Programmer's Reference Manual

Z-100 PC Series Computers

595-3158

TM-150

ZENITH data systems

Programmer's Reference Manual

Z-100 PC Series Computers

ZENITH | data systems

LIMITED RIGHTS LEGEND

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

Printed in the United States of America

Zenith Data Systems Corporation
St. Joseph, Michigan 49085

Contents

Contents

Contents

Contents

Figures

Tables

Abbreviations

AEN	Address Enable
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input Output System
CD	Carrier Detect
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CRC	Cyclic Redundancy Check
CTS	Clear to Send
DIP	Dual Inline Package
DMA	Direct Memory Access
DOS	Disk Operating System
DSR	Data Set Ready
ECC	Error Correction Code
IC	Integrated Circuit
I/O	Input/Output
LSB	Least-Significant Bit
MHz	Megahertz
MSB	Most-Significant Bit
ms	millisecond
NMI	Non-Maskable Interrupt
ns	nanosecond
PPI	Programmable Peripheral Interface
RAM	Random Access Memory
RGB	Red, Green, Blue
ROM	Read-Only Memory
VDC	Volts Direct Current

Chapter 1

Introduction and Specifications

Introduction

This manual is designed with the programmer in mind. It contains useful information about the Z-100™ PC series of IBM PC XT-compatible Zenith Data Systems Computers.

Here you will find information about the hardware in these computers and how to use the system Read-Only Memory (ROM) to gain the best performance in your machine and assembly language programs. This manual is designed to supplement the information in the Programmer's Utility Pack, a software package that contains a screen editor and the assembly language utilities associated with MS-DOS. Also, as part of this manual set, the iAPX 88 Book from Intel is included to provide you with a complete, comprehensive, description of the Intel 8088 and its instruction set.

Since most of the information here is concerned with programming the Zenith personal computers, you will notice that there is little information about the circuitry of those machines. For more information on the technical aspects of the Z-100 PC Series Computers, a comprehensive set of service manuals is available from Zenith Data Systems. These service manuals go beyond the simple servicing of your computer, including complete circuit descriptions as well as data sheet reprints of the more complex integrated circuits that are used in these Zenith personal computers.

Specifications

Computer Dimensions

Z-150	15.75 × 6.25 × 16.5 inches (40.0 × 15.9 × 41.9 cm)
Z-160	19.88 × 8.5 × 19.5 inches (50.5 × 21.6 × 49.5 cm)

Backplane 62-pin, 8-slot, .825-inch separation, IBM PC XT bus compatible

Keyboard 84 keys with extended function capabilities, 8048 keyboard processor

Keyboard Dimensions

Z-150	18.0 × 7.88 × 1.38 inches (45.7 × 20.0 × 3.5 cm)
Z-160	18.38 × 7.5 × 1.38 inches (46.7 × 19.0 × 3.5 cm)

Power Supply 168 watts
+5V @ 16.4A maximum
+12V @ 5.4A maximum
−12V @ .25A maximum
+12V regulated @ 1.5A maximum
100-130 VAC or 200-230 VAC switch-selectable input

Processors Intel 8088 or Harris 80C88 8/16-bit microprocessor
Optional Intel 8087 Numeric Data Coprocessor
4.77 MHz clock

Memory 128K Random Access Memory (RAM) minimum standard.
Expandable to 320K on first memory card with 64K devices.
User memory expandable to 640K by adding optional second memory card.
First memory card optionally expandable to 640K with 256K devices when these become readily available.

Video Standard IBM PC color graphics compatible with red, green, blue (RGB) color or composite monochrome monitors.

Text: 80 characters × 25 lines or 40 characters × 25 lines, software selectable.
Graphics: point addressable 640 × 200 pixels or 320 × 200 pixels, software selectable.
Eight colors, two intensities for RGB output or 16-level gray scale for monochrome output.

Z-160—internal monitor is 9″ amber monochrome display or optional green display.

Audio One 8-ohm, 2-inch speaker

Mass Storage One or two 5.25-inch double-sided, double-density, 40-track floppy disk drives.
MS-DOS Version 2 maximum capacity of 360K per disk.
MS-DOS Version 1.25 maximum capacity of 320K per disk.

Optional 10M capacity Winchester disk drive with up to four separate partitions.

Input/Output One parallel Centronics-compatible printer port.
One or two RS-232 serial asynchronous Input/Output (I/O) port(s).

Shipping Weight

Z-150 Computer, keyboard, two floppy disk drives: 41 pounds (19.1 kg)
Z-160 Computer, keyboard, two floppy disk drives: 55 pounds (25.0 kg)

Chapter 2

System Operating Features

Introduction

This chapter contains information concerning the built-in features of your computer with emphasis on those which will be of primary use in programming the various features and functions.

Among the items covered are the following:

- Monitor ROM
- System Memory and I/O Address Maps
- Interrupt Vectors
- Support Packages

Monitor ROM

Your computer contains very powerful programming, servicing, and testing routines that are programmed permanently into the machine's ROM. These routines may be used to manipulate and program many of the features and functions of your computer, such as accessing the user and video RAM and implementing the varied video text and graphics capabilities.

Also incorporated in the ROM code, are provisions for loading (booting) an operating system, such as MS-DOS, from any disk drive in your system. The Dual In-line Package (DIP) switches on the Central Processing Unit (CPU) card may be set to implement an autoboot at powerup or a keyboard initiated manual boot. Refer to the Configuration section in Chapter 4 for DIP switch setup instructions.

NOTE: Also available is the Disk-Based Diagnostics for the Z-100 PC Series, Model CB-5063-13, a highly comprehensive system checkout and servicing utility. Details on the use of this utility are provided with the Disk-Based Diagnostics package.

The Monitor ROM features include four types of routines:

- Power-up checks
- User-selected tests
- Interactive programming and debugging routines
- Programmable I/O facilities

Power-Up Checks

The following hardware checks are automatically made when the computer is turned on:

- CPU
- ROM
- User RAM
- Interrupt control and timer circuits
- Parity RAM
- Keyboard microprocessor
- Disk drive read
- Disk drive seek function
- Disk controller
- Disk Direct Memory Access (DMA) overrun
- Disk sector
- Disk Cyclic Redundancy Check (CRC)
- Disk address mark

Error Messages

Table 2.1 describes possible screen error messages that may occur at powerup and what you can check to correct the problem. If your action does not resolve the problem, you will need to have your computer serviced by a qualified individual.

Table 2.1. Possible Power-Up Diagnostic Messages and Explanations

```
+++ ERROR:  CPU failure! +++
+++ ERROR:  ROM checksum failure! +++
```

These two messages indicate that the CPU card may be malfunctioning. The checksum message is a result of a mismatch between a predetermined value and a value derived from the contents of system ROM. Turn the machine off, then on again. If the same message reoccurs, you will need to have your system serviced by a qualified individual.

```
+++ ERROR: RAM failure! Address: XXXX:YYYY, Bit: N, Chip: UXXX +++
+++ ERROR: Parity hardware failure! Address: XXXX:YYYY, Bit: N, Chip: UXXX +++
+++ ERROR: Parity failure! Address: XXXX:YYYY, Chip: UXXX +++
```

These messages indicate that the CPU is unable to read or write to the RAM or video RAM memory. If the chip number displayed is a 400 number, the failure is on the RAM card. If the chip number is a 300 number, the failure is on the video card. Before replacing a card, check that the card is properly seated in the backplane slot and that the CPU DIP switches are correctly set for the amount of memory installed in the computer.

```
+++ ERROR: Timer interrupt failure! +++
```

This message indicates that the timing logic on the CPU card may have failed. Make sure that the card is properly seated and set up for the options installed. Also check that all optional cards are set up correctly.

Table 2.1 (continued). Possible Power-Up Diagnostic Messages and Explanations

```
+++ ERROR: Invalid/No keyboard code received! +++
```

A message of this type indicates that the keyboard did not send the proper code at powerup to indicate proper functioning. The most likely cause is a disconnected keyboard. Check the cable (both ends in transportable systems) to make sure it is connected.

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

These messages usually occur when you are attempting to boot an operating system. The cause is usually an open drive door or not having a disk properly inserted in the system.

```
+++ DISK ERROR: Bad disk controller! +++
+++ DISK ERROR: DMA overrun error! +++
```

Errors of this nature usually indicate a malfunction on the disk controller card, but may also be caused by other defective cards in the system. If any nonstandard cards have been installed, they should be suspected first in an error condition of this nature.

```
+++ DISK ERROR: Sector not found! +++
+++ DISK ERROR: CRC error! +++
+++ DISK ERROR: Invalid address mark detected! +++
```

These errors normally indicate that an operating system was not found on the selected drive disk, or that you have a defective drive. First, try a different disk. If this error occurs often, it may be necessary to have the drive aligned or the disk controller card checked by a qualified individual.

No Error Message—Occasionally, a malfunction may occur that, by its nature, prevents anything, including an error message, from being echoed to the monitor screen. If you are using a color RGB monitor, first check the jumpers on the video card. Also check to make sure that you are allowing enough time (up to thirty seconds for Winchester systems) for any disk I/O problems to "time out".

User-Selected Tests

In addition to the automatic checks, you may select test sequences from a menu to check the following devices and parameters:

- Disk drives
- Keyboard
- Memory
- Power-Up status

These tests will produce a display that shows you the test name, number of passes made, and keyboard key sequence required to manually end the test. If an error is encountered, the test will end and an error message will be displayed.

Unlike the power-up memory test, which only tests the first and last 64K banks of RAM, the memory test selected here checks all the user RAM plus the 16 kilobytes of video screen RAM.

Selecting a Test

Tests may be performed by entering TEST at the system prompt arrow (->). The following menu will appear on the screen:

```
CHOOSE ONE OF THE FOLLOWING:

1. DISK READ TEST
2. KEYBOARD TEST
3. MEMORY TEST
4. POWER-UP TEST
5. EXIT


ENTER YOUR CHOICE:
```

Now you can select one of the desired test sequences by number. The main advantage of testing in this fashion is that these sequences will run continuously until you stop them or an error is encountered. These routines are extremely useful if a malfunction is intermittent, time dependent, or a result of heat buildup. An onscreen count is displayed to keep track of the number of times the particular test has run. The possible error messages are the same as those described under Power-Up Checks.

After a test is selected a second screen appears, with the test name at the top of the screen and the message TYPE <ESC> TO ABORT in the bottom left corner of the screen. The test count is displayed in the center of the screen. To end the test, press the **ESC** key. The count will stop and the message TYPE <ESC> TO EXIT will appear at the bottom of the screen. Press the **ESC** key to return to the test menu.

If an error ends the test, an error message appears underneath the test name and the message TYPE <ESC> TO EXIT will appear at the bottom of the screen.

Monitor ROM Routines

Another feature of your computer is the collection of monitor ROM routines. While these routines are primarily designed for use in writing, testing, and debugging machine language programs, some of the commands and features can provide an efficient means of locating hardware problems. Use of these routines assumes you have a knowledge of programming the 8088 microprocessor at the assembly level and an understanding of terminology associated with this level of programming, such as "flag," "register," "pointer," "bit," "byte," and so on.

The commands and features of this package are comprised of a substantial portion similar to the DEBUG utility supplied with MS-DOS. Among the more useful utilities included are the abilities to:

- Boot an operating system from a disk drive
- Display the contents of memory
- Execute a program
- Set program breakpoints
- Input or output values to or from I/O ports
- Search memory for a byte pattern
- Disassemble user program memory
- Change video and scrolling modes

NOTE: Because of the interrupt driven nature of the keyboard, user-written programs must incorporate commands to enable interrupts or the keyboard information will not be recognized by the microprocessor. Programs performing I/O through the ROM firmware already meet this requirement as the ROM routines automatically enable interrupts for short periods of time.

Command Summary

To use the ROM routines, the system prompt arrow (->) must be displayed. This means the Disk Operating System (DOS) must not have been booted up or you must exit the operating system to the monitor prompt by pressing the CTRL-ALT-INS keys. The following functions and commands will then be available.

NOTE: *Range* in the following statements is *address*, *address*, ¦ L*length* and *List* is *byte* ¦ *"string"*.

Command: Help

Syntax: ?

Purpose: Displays a list of commands and syntax diagrams.

Command: Boot from disk

Syntax: B[*drive type*] [*drive number*] [:*partition*]

Purpose: Reads the operating system boot code from disk and executes it. If an error is detected an error message is displayed. Drive types are F(loppy) or W(inchester) with the default determined by DIP switch settings or last booted drive type. Drive number is 0, 1, 2, or 3 for drive A, B, C, or D, respectively. Partition is separated from the drive number by a colon and is 1, 2, 3, or 4 with the default set by the PART program.

Example: BF3 *RETURN*

In this example, the system will boot the operating system from floppy disk drive number 3.

Command: Color bar

Syntax: C

Purpose: Paints the screen with sixteen different vertical color bars, including black and white (eight, if the monitor does not respond to the intensity bit). Use this display to adjust the proper range of tints on a color monitor or gray scale levels on a monochrome monitor.

Command: Display memory

Syntax: D[*range*] [L*length*]

Purpose: Displays the contents of a block or portion of memory in hexadecimal and ASCII. If the ASCII value represents a nonprinting character, an ASCII period '.' is displayed. If *length* is specified, the display will consist of number of bytes.

Example: `D3312:0,13E` *RETURN*

In this example, the screen will display the contents starting at memory address 33120H and ending at address 3325EH.

Command: Examine memory

Syntax: `E`*address*
address value.-¦ ¦*number* value.-¦ ¦*number*...

Purpose: Examines and/or changes the contents of a memory location. Displays the value of a memory byte and requests user input. If the minus key (−) is pressed, the contents of the previous memory location is shown, and if the space bar is used, the next byte is displayed. If instead of the hyphen or space a hexadecimal number from 00 to FF is entered, this value will replace the contents at *address*. Pressing the RETURN key will return to the system prompt.

Example: `E0:100` *RETURN*

This will cause the address 0000:0100 to be printed, followed by the contents of that memory address. The flashing cursor will indicate that the computer is waiting for you to enter a value to be placed in memory, to press the minus '−' key to display the previous memory location and contents, or to press the Space Bar to display the contents of the next memory location. If you press the RETURN key, the system prompt will be displayed.

Command: Fill memory

Syntax: `F`*range*`,`*list*

Purpose: Fills specified *range* of memory with data contained in *list*. The data set is recirculated until specified range is filled.

Example: `F1800:0,3FFF,05,04,03,02,01,"Ignition!"` *RETURN*

In this example, the computer will start at memory location 18000H and continuously fill the next 3FFF memory locations with information composed of numerals 5, 4, 3, 2, 1, and character string "Ignition!".

Command: Execute (Go)

Syntax: G[=*address*] [*breakpoint*]...

Purpose: Initiates program execution with up to eight breakpoints. When a breakpoint is encountered, the routine saves the processor status for the **R** and **G** commands and displays the microprocessor register contents.

Example: G=5000:3ACD,4BCD *RETURN*

In this example, the computer will attempt to execute the commands beginning at 53ACD. If location 54BCD is executed the program will stop and the contents of the user registers will be shown. Entering just G will cause the program to pick up where it left off with no breakpoints.

Command: Hexadecimal Math

Syntax: H*hexadecimal value*,*hexadecimal value*

Purpose: Performs hexadecimal arithmetic on the two hexadecimal values. Returns the sum and the difference.

Example: H43C7,99FA *RETURN*

In this example, the computer will display the following:

```
Sum: DDC1 Diff: A9CD
```

Command: Input from Port

Syntax: I*port*

Purpose: Returns the contents of the specified port.

Example: I3FE *RETURN*

In this example, the computer will return the contents of input port 3FEH.

Command: Move memory block

Syntax: M*range*,*address*

Purpose: Moves the block of memory specified by *range* to the destination, *address*. The move is implemented in a fashion which prevents overlapping moves from writing over other data.

Example: M3219:FEDC,FFFF,3905:0 *RETURN*

In this example, the computer will move 124H bytes of data from 4206CH to 39050H.

Command: Output to port

Syntax: O*port*,*value*

Purpose: Writes a value to the specified port address.

Example: 021,BC *RETURN*

In this example, the computer will write the value 0BCH to output port 21H.

Command: Examine registers

Syntax: R[*register name*]

Purpose: Examines and, optionally modifies one of the following 8088 registers: AX, BX, CX, DX, SI, DI, BP, SP, SS, CS, DS, ES, IP, PC, or FL. IP and PC both refer to the instruction pointer.

If no register is specified, all register contents are displayed.

If a single register is specified, the contents of that register are displayed and the user prompted for input. If a valid hexadecimal number is entered, the specified register's contents are changed to that value.

The FL (flags) register contents are not displayed in hexadecimal, but as a two-letter abbreviation representing the specific flag's condition as defined in Table 2.2.

Table 2.2. Flag Register Messages

FLAG	ON		OFF	
Overflow	OV	(overflow)	NV	(no overflow)
Direction	DN	(down)	UP	(up)
Interrupts	EI	(enabled)	DI	(disabled)
Sign	NG	(negative)	PL	(plus)
Zero	ZR	(zero)	NZ	(not zero)
Auxiliary Carry	AC	(carry)	NA	(no carry)
Parity	PE	(even)	PO	(odd)
Carry	CY	(carry)	NC	(no carry)

To change a given flag register, any of the abbreviations may be entered at the `->R` user-input prompt.

Example: `NG PO` ***RETURN***

This will cause the computer to set the Sign flag and reset the Parity flag. If more than one abbreviation is entered for the same flag, the flag condition will conform to the last entry.

Command: Search memory

Syntax: `S`*range*`,`*list*

Purpose: Searches the specified *range* for a string composed of *list*. Each time the string is found, the starting address of the string is displayed.

Example: `S2400:15,8CD,"Alpha",3,"Beta"` ***RETURN***

The computer will scan 8B9 bytes of memory beginning at 24015H for a sequence composed of the character string "Alpha", the numeral 3, and the character string "Beta".

Command: Trace program

Syntax: T[=*address*] [, *value*]

Purpose: Traces a user program. Single steps through the program beginning at *address*, displaying the microprocessor register contents as each instruction is executed. The *value* determines how many times to execute before returning control to the user. Default is one execution. The default address is the current address.

Example: T=011A, 10 *RETURN*

This causes ten instructions in the current program segment to be executed and register contents displayed, starting at address 11AH.

Command: Extended diagnostics

Syntax: TEST

Purpose: Provides extended diagnostics. This command results in the menu display covered under the User-Selected Tests section in this manual.

Command: Unassemble program

Syntax: U*range*

Purpose: Returns the assembly level instructions represented by the object code found in *range*.

Command: Set Video/Scroll

Syntax: V[M*video mode*] [S*scroll* mode] [100] [150]

Purpose: This command is used to set the current video mode or scroll mode. Use 100 or 150 to determine which hardware mode to use (100 signifies Z-100 bit-mapped graphics with Z-319 bit-mapped video graphics accessory, 150 signifies IBM PC-compatible graphics). Table 2.3 shows the various video modes obtainable with the VM command.

Table 2.3. Video Mode Selection

CODE	MODE
0	40 columns × 25 rows, monochrome at the RGB output.
1	40 columns × 25 rows, color at the RGB output.
2	80 columns × 25 rows, monochrome at the RGB output. Single text pages may be hardware scrolled without affecting other pages.
3	80 columns × 25 rows, color at the RGB output. Single text pages may be scrolled.
4	320 horizontal × 200 vertical pixels, color at the RGB output. Text emulation provided for 40 characters × 25 rows.
5	320 horizontal × 200 vertical pixels, monochrome at the RGB output 40 × 25 text emulation.
6	640 horizontal × 200 vertical pixels, monochrome at the RGB output. Software, hardware jump, and smooth scrolling available (see Scroll command).
7	80 × 25 text display at the monochrome output.

Example: VM6 *RETURN*

In this example, high resolution graphics (640 × 200 pixels) would be selected.

Table 2.4 indicates the scroll modes obtainable with the VS command. The scroll mode determines the manner in which information on the screen is handled once the screen is full.

Table 2.4. Scroll Mode Selection

CODE	SCROLL MODE
0	Software compatible scroll mode. Information is scrolled by actually moving screen memory contents. Also the default scrolling mode.
1	Hardware jump scrolling. Incompatible with some applications packages but faster than the software mode. May only be used when in 80 × 25 text or the graphics video modes.
2	Smooth scrolling. Usable only in the graphics modes. Moves characters upward in small segments resulting in a more pleasing and readable display.

Example: `VM6 S2` *RETURN*

In this example, the smooth scrolling mode would be selected.

Special Keys

Table 2.5 defines the keyboard keys and associated functions recognized by the ROM diagnostics. These keys permit certain editing operations and control the format of the screen display.

Table 2.5. Special Keys

KEY	FUNCTION
BACK SPACE	Press to correct typographical errors prior to command line execution.
RETURN	Executes the command line.
, or (space)	(Comma or space) Used as delimiters interchangeably between elements of a command.
SCROLL LCK or **CTRL-S**	Pause screen output until depressed a second time.
CTRL-NUM LCK	Halts computer operation until any other non-control key is pressed.
CTRL-SCROLL LCK or **CTRL-C**	Terminates any executing command and returns control to user.
CTRL-ALT-DEL	A 3-key combination which results in a system reset. System is initialized and the power-on diagnostics are performed.
CTRL-ALT-RETURN	A 3-key sequence used to interrupt execution of a program, usually during the debugging process. Keyboard activated equivalent of a program breakpoint. User may then modify or examine microprocessor registers, I/O ports or memory contents. To restart program execution at the point interrupted, the G command is entered, without *address*.
CTRL-ALT-INS	A 3-key sequence used to reboot the computer without booting from the disk, even if autoboot is enabled. This feature will allow you to boot from an alternate drive.

System Memory and I/O Address Maps

This section shows the memory layout scheme of your computer and the locations of the I/O ports. Table 2.6 is a general memory breakdown of the system.

Table 2.6. System Memory Map

TYPE OF MEMORY	HEXDECIMAL ADDRESS	DECIMAL ADDRESS
1st RAM Card	00000-4FFFF	0000000-0327679
2nd RAM Card	50000-9FFFF	0327680-0655359
Reserved	A0000-AFFFF	0655360-0720895
Monochrome Graphics	B0000-B3FFF	0720896-0737279
Reserved	B4000-B7FFF	0737280-0753663
Color Graphics	B8000-BBFFF	0753664-0770047
Reserved	BC000-BFFFF	0770048-0786431
Reserved for Bit-Mapped Video Graphics Card	C0000-EFFFF	0786432-0983039
MFM-150 Monitor Scratchpad RAM	F0000-F3FFF	0983040-0999423
Winchester Drive Buffer	F4000-F7FFF	0999424-1015807
System ROM	F8000-FFFFF	1015808-1048575

Table 2.7 indicates the I/O port addresses and the respective device assignments. For further breakdowns of the individual device I/O ports, refer to the chapter that covers the specific device.

Table 2.7. I/O Port Addresses

DEVICE	HEXADECIMAL ADDRESS RANGE
DMA Processor	000-00F
Interrupt Generator	020-021
System Timer	040-043
PPI Status Port	060-063
DMA Page Registers	080-083
Non-Maskable Interrupt (NMI) Enable Register	0A0
Diagnostic LEDs	0C0
Reserved	0C1-0CF
Reserved	0E0-0EF
Game I/O	200-20F
Reserved	278-27F
Serial Device	2F8-2FF
Parallel Printer #1	378-37A
Monochrome Monitor/ Parallel Printer #2	3BC-3BE
Color Graphics Control	3D0-3DF
Floppy Disk Controller	3F2-3F5
RS-232C Serial Interface #1	3F8-3FE

Interrupt Vectors

The interrupt vectors recognized by the microprocessor in your computer allow the different devices on the I/O bus to halt CPU operation when an operation related to the device requires service.

Table 2.8 lists the interrupt vectors which may occur as the result of various assembly level programming operations.

Table 2.8. Program Interrupt Vectors

VECTOR	FUNCTION
0	Divide by 0
1	Single Step
2	Non-Maskable Interrupt
3	Breakpoint
4	Overflow
5	Print Screen
6	Not used
7	Not used

Table 2.9 lists the interrupts recognized by the Intel 8259 Programmable Interrupt Controller.

Table 2.9. 8259 Programmable Controller Interrupts

VECTOR	FUNCTION
8	Time-of-day
9	Keyboard
A	Reserved for Z-319 card
B	Communications (COM1)
C	Communications (COM2)
D	Winchester disk drive
E	Floppy disk drive
F	Parallel printer interrupt

Table 2.10 lists the Basic Input/Output System (BIOS) entry point interrupts.

Table 2.10. BIOS Entry Point Interrupts

VECTOR	FUNCTION
10	Video I/O
11	Determine I/O configuration
12	Determine memory size
13	Disk I/O
14	Serial I/O (RS-232 I/O)
15	Not used
16	Keyboard I/O
17	Printer I/O
18	Not used
19	Boot loader (Boot operating system)
1A	Set/read time-of-day

In addition, several special vectors are recognized, as defined by Table 2.11.

Table 2.11. Special Interrupt Vectors

VECTOR	FUNCTION
1B	Keyboard break (user entry)
1C	Timer tick (user entry)
1D	Video initialization
1E	Diskette parameters
1F	Graphics characters

For further details on the interaction of interrupt vectors in programming situations, refer to the material contained in the chapter which relates to the parameter of interest. Information on parameter passing may also be found in these chapters.

Support Packages

You may also be interested in the optional plug-in cards, and the various programming languages, applications, and utility packages, available for your computer from Zenith Data Systems. These options include detailed implementation of many of the programming features of your system. Contact your Zenith Data Systems dealer for details on these packages.

Chapter 3

System Input/Output

Introduction

This chapter describes the manner in which information is passed between the different elements that make up the computer system.

The information contained in this chapter is intended as a general overview of the system only. It provides preparatory material for the chapters which follow and contains specific programming information and data for the various functions and features available in your system.

System Layout

Figure 3.1 shows the overall block diagram of the system.

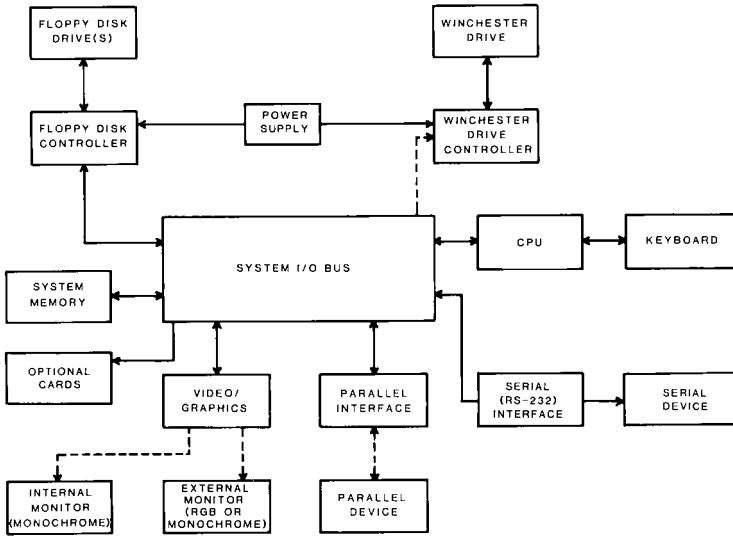

Figure 3.1. System Block Diagram

As you can see, the main artery for system communication is the I/O bus, which acts as the main trunk for information transfer. What is not readily apparent from the block diagram, is that the CPU acts as the main traffic control element for information flow.

Central Processing Unit

The CPU is really the operational heart of any computer system. The 8088 microprocessor is the main decision-making element which keeps everything happening at the right time. The 8088, the DMA controller, the program interrupt controller, and three independent 16-bit timing counters make up the CPU subsystem in your computer.

An optional 8087 coprocessor may also be installed in the system. This device will take over the higher mathematical operations ordinarily performed by the 8088 under software, freeing the 8088 to other tasks, such as controlling information flow. The main advantage of adding the coprocessor is an apparent increase in speed of the CPU when running programs which can take advantage of it.

System Bus Board

The 8-connector array on this card is basically an extension of the microprocessor I/O bus, but demultiplexed, and enhanced by the addition of interrupts and DMA functions. The bus interface is implemented via a 62-pin edge connector on the card or device which plugs into one of the eight backplane sockets. The bus supplies the following:

- Eight bits of bidirectional data
- Twenty address lines
- Six interrupt levels
- Control lines for memory and I/O read and write
- Clock and timing control lines
- Channel check line for device error reporting or parity error reporting by add-on memory
- Operating power consisting of ± 5 VDC, ± 12 VDC, and ground
- READY line for use by slow devices

System Input/Output

Connection to the bus by the cards in the system is also made through the edge connectors on the backplane board. Each of the standard cards, and any custom add-on cards, interface with the CPU through one of these connectors.

This chapter will be limited to defining the signals which appear on the electrically parallel pins of these connectors. Details on the remainder of the system can be found in the individual chapters which relate to a given function.

Table 3.1 defines the I/O bus signals.

Table 3.1. System Bus Signal Names

PIN	SIGNAL	DEFINITION
A1	I/O CHCK*	I/O channel check. Provides the CPU with parity error status for memory or other I/O devices. Active low indicates error.
A2	D7	Data bit 7
A3	D6	Data bit 6
A4	D5	Data bit 5
A5	D4	Data bit 4
A6	D3	Data bit 3
A7	D2	Data bit 2
A8	D1	Data bit 1
A9	D0	Data bit 0
A10	I/O CHRDY	I/O channel ready. Used by slower I/O devices to ensure data is not lost during read and write operations. May be held low (not ready) up to 10 CLK cycles (210 ns).
A11	AEN	Address enable. Assigns control of read and write operations to the DMA controller.
A12	A19	Address bit 19
A13	A18	Address bit 18
A14	A17	Address bit 17
A15	A16	Address bit 16
A16	A15	Address bit 15
A17	A14	Address bit 14
A18	A13	Address bit 13
A19	A12	Address bit 12
A20	A11	Address bit 11
A21	A10	Address bit 10
A22	A9	Address bit 9
A23	A8	Address bit 8

Table 3.1 (continued). System Bus Signal Names

PIN	SIGNAL	DEFINITION
A24	A7	Address bit 7
A25	A6	Address bit 6
A26	A5	Address bit 5
A27	A4	Address bit 4
A28	A3	Address bit 3
A29	A2	Address bit 2
A30	A1	Address bit 1
A31	A0	Address bit 0
B1	GND	Ground
B2	RESET	When high, resets or initializes system logic devices.
B3	+5 VDC	+5 VDC bus.
B4	IRQ2	Interrupt request 2. Not used. Available for assignment to a user-selected device.
B5	−5 VDC	−5 VDC bus.
B6	DRQ2	DMA request 2. Assigned to floppy disk controller.
B7	−12 VDC	−12 VDC bus.
B8	N.C.	No connection.
B9	+12 VDC	+12 VDC bus.
B10	GND	Ground
B11	MEMW*	Memory write. When low, causes data on data bus to be stored in memory.
B12	MEMR*	Memory read. When low, causes memory to drive data onto the data bus.
B13	IOW*	I/O write. When low, instructs an I/O device to read the data on the data bus.
B14	IOR*	I/O read. When low, instructs an I/O device to drive its data onto the data bus.
B15	DACK3*	DMA acknowledge 3. Assigned to the Winchester drive controller.
B16	DRQ3	DMA request 3. Assigned to the Winchester drive controller.
B17	DACK1*	DMA acknowledge 1. Not used. Available for user assignment.
B18	DRQ1	DMA request 1.

Table 3.1 (continued). System Bus Signal Names

PIN	SIGNAL	DEFINITION
B19	DACK0	DMA acknowledge 0. Assigned to timer #1. Initiates memory refresh cycle.
B20	CLK	System clock 4.77 MHz.
B21	IRQ7	Interrupt request 7. Assigned to parallel interface.
B22	IRQ6	Interrupt request 6. Assigned to floppy disk controller.
B23	IRQ5	Interrupt request 5. Assigned to Winchester drive controller.
B24	IRQ4	Interrupt request 4. Assigned to Serial port #1 (fixed).
B25	IRQ3	Interrupt request 3. Assigned to Serial port #2 (configurable).
B26	DACK2	DMA acknowledge 2. Assigned to floppy disk controller.
B27	T/C	Terminal count. Goes high when terminal count for any DMA channel is reached.
B28	ALE	Address latch enable. Generated by bus controller to indicate valid processor addresses to the I/O channel.
B29	+5 VDC	+5 VDC bus.
B30	OSC	Oscillator. 14.31818 MHz clock that provides the basic timing for the system.
B31	GND	Ground

Chapter 4

Central Processing Unit

Introduction

The CPU in your computer contains an 8088 microprocessor, DMA and interrupt processing circuitry, system clock, and timing and control generation circuitry.

From a programming standpoint, the CPU functions are largely accessed through the built-in ROM routines, or by means of iAPX 88 assembly level instructions. The instruction set and programming information for the 8088 microprocessor are detailed in the iAPX 88 Book included in this package, and will not be repeated here. Only the information pertinent to the unique features and functions of your machine will be addressed in this chapter.

Refer also to Chapter 2 for details on the use of the built-in ROM routines as they apply to CPU operations.

Configuration

Configuration of the hardware features and options in your system is selected by DIP switches SW1 and SW2 and by jumper block P203. P203 is a ROM size select jumper block, which is set at the factory, and should not require changing.

DIP Switches

Refer to Figure 4.1 for the location of SW1 and SW2, the two 8-section DIP switch packages. These two switches, when properly configured, will indicate to the system whether floppy disk drives are present and how many, the presence or absence of the 8087 coprocessor, memory device base size, monitor line length at powerup, user memory size, operating system autoboot active or not, and whether the monitor screen refresh frequency is 50 or 60 Hz.

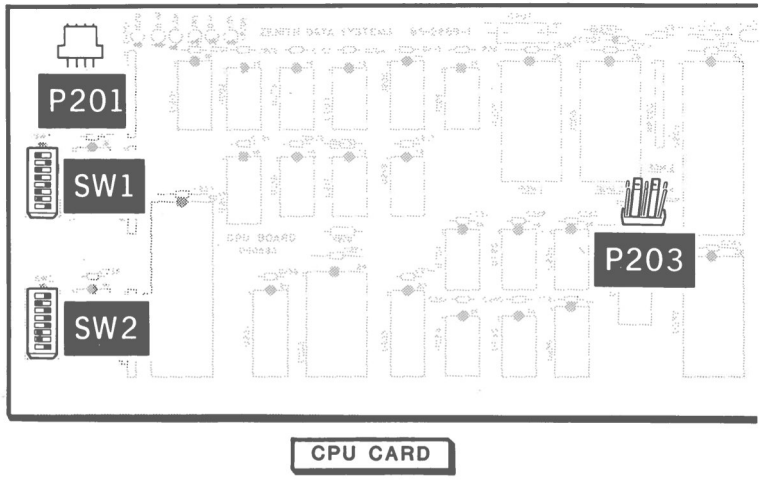

Figure 4.1. CPU Card Component Locations

Figure 4.2 shows SW1 set for a typical system configuration. Refer to Table 4.1 for the proper switch settings for your system.

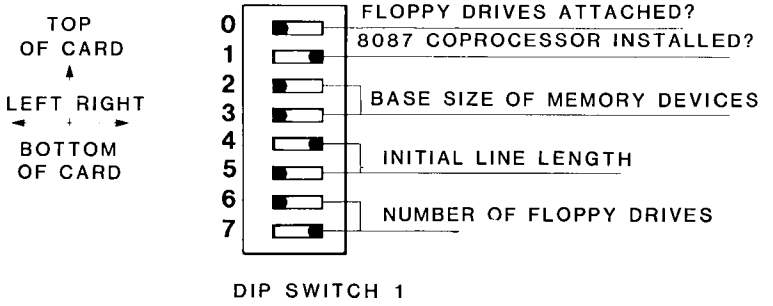

Figure 4.2. DIP Switch SW1 Settings

Table 4.1. DIP Switch SW1 Settings

FUNCTION	CONFIGURATION	SECTION AND SETTING		
		0		
Floppy Disk Drives	one or more	Left		
	none	Right		
		1		
8087 Coprocessor	installed	Left		
	not installed	Right		
		2	3	
Memory device base size	64K	Left	Left	
		other options reserved		
		4	5	
Monitor line length	80 characters	Right	Left	
	40 characters	Left	Right	
		6	7	
Number of Floppy Drives	1	Right	Right	
	2	Left	Right	
	3	Right	Left	
	4	Left	Left	

Floppy Drives Present

Section 0 of switch SW1 indicates the presence or absence of floppy disk drives attached to the computer. If set to the left, one or more drives are present; if set to the right, no floppy drives are attached.

8087 Coprocessor Installed

Section 1 of SW1 tells the system if an 8087 coprocessor has been installed. If yes, section 1 is positioned to the left; if not, to the right.

Memory Device Base Size

Sections 2 and 3 of SW1 indicate the base memory device size. 64K is the only presently acceptable option. Sections 2 and 3 are both positioned to the left; any other combinations are reserved.

Monitor Line Length

Sections 4 and 5 of SW1 define the monitor line length, in characters, when the computer is turned on. Position these sections to the desired configuration for either 40- or 80-character line length.

Number of Floppy Drives

Sections 6 and 7 indicate the number of floppy disk drives present in the system. Set these sections to the positions which reflect your floppy drive complement.

Refer to Figure 4.3 for a typical DIP switch SW2 setup. Table 4.2 defines the options obtainable with the SW2 settings.

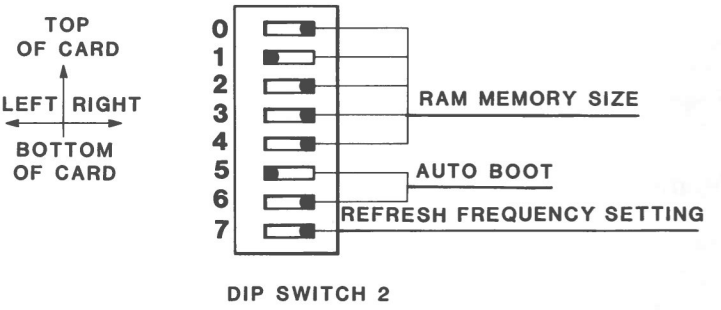

Figure 4.3. DIP Switch SW2 Settings

Table 4.2. DIP Switch SW2 Settings

FUNCTION	CONFIGURATION	POSITION AND SETTING				
		0	1	2	3	4
Memory Size	128K	Right	Left	Right	Right	Right
	192K	Right	Right	Left	Right	Right
	256K	Right	Left	Left	Right	Right
	320K	Right	Right	Right	Left	Right
	384K	Right	Left	Right	Left	Right
	448K	Right	Right	Left	Left	Right
	512K	Right	Left	Left	Left	Right
	576K	Right	Right	Right	Right	Left
	640K	Right	Left	Right	Right	Left

		5	6
Autoboot	Floppy Drive	Left	Right
	Winchester	Right	Left
	Disabled	Left	Left

		7
Monitor Screen Refresh Frequency	60 Hz	Right
	50 Hz	Left

Memory Size

The first five sections of SW2 tell the system how much user memory you have. Don't forget to change these sections if you add more memory. There is also a memory block select jumper on the memory card which must be configured if you add a second card. Details will be included with the installation instructions for the add-on card.

Autoboot from Selected Drive

Sections 5 and 6 determine if the operating system is to be autobooted from the Winchester drive or floppy drive. If neither autoboot option is selected (sections 5 and 6 are both left, or both right), autoboot is not performed, and the operating system must be booted manually.

Monitor Sync Frequency

Section 7 of SW2 tells the CPU at what vertical sync frequency the monitor should run. Select the position that corresponds to your AC power line frequency. Normally, this will be 60 Hz in North America.

Programming CPU Functions

The portion of this manual on programming the CPU operations will, of necessity, be brief. The operations are largely "housekeeping" types of routines. Also included are functions which cannot be otherwise put into the other sections.

Included here are system functions which will print screen routines, determine I/O configuration, determine memory size, and set and read the time-of-day. A method for programming rudimentary sound, at the assembly language level, is also provided. Use of these functions assumes you have a knowledge of this level of programming.

System Functions

Interrupts 5H, 11H, 12H, and 1AH are used to call a desired system function through the 8088 microprocessor.

Interrupt 5H—Print Screen Contents

When this interrupt is implemented, the contents of the screen are dumped to an attached printer. Only valid characters will be printed. Graphics containing characters not recognized by the printing device will be ignored.

As an aid when using the print screen routine, a byte of global RAM has been reserved at address 0050:0000 as a status byte. During execution of the print screen function this byte is set to a value of 1, which represents a flag value used by the routine to prevent other print screen requests from interrupting an already executing dump. Upon completion of the operation, the status byte will be changed to 0 if there were no errors, or to FFH if an error was encountered. The most common error would be a printer time-out error.

Interrupt 11H—Determine I/O Configuration

This function is useful when applications programs need to know what hardware is installed in the system. The hardware configuration is returned as bit-mapped data in 8088 register AX as defined in Table 4.3.

Table 4.3. I/O Configuration Data

BIT #	MEANING
0	Disk drives installed in system.
1	8087 coprocessor installed.
2 & 3	Amount of base RAM (device size) installed on the user memory card(s).
0 0	16K
0 1	32K
1 0	48K
1 1	64K
4 & 5	Initial video mode
0 0	Unused
0 1	40 × 25 Monochrome on the color card.
1 0	80 × 25 Monochrome on the color card (RGB output).
1 1	80 × 25 Monochrome on the monochrome card (Monochrome output).
6 & 7	Number of floppy drives—bit value plus 1 if bit 0 is set.
0 0	1 drive
0 1	2 drives
1 0	3 drives
1 1	4 drives
8	Unused
9 10 & 11	Number of RS-232 cards in the system. The standard I/O on this system emulates the IBM PC I/O, so the minimum will be two.
12	Game I/O card is installed, for using joysticks, paddle controllers, etc.
13	Unused
14 & 15	Number of printers installed.

Interrupt 12H—Determine Memory Size

When this function is called, a value representing the number of contiguous 1K blocks of user memory is returned in register AX. For example, a value of 256 would indicate that the system contains 256 kilobytes of RAM.

Interrupt 1AH—Set/Read Time-of-Day

The time set/read function, which permits reading and writing the software time variables, is actually interrupt driven by the CPU's 8253 timer/counter.

If the value in AH is 0, the current time-of-day is read and returned in CX and DX. CX contains the high part of the count, and DX the low. If the timer has overflowed to the next day, the contents of AL will NOT be 0; otherwise it will be 0.

To set the time, the desired high count is placed in the CX register, and the low count in DX. The contents of AH must NOT be 0 when the interrupt is called.

The counts mentioned are multiples of the 18.2159 Hz system clock input (1 second = 18.2159). Therefore, 1 hour would be represented by a count of approximately 65,577 (18.2159 × 3600); CX would contain the value 1, and DX the value of 41 when the time was read.

Jump Vectors

There are three jump vectors which may be of interest to those creating applications programs. They are:

F000:FFF0—	Power-up reset vector.
F000:FFED—	Jump unconditionally to the MFM-150 monitor/debugger.
F000:FFEA—	Set machine mode if AL= 0FFH, Z-100 mode will be entered using the Z-319 Bit-Mapped Video Graphics Card. If AL=0, Z-150 mode will be entered using the Z-309 Video Card.

Sound Programming

Sound may be programmed in one of three different ways:

- Pulse train generated by toggling a program control register bit
- The output of Channel 2 of the timer/counter programmed to deliver a sound waveform to the speaker
- Clock input to the timer/counter may be modulated by a program controlled I/O register bit

The three methods may be implemented simultaneously.

Chapter 5

Keyboard

Introduction

The keyboard entry point permits an application program to determine the status of the keyboard or to receive characters entered through the keyboard.

Interrupt 16H—Keyboard Input/Output

Three operation codes are used with this interrupt to determine the function to be performed.

0—Get a character from the keyboard
1—Determine if a key code is waiting in the keyboard's buffer
2—Return the keyboard's shift status

To use this interrupt, load the desired operation code in register AH and then execute an INT 16H instruction. Table 5.1 describes the results for each operation code.

Table 5.1. Responses to Interrupt 16H

OPERATION CODE	CONDITION	RESPONSE
0	Character Entered	ASCII code will be in register AL. Scan code will be in register AH. Character is removed from the keyboard buffer.
	Character Not Entered	Processing will halt until a key is pressed. The response will then be as when a key was entered.
1	Character Entered	The 0 flag will be false. AL will contain the ASCII code. AH will contain the scan code. Character will remain in the keyboard buffer.
	Character Not Entered	The 0 flag will be true.
2		Returns the status of the special function keys in register AL. Register is bit-mapped with bits that are set indicating which keys are pressed or which modes are active.
	Right SHIFT key	Bit 0 set
	Left SHIFT key	Bit 1 set
	CTRL key	Bit 2 set
	ALT key	Bit 3 set
	SCROLL LCK active	Bit 4 set
	NUM LCK active	Bit 5 set
	CAPS LOCK active	Bit 6 set
	INS active	Bit 7 set

Keyboard Configuration

This section repeats much of the information found in the *User's Guide* and *Operations Manual.* It describes the various groups of the keyboard, how they are typically used in software applications, and the codes produced by each key under various conditions.

Figure 5.1 identifies the twenty six alphabetic keys, the CAPS LOCK, and the two SHIFT keys. The keys are arranged following the American standard typewriter arrangement, and the two SHIFT keys and CAPS LOCK key function in a manner similar to a typewriter. The exception is that the CAPS LOCK key affects *only* the alphabetic keys. All other keys are unaffected by its action.

A small red light in the key indicates the function of the CAPS LOCK key. If the light is on, the function is on and alphabetic keys are entered as uppercase. If the light is off, the alphabetic keys are received as lowercase unless one or both of the two SHIFT keys are down at the time of a key entry.

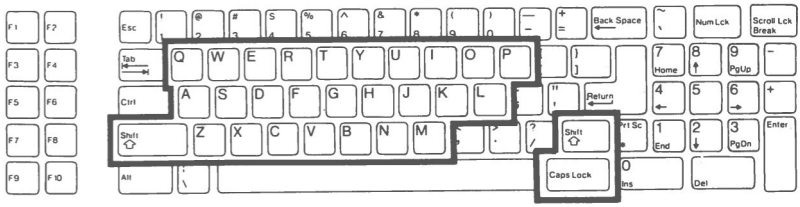

Figure 5.1. Alphabetic Keys

In addition to the codes generated by the keys themselves or with the CAPS LOCK or SHIFT key(s) pressed, alternate codes can be generated from these twenty six keys by holding down the control (CTRL) or alternate (ALT) key.

The scan codes generated from the alphabetic keys are listed in Table 5.2. The least-significant