the disk-based diagnostics and review the material in Disk Problems in this chapter.	—	—

2f. Proceed to Disk Problems in this Chapter. — —

2g. Replace the main board if the memory failure occurred in the first 128K, the expansion memory board if the failure occurred above 128K. Return to step 2d.

Common Customer Complaints

If you cannot turn on the computer or boot a bootable disk, use the troubleshooting procedure presented earlier in this chapter.

Customer complaints can range from disconnected cables to intermittent problems associated with software. In examining a customer complaint, the service technician must determine whether the complaint is valid (and not an operator error) and then whether the complaint is hardware or software oriented. The most commonly expected customer complaints are printed in italics with an explanation following.

The computer doesn't operate very long on the battery. The complaint usually results from misuse of the battery. Review the battery care material in Chapter 15 with the operator. If this complaint persists, use the material in Chapter 15 to test the battery.

I can't get the computer to turn on. Make sure the computer has either a fully charged battery pack or is plugged into an operating external adapter. Check the output voltage of the battery pack. It must measure at least +12 volts DC. Check the power source if an adapter is used and finally, check the output of the adapter.

NOTE: A defective AC adapter is not cost-effective to repair; it should be replaced.

I can't see anything on the screen. Make sure the computer is turned on. Try adjusting the CONTRAST and BRIGHTNESS controls. Toggle the LCD/RGB key (FN-F10). If the computer still does not display anything, try resetting to the Monitor program and adjusting the controls. If you still cannot get a display, toggle to the RGB video mode and connect an external monitor. If there is no output, replace the main board and retest the computer.

If you have RGB video output, but cannot toggle the LCD/RGB key and produce a display on the LCD, disassemble the computer and check the cables going from the main board and power supply to the LCD.

The disk I want to boot prints No System on the screen. The disk is formatted, but does not contain an operating system. Explain this to the operator or refer the operator to the MS-DOS manual.

The screen is too dark. Adjust the brightness and contrast controls for the best display. If the computer does not have a backlight, room lighting conditions may be insufficient for the display.

The characters on the screen are too light. Adjust the brightness and contrast controls for the best display. If the computer does not have a backlight, room lighting conditions may be insufficient for the display.

I can't see all of the display. It is possible that the operator is using a program that automatically checks for a color controller and goes into a color mode if it finds one. In this case, have the operator use the FN-F8 and FN-F9 keys to switch LCD display palettes. If the complaint persists, check the output with an external monitor. If the complaint is valid, replace the LCD panel and/or power supply.

Disk Problems

Disk problems experienced by the operator can come from a number of areas, most of which are preventable. Two problem areas are common: disk that are stuck, usually because they were inserted incorrectly; and disk read problems, which result in booting problems, bad data, and several different types of error messages. By exercising normal disk care, the operator can avoid all of these problems.

Stuck Disks

There are several reasons that a disk will jam in the computer. The following discussion covers various causes of jammed disks and how to remedy the situation.

CAUTION: Under no circumstances should you forcibly attempt to insert a disk or push the disk drive into operating position. If you force a disk into the drive, you may damage the disk, the drive, or both. If you force the disk drive into its operating position, particularly if the disk is not inserted fully into the drive, you could jam the disk drive into the computer, in which case the computer would have to be disassembled and the drive removed.

Disk Ejection Problems

Two problems may cause a disk not to eject.

- The first and most common reason a disk will not eject is that the drive has not been pushed into its operating position, latched, and released back into the load/unload position.

  Try pushing the disk drive back into the operating position. Do not force the drive if it does not move freely into the operating position. It is possible that the disk is not fully inserted into the drive. If this is the case, proceed to "Disk Insertion Problems."

- The second reason is that the disk latch may not release. If the disk will not eject, take a pair of tweezers and carefully grasp the disk and gently remove it from the drive. Once the disk has been removed from the drive, proceed to Disk Insertion Problems to check out the drive and disk.

  If the disk will not move and the tweezers slip off the disk, then the disk is jammed into the drive. Take the computer to a qualified service center.

Disk Insertion Problems

It is possible that you may not be able to fully insert a disk into the drive. If the disk will not slide all the way into the drive, you will not be able to push the drive into the operating position. Do not force it.

If the disk slides all but about an eighth of an inch into the drive, then the disk latch has latched. There are three ways that the disk latch can be latched.

- The first and most common way that the latch can become latched is that the operator attempted to place a disk into the drive with the wrong end first. With the disk placed in the drive with the wrong end first, you will probably note that the metal disk cover can be seen sticking slightly out of the drive. This indicates that the disk has been inserted into the drive with the wrong end first.

  To correct the problem, remove the disk, push the drive into the operating position so that it latches, and release the drive back into the load/unload position. This will release the latch so that the disk can be properly inserted all the way into the drive.

- The second, but less common way that the latch can be latched is that the operator correctly placed the disk into the drive, but then removed it without first pushing the disk drive into the operating position.

  To correct the problem, remove the disk, push the drive into the operating position so that it latches, and release the drive back into the load/unload position.

- The third way the latch can be latched is that the latch, for some unknown reason, is binding. This is very unusual, but can be caused by forcibly attempting to insert a disk into the computer incorrectly or attempting to push the disk drive into the operating position if the disk has not been fully inserted.

  Before taking the computer to an authorized service center, make sure the latch is not binding. Remove any disk that might be in the drive, push the drive into the operating position so that it latches, and release the drive back into the load/unload position. Read the instructions on inserting disks into the drive and attempt to place a disk into the drive correctly. If the latch is still latched and you cannot get it to release using this procedure, take the computer to an authorized service center for service.

If the disk slides all but about a half an inch into the drive and wants to eject, then the disk is upside down. Remove the disk and note the position of the disk hub. It should face down, toward the keyboard.

Disk Read Problems

Five areas can give the operator disk read problems: incompatible operating system or format, unformatted disk, formatted data disk, magnetically altered disk, and physically damaged disk.

Incompatible operating system or format — Disks are transportable from computer to computer when those computers share a common size and disk packaging arrangement. As such, 5.25-inch disks, whether they have MS-DOS or some other operating system, can be placed in drives connected to this computer. The same is true for 3.5-inch disks.

Some operating systems format the disk differently than MS-DOS and as such, the disk formatted by a different operating system cannot be read by MS-DOS and vice-versa, except by special software. In most cases MS-DOS or the computer will return a error message that indicates a bad disk.

Unformatted disk — New disks are not formatted. If the operator attempts to use an unformatted disk, the drive will not have any references with which to work. As such, error messages that indicate a bad disk will be generated.

Formatted data disk — Disks that are formatted as data disks contain a short program in the boot track that will print "No System" on the screen and lock up the computer. This prevents the computer from attempting to execute data as if it were a program. However, an inexperienced operator may misinterpret this message and call for service, which is not needed. A disk that contains the operating system on it must be used for booting this computer.

Magnetically altered disk — Disks that are carelessly handled can have the magnetic patterns on them altered. When this happens one or more of the disk error messages described in Chapter 17 can result. The operator needs to review the disk care information in Chapter 7 and avoid placing the disk(s) anywhere they may be affected by sunlight or magnetic fields.

Physically damaged disk — Disks may be damaged by fingerprints, dust, dirt, moisture and liquids, and rough handling. The operator should avoid using damaged disks because they can also damage the disk drive in which they may be placed.

Power Problems

Most power problems will come from incorrectly charging the battery or damaged cables. While physical damage is often visible, excessive flexing of cables can result in hidden damage. Cables that are broken internally will probably be intermittent. You can check this kind of cable problem by flexing the cable and observing the results on a meter. The wiring of the computer will allow the computer to operate from both the battery and an external power adapter at the same time. If the adapter supplies more voltage than is present in the battery pack, then the batteries will charge. If the external voltage is less than is present in the battery pack, then the computer will continue to operate, as long as sufficient voltage is in the battery.

If you suspect that you may have a cable problem with an external adapter, use a voltmeter to measure the output voltage of the adapter while you flex the cables. Adapters with damaged cables should be replaced.

NOTE: A defective external power adapter is not cost-effective to repair; it should be replaced.

If the operator experiences a short battery life, review the power section of Chapter 7.

Chapter 17

Diagnostics

One of the advanced features included in the Laptop computer is the use of firmware based diagnostic routines. There are two sets of routines that are commonly used, powerup self tests and user diagnostics. This chapter explains some of the features of these routines and provides information on the error messages that a user is likely to encounter.

Automatic Powerup Self-Tests

During the initial powerup sequence, the Monitor program performs a number of tests to make sure the computer is ready to function, including a self-test of all circuits in the system. In addition, all circuits are initialized and the disk drive heads are synchronized with the rest of the system. If a malfunction is detected, one or more messages is sent to the system's display to alert the operator of a problem. If the video portion of the computer is functional the message will be displayed on the screen. The messages are summarized in Table 17-1. You should refer to this table and Chapter 16 if you encounter any difficulties in operating your computer.

When the tests are sucessfully finished, the computer will attempt to load sector 1 of track 0 into memory and execute it. This is the automatic boot procedure (autoboot). When autoboot starts, a disk must be placed in the drive and the door must be shut within 20 seconds or an error message will be displayed (see the first message in Table 17-1). Refer to Chapter 8 for additional information concerning system drive startup options.

Error Messages

Table 17-1. Error Messages

MESSAGE POSSIBLE PROBLEM

```
+++ DISK ERROR: Drive not ready! +++
```

This error message is usually generated when the computer attempts to boot the operating system from the disk and there is not a disk in the disk drive. Make sure that the shipping insert has been removed from the disk drive. Also make sure that there is a disk in the drive, that it is inserted correctly, and that the drive is fully closed. If this message appears and a disk is properly inserted in the drive, try another disk or try booting from the other disk drive, since this message can also be caused by a faulty disk or defective disk controller circuit.

```
+++ DISK ERROR: Bad disk controller! +++
+++ DISK ERROR: DMA overrun error! +++
```

These error messages usually indicate a malfunction of the disk controller circuit, but may also be caused by other defective circuits in the system. This could result from an incorrectly formatted disk or a damaged disk.

Table 17-1 (continued). Error Messages

MESSAGE POSSIBLE PROBLEM

```
+++ DISK ERROR: Seek Error! +++
```

This error usually occurs when the drive has moved the read/write head to a new track but was unable to find a matching sector header for the track.

```
+++ DISK ERROR: Sector not found! +++
+++ DISK ERROR: CRC error! +++
+++ DISK ERROR: Invalid address mark detected! +++
```

These error messages normally indicate that an operating system was not found on the disk, that the disk may be faulty, or that the drive or cable may be faulty. First try a different disk and then the other drive. If this error recurs, you should have the drive and/or disk controller circuit checked.

```
+++ ERROR: CPU failure! +++
```

This error message indicates that some area of the CPU logic is faulty. If this error message appeared after the system was reassembled, it is possible that the main board was incorrectly mounted during reassembly and is being stressed in such a manner as to cause this problem.

```
+++ ERROR: ROM checksum failure! +++
```

This message indicates the Monitor program ROM values have been modified from the original material that was programmed into the chip. Turn the computer off, then on again. If the same message appears, have the computer serviced by a qualified service center.

```
+++ ERROR: RAM failure! address:XXX:YYYY, Bit: n, Chip XXX +++
```

This message indicates that the CPU is unable to read from or write to some portion of memory. Make sure the expansion memory boards are properly installed. The chip number displayed is the number of the chip that the computer has associated with the problem. Have the computer serviced by a qualified service center.

```
+++ ERROR: Timer Interrupt failure! +++
```

This error message indicates the possible failure of interrupt controller and/or the timing logic. If this error message appeared after the system was reassembled, it is possible that the main board was incorrectly mounted during reassembly and is being stressed in such a manner as to cause this problem.

Table 17-1 (continued). Error Messages

MESSAGE POSSIBLE PROBLEM

```
+++ ERROR: Invalid Command! +++
```

This error most often occurs when a command entry has been typed incorrectly. Normally, these are syntax errors relating to incorrect spelling and missing spaces or commas. Carefully check the information you are trying to enter at the keyboard.

```
+++ ERROR: Calendar Failure! +++
```

This error indicates that some portion of the circuitry related to the internal real time clock circuits has failed. This could result if power to the circuitry has been removed for a period of three days or more. Power removal means that the internal battery has been completely discharged, or the external power supply has been removed with no battery installed.

```
+++ ERROR: Divide By Zero! +++
+++ ERROR: Overflow! +++
```

These errors typically result when some process has created an illegal processor state. Normally, these errors are encountered in math processes. If encountered during testing they could indicate possible problems in the CPU, the ROM or in system memory.

```
+++ ERROR: Non Maskable Interrupt Received! +++
```

This error results when the system encounters an interrupt level that is not defined internally. It may be possible for this error to be created by using the internal debugger to enter specific code combinations into system memory.

User-Executed Tests

The user-executed tests are identified in the command summary as "Extended diagnostics." To display a menu from which each test may be executed, enter **TEST** at the Monitor program prompt and press the **RETURN** key. The menu is illustrated in Figure 17-1.

```
CHOOSE ONE OF THE FOLLOWING

1. DISK READ TEST
2. KEYBOARD TEST
3. MEMORY TEST
4. POWER-UP TEST
5. EXIT

ENTER YOUR CHOICE:
```

Figure 17-1. Test Menu

The Disk Read Test

The disk read test continuously reads the first sector on track 0 of the default boot drive. This sector is the first sector of the boot track and must be read successfully for the computer to be able to boot a disk.

Although the first sector of the boot track is being read, the disk does not need an operating system on it. Any formatted disk, whether it has the operating system installed on it or not, can be used for this test.

The disk read test is a non-destructive test. This means that if the disk you are testing has an operating system (MS-DOS) or data on the first track it will not be erased or destroyed by the test. During operation, the test will display the test count, indicating the number of times that the first track has been read.

Disks that are formatted for data still have valid data in sector 1 of track 0. The data contains a short routine that, when read by the boot routine, will display "No system" and lock up the computer. This message tells the operator that the disk was formatted, but did not contain an operating system on it.

By continuously reading the first boot sector, you can determine if you have a reliable disk/drive combination. Errors that occur during this test can be attributed to the disk, the drive, or both.

This test can also be used to run down the battery during battery tests. A fully charged battery will run about two hours using this test before the low power indicator starts flashing. Even though the disk motor is running, maximum current is still not being drawn by the disk drive. Maximum current will be drawn with the motor running and the head being moved at the same time.

The Keyboard Test

The keyboard test allows you to press most of the keys on the keyboard and generate a keycode similar to those presented in Chapter 10. Some of the keys will cause the computer to exit the test and not all codes produced are displayed. For the codes generated by this test, see "Keyboard Codes" in Chapter 10 of this manual.

When this test is operating the display will contain several items of information. Each time a key is pressed, if it is a printable ASCII character, the entire display will be filled with the character. In the upper right corner of the screen, the character code will be displayed if the key produces one. If the key does not produce a code, the last code displayed will remain on the screen.

The Memory Test

During the powerup self-tests, only the first and last bank of memory (RAM) is tested. The memory test checks all memory, including video memory, for errors. While the video memory is being tested, various patterns will be seen on the active screen.

The display will normally contain a status count indicating how many times the memory test has been started. This count is updated each time the memory test has been completed for all available memory. In the upper right corner of the screen the current bank of memory being tested is displayed in hexadecimal format. If an error is detected, the test will stop and information about the error will be displayed on the screen. This information is useful to service and technical personnel in the event the unit should require repairs.

The Powerup Test

The powerup test repeatedly runs the self-tests that are executed when the computer is first turned on. The self-tests include a check of all circuits in the system. If a malfunction is detected, one or more messages are sent to the system display to alert the operator of a problem. If the video portion of the computer is functional the message will be displayed on the screen. The messages are summarized in Table 17-1.

While the powerup tests are not exhaustive, some random fault conditions can be detected using this process. The disk-based diagnostics are more useful in exhaustively and repeatedly testing key elements of the computer. For more information on using the disk-based diagnostics, refer to the documentation that accompanies that software package.

Exiting the User-Executed Test Menu

All user-executed diagnostic tests will display a status screen while operating. The ESC key is used for exiting the test program. With the exception of the keyboard test, the escape key first halts the test, and then pressing it a second time will redisplay the user-executed test menu. In the keyboard test, the first time you press the ESC key will redisplay the user-executed test menu.

Once you have exited to the menu, you may return to the monitor program by pressing the number 5 in the keyboard row. This will cause the monitor prompt to be displayed on the screen.

NOTE: The CTRL-FN-ALT-DEL/INS key sequences are not recommended methods to use to exit from these tests. Since the computer is in a test mode, exiting in this manner may produce unpredictable results.

Chapter 18

Installation of Options

The following instructions are intended for service personnel only. If you have purchased one of the options described in this manual, refer the installation to qualified service personnel.

Memory Expansion

The computer is supplied with 128K of internal RAM on the main board and 512K of expansion RAM on an expansion board. The board may be supplied already installed in the computer or as optional accessory to expand the internal memory to a maximum of 640K.

WARNING: To avoid shock hazards, make sure the power switch is OFF and external power sources are disconnected before beginning any disassembly procedure. It is also a good idea to disconnect any peripherals that are connected to the computer to prevent damage to their cables.

Refer to Chapter 19 and remove the battery cover.

CAUTION: Some integrated circuits are electrostatic-sensitive and can be damaged by static electricity if they are handled improperly. Once you remove a circuit board from its protective packaging or the computer, do not handle the board unnecessarily.

Installing the Board Into the Computer

Perform the following steps to install the memory board. The following instructions refer to the module as an assembly:

1. Position the computer as shown in Figure 18-1.
2. Position the assembly as shown in the illustration. Make sure the connector in the memory board assembly is lined up with the connector in the computer.
3. Press the memory board assembly all the way into the connector.
4. Secure the assembly with the screw and fiber washer supplied with the assembly.
5. Replace the battery cover and screws. Be careful not to overtighten the screws.

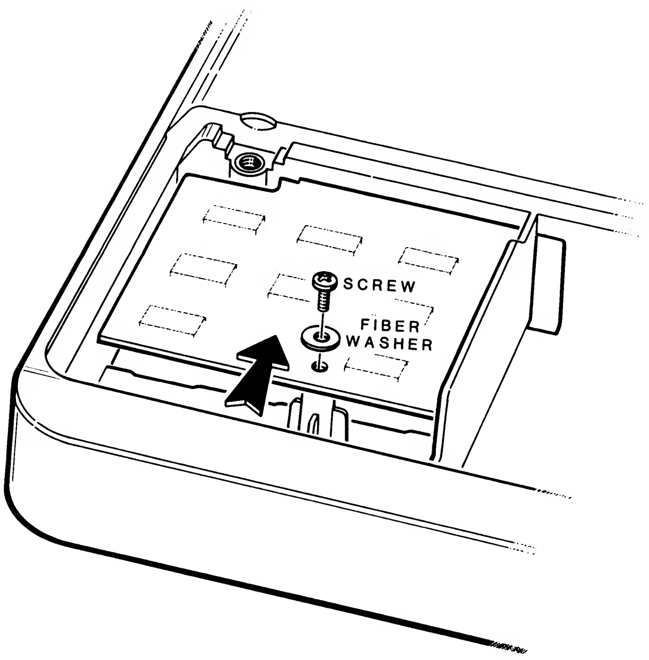

Figure 18-1. Installing the Expansion Memory

Internal Modem

Because more than one modem accessory is available and the design of each board may be slightly different, the following instructions are general and apply to all modem boards that can be installed in the Laptop computer.

WARNING: To avoid shock hazards, make sure the power switch is OFF and external power sources are disconnected before beginning any disassembly or reassembly procedure. Also, disconnect any peripherals that may be connected to the computer to prevent damage to the cables.

CAUTION: Some integrated circuits are electrostatic-sensitive and can be damaged by static electricity if they are handled improperly. Once you remove a circuit board from its protective packaging or the computer, do not handle the board unnecessarily.

1. Refer to Chapter 19 and remove the modem cover.

2. Open the modem accessory package and remove the circuit board.

3. Refer to Figure 18-2. Compare the connector on the modem board with the inset in the figure. One side of the connector is wider than the other. Position the board so that the wider side of the connector is up, as shown in the inset.

4. Slide the modem board into the modem slot of the computer and carefully engage the connector with its socket. Be sure that the connector and socket are properly lined up. Press the board into the computer.

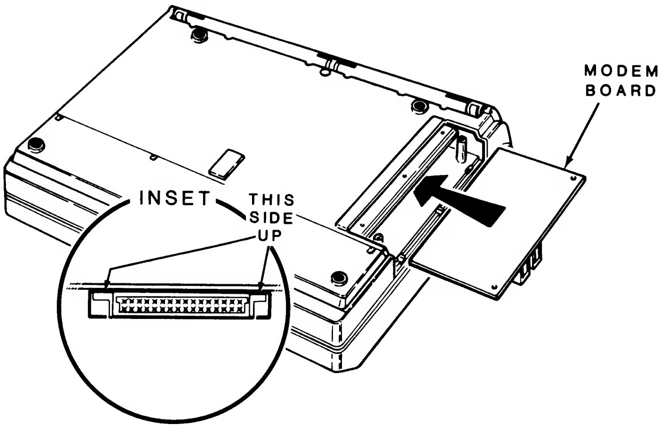

Figure 18-2. Installing the Internal Modem Board

5. Replace the modem cover and screws.

6. Refer to Chapter 2 to connect the modem to a telephone and telephone line.

8087 Numeric Processor Extension

To install the 8087 numeric processor extension, you will need to disassemble the computer and remove the main board. These instructions are intended only for a qualified service technician.

WARNING: To avoid shock hazards, make sure the power switch is OFF and external power sources are disconnected before beginning any disassembly or reassembly procedure. Also, disconnect all peripheral cables connected to the computer before you disassemble it.

CAUTION: Some integrated circuits are electrostatic-sensitive and can be damaged by static electricity if they are handled improperly. Once you remove a circuit board from the computer, do not handle the board unnecessarily.

1. Refer to Chapter 19 and remove the battery, expansion memory assembly, internal modem, bottom cover, and main board.

2. Refer to Figure 18-3 and position the main board as shown. Note the socket into which the 8087 is to be installed.

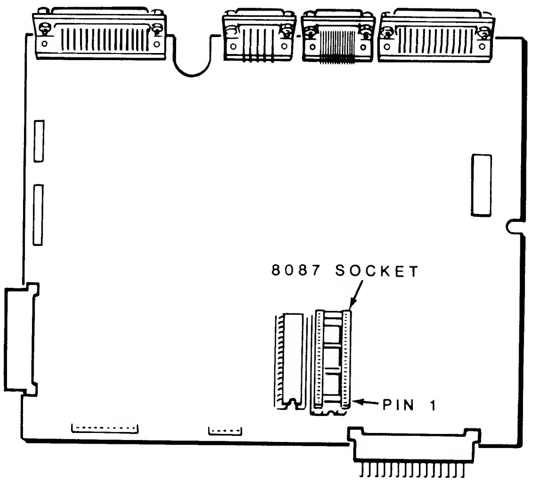

Figure 18-3. The Main Board

CAUTION: The 8087 is electrostatic-sensitive and can be damaged by static electricity if it is handled improperly. Once you remove the IC from its protective packaging, do not lay it down or let go of it until it is installed in the board. When you bend the leads of the IC, hold the IC in one hand and place your other hand on the work surface before you touch the IC to the work surface. This will equalize the static electricity between the work surface, you, and the IC.

3. Remove the 8087 integrated circuit from its protective packaging.

4. Refer to Figure 18-4. Before you install the 8087 in the computer, lay it down on its side as shown. Carefully roll it to bend the lower pins into alignment. Then turn the 8087 over and bend the pins on the other side in the same manner.

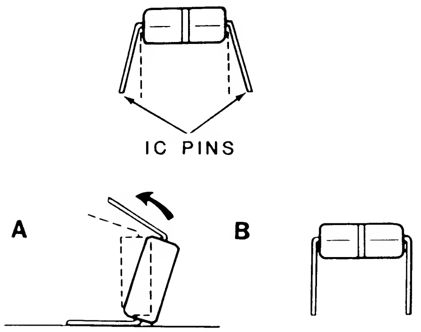

Figure 18-4. Straightening the Pins of the 8087

5. Refer to Figure 18-5 and position pin 1 of the 8087 over pin 1 of the socket. Make sure each pin is in line with and started into the socket.

6. Firmly and carefully press the integrated circuit completely into the socket.

7. Refer to Chapter 19 to reassemble the computer.

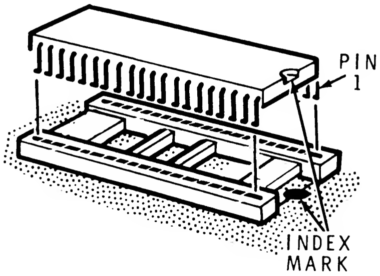

Figure 18-5. Installing the 8087

Chapter 19

Disassembly and Replacement Parts

This chapter is intended for service personnel only. The disassembly procedures are used for installing options or replacing defective parts, should the occasion arise. A parts list is provided at the end of this chapter.

Disassembly Procedures

WARNING: to avoid any shock hazard, make sure the power switch is OFF and external AC adapters are disconnected before disassembling the computer. Also, disconnect and remove all external cables, including those that connect to telephones, telephone lines, and peripherals.

Battery and Expansion Memory Compartment

Refer to Figure 19-1 and place the computer face down on a soft, clean surface. Position the computer as illustrated.

To remove the battery cover:

1. Loosen the four screws labeled A. They are held in by retainers and do not come out.
2. Rotate the back of the cover up as shown and lift the cover up away from the computer. Set the cover to one side.

To replace the cover:

1. Slide the tabs on the cover into the four holes in the front of the case.
2. Rotate the back of the cover down into place.
3. Tighten the four screws labeled A. Do not over tighten them or they will strip out.

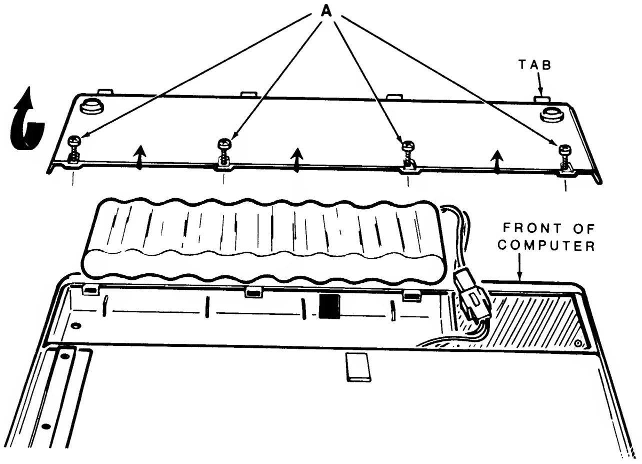

Figure 19-1. Battery Cover Removal

Battery pack

Refer to Figure 19-1 for the following steps:

1. With the top closed, place the computer face down on a soft, clean surface and remove the battery cover.
2. Lift the battery pack free of the computer.
3. Carefully disconnect the 2-wire cable connected to the battery pack from the cable that comes out of the computer.

Expansion Memory

Refer to Figure 19-2 for the following steps:

1. With the top closed, place the computer face down on a soft, clean surface and remove the battery cover.
2. Remove the screw labeled A. Set the screw and the fiber washer to one side.
3. To disconnect the memory expansion board, gently rock the board from side to side while pulling it toward the front of the computer. Set the board to one side.

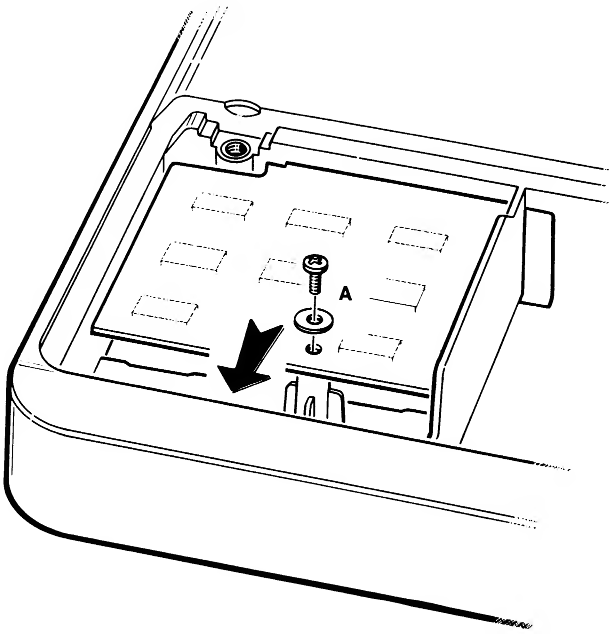

Figure 19-2. Expansion Memory Board Removal

Internal Modem

Refer to Figure 19-3 and the following steps to remove the modem cover:

1. With the top closed, place the computer face down on a soft, clean surface.
2. Remove the three screws labeled A and set them to one side.
3. Remove the modem cover by rotating it up and to the left as illustrated.

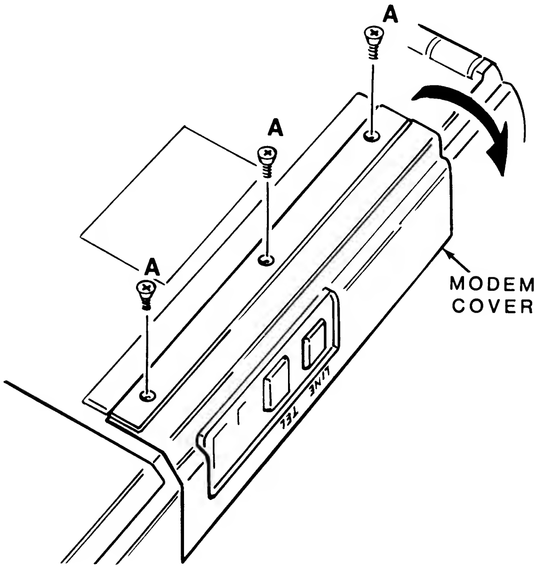

Figure 19-3. Modem Cover Removal

4. Refer to Figure 19-4 and remove the modem card by gently pulling on one side of the card and then the other, as illustrated.

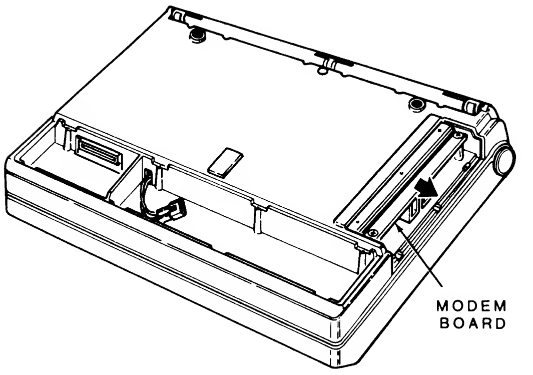

Figure 19-4. Modem Card Removal

Computer Disassembly

WARNING: Before proceeding any further, disconnect all cables and power sources from the computer. Failure to do so may result in a shock hazard.

CAUTION: Be very careful when you remove the bottom cover of the computer. There are a number of cables connected to various assemblies within the computer. If you do not disconnect them carefully, they can be damaged, making the computer inoperable.

CAUTION: Do not attempt to disassemble the computer without first removing the internal modem card, the battery, and the expansion memory board.

Bottom Cover

Refer to Figure 19-5 and the following steps to prepare the bottom cover for removal:

1. Remove the two screws labeled A and set them to one side. Remove the four screws labeled B (they are slightly larger than the screws labeled A). These six screws secure the connectors at the back of the computer.
2. Remove the four short screws labeled C and set them to one side.
3. Remove the screw labeled D and set it to one side. You may have to turn the computer over and gently shake it to make this screw fall out.

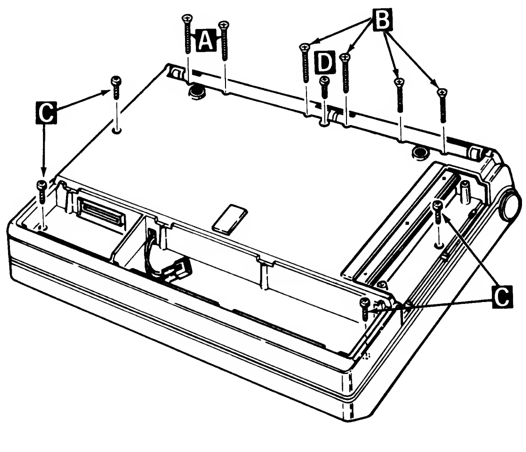

Figure 19-5. Preparing Bottom Cover for Removal

4. Refer to Figure 19-6 and remove the on/off switch button by pulling it out, away from the computer.

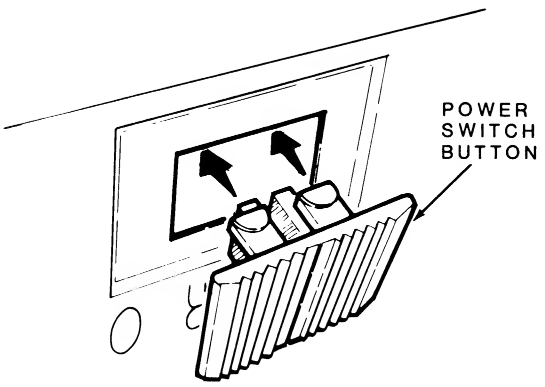

Figure 19-6. On/Off Switch Button Removal

Refer to Figure 19-7 and the following steps to remove the bottom cover:

1. Open the back door of the computer and remove the two screws labeled A and set them to one side.
2. Gently grasp the back door of the computer and lift it up. The back of the computer should be loose.
3. Slide the bottom cover of the computer toward the front until the front of the cover is free of the tab shown in the illustration. The bottom cover should lift free of the rest of the computer, except for the battery cable. Be careful not to bend the pins of the memory expansion connector.
4. Feed the battery cable through the bottom cover.
5. Set the bottom cover to one side.

To replace the cover, reverse this procedure, making sure that all cables are routed where they will not be pinched or be in the way of the modem board when it is installed.

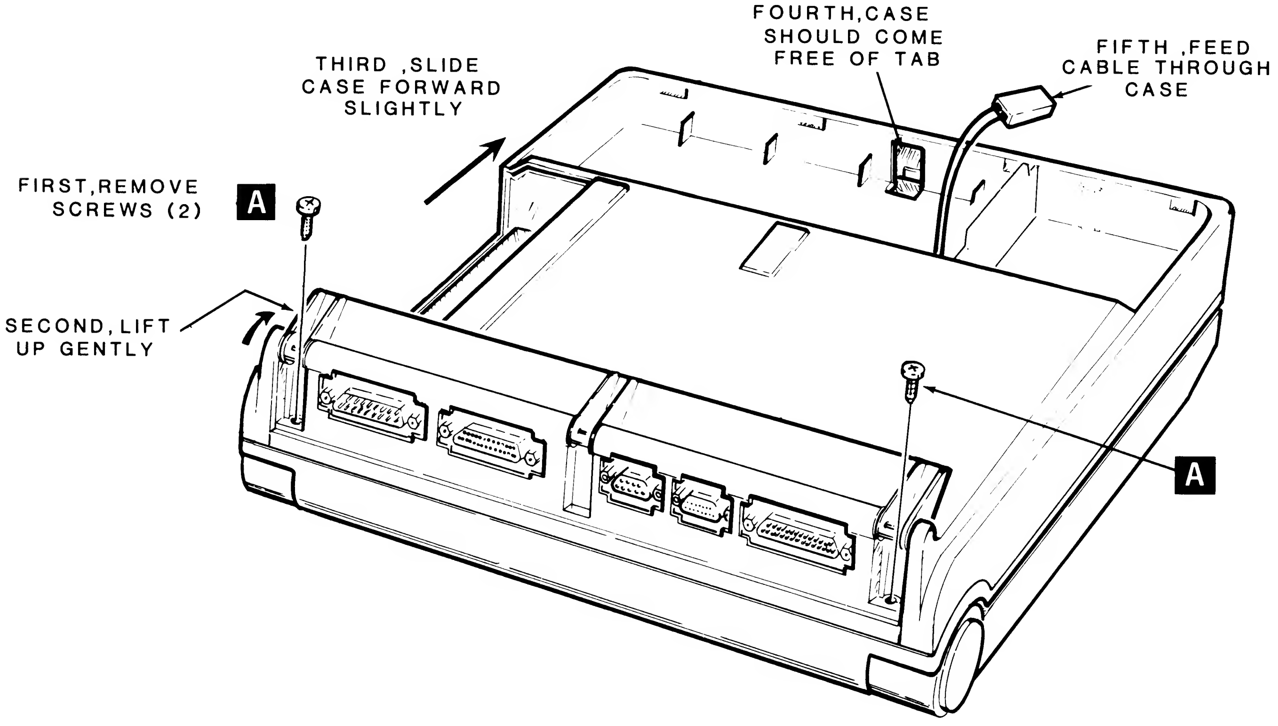

Figure 19-7. Bottom Cover Removal

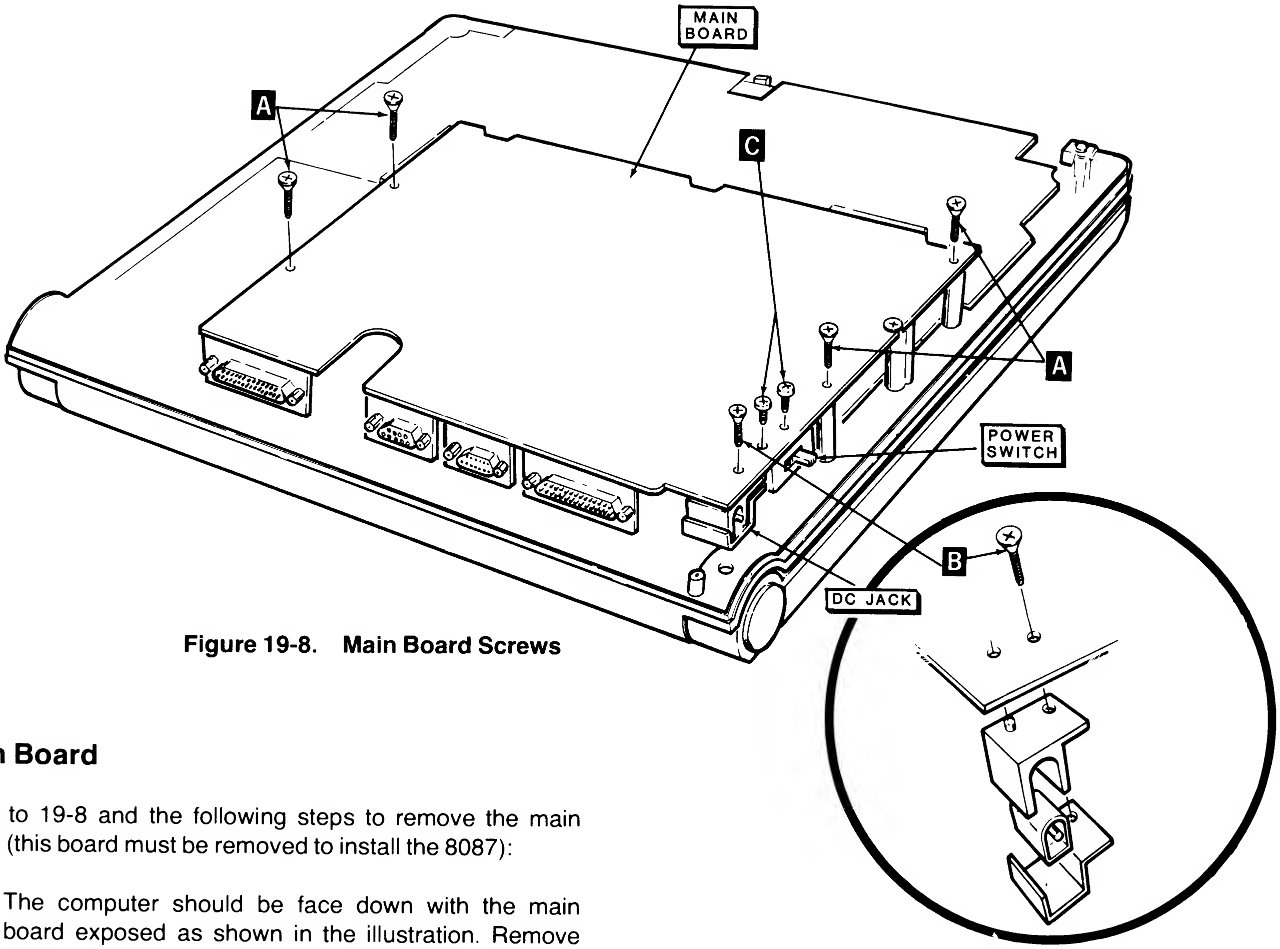

Figure 19-8. Main Board Screws

Main Board

Refer to 19-8 and the following steps to remove the main board (this board must be removed to install the 8087):

1. The computer should be face down with the main board exposed as shown in the illustration. Remove the four screws labeled A and set them to one side.

2. Remove the screw labeled B that secures the DC jack to the main board. There are three parts to this assembly and they will fall apart when the screw is removed. Examine the insert for their relationship when you reassemble the DC jack and the main board.

3. Remove the two screws and nuts labeled C that secure the power switch to the main board.

CAUTION: The main board will now be loose. However, do not attempt to lift it away from the computer. There are a number of cables that can be easily damaged.

4. Refer to Figure 19-9 and open the computer as shown. Swing the bottom of the computer open until it is standing in a vertical position.

5. Carefully move the front edge (now the top edge) of the main board away from the computer about four inches as illustrated.

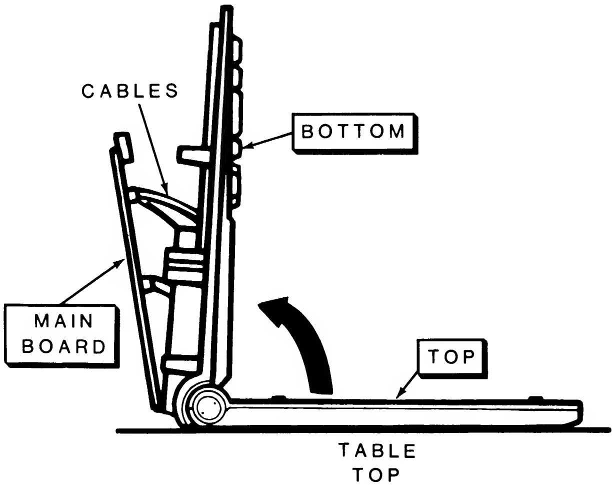

Figure 19-9. Freeing the Main Board

Refer to Figure 19-10 for the following steps.

CAUTION: The cable at CN6 is very easily damaged. Use care when you disconnect it. Do not flex or reform it.

6. Refer to the inset and open the connector by lifting the slide away from the board and then gently disconnect the ribbon cable at CN6.

7. Gently disconnect the 7-pin cable connector at CN10. Lower the main board down onto the work surface.

NOTE: At this point you can install the 8087 numeric processor extension as described in Chapter 18.

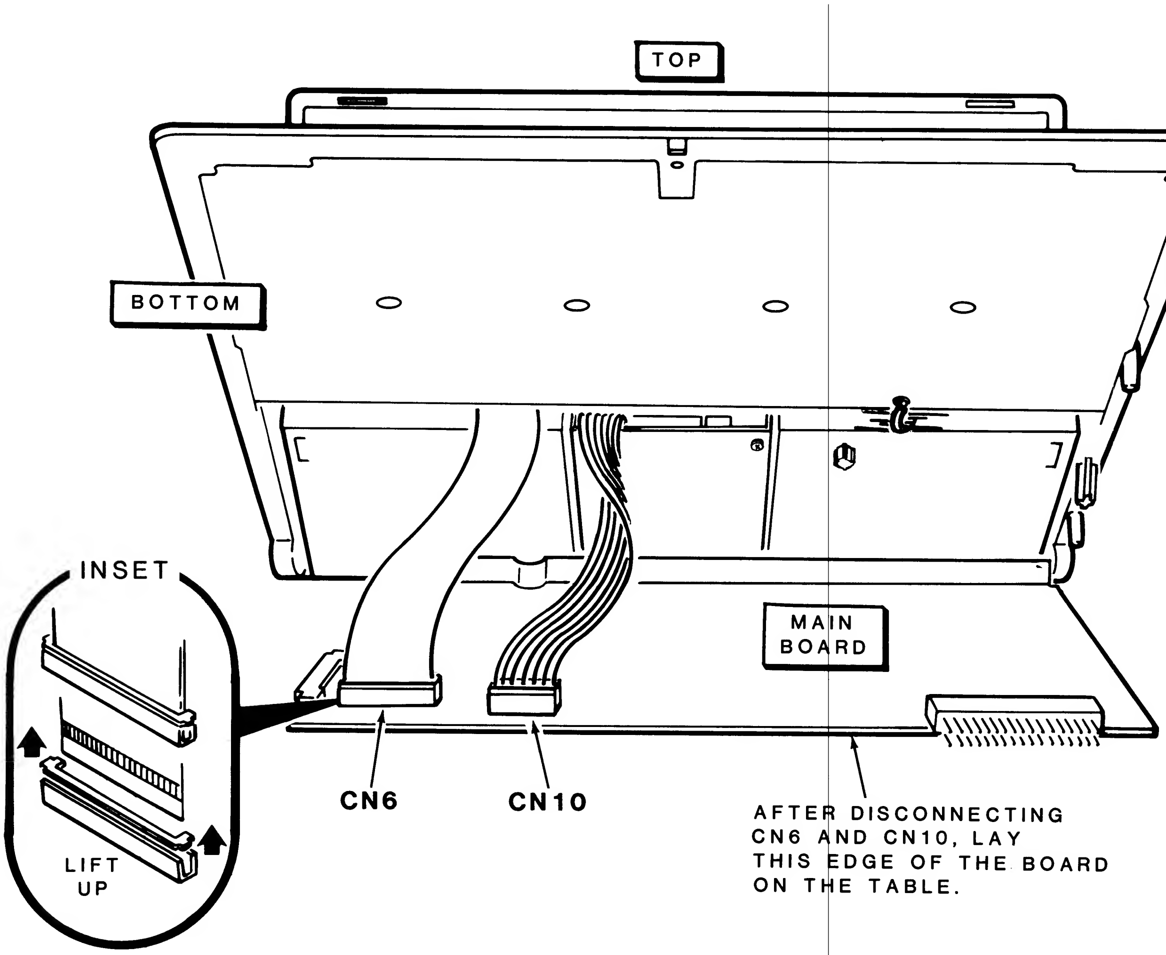
TOP
BOTTOM
INSET
LIFT UP
MAIN BOARD
CN6
CN10
AFTER DISCONNECTING CN6 AND CN10, LAY THIS EDGE OF THE BOARD ON THE TABLE.

Figure 19-10. Disconnecting CN6 and CN10

Refer to Figure 19-11 for the following steps.

CAUTION: The next two cables are very easily damaged. Use care when you disconnect them. Do not flex or reform either cable.

8. Open the connector by lifting the slide away from the board and then gently disconnect the ribbon cable at CN8.

9. Open the connector by lifting the slide away from the board and then gently disconnect the ribbon cable at CN7.

10. Gently disconnect the 4-pin cable connector at CN14.

11. Gently disconnect the 8-pin cable connector at CN9.

12. Gently disconnect the 13-pin cable connector at CN4.

13. Gently disconnect the 10-pin cable connector at CN1.

14. Set the main board to one side.

Reverse this procedure for reassembly. During reassembly, carefully position all cables where they will not be damaged. Also, do not unnecessarily flex any of the cables, particularly the ribbon cables that go to the two disk drives and the keyboard.

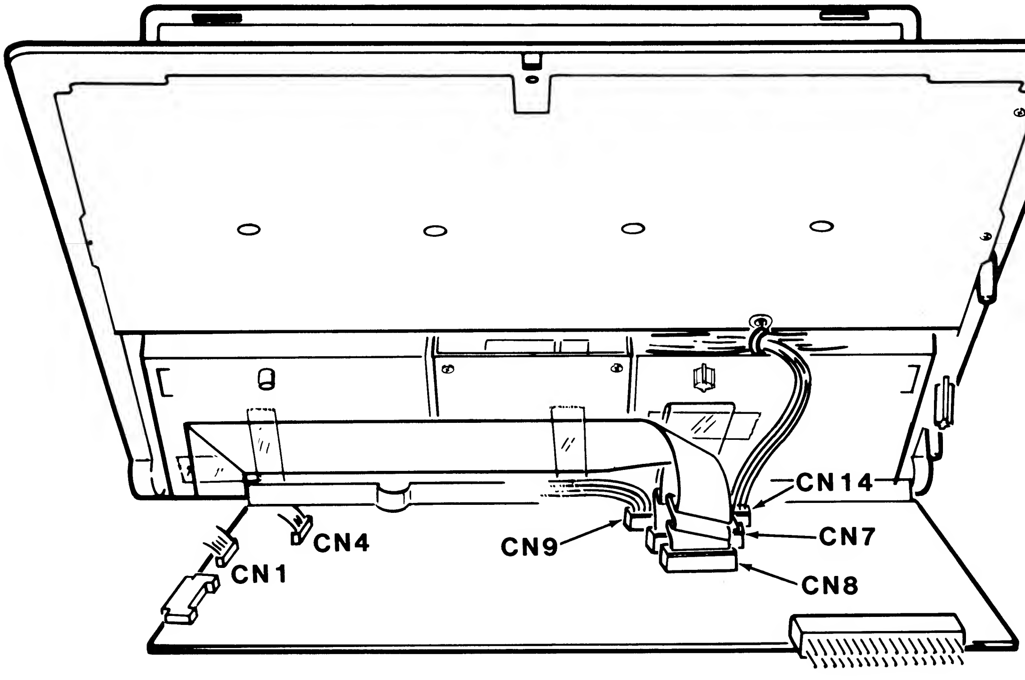

Figure 19-11. Disconnecting CN1, CN4, CN7, CN8, CN9, and CN14

Z-180 PC Series Laptop Computers Owner's Manual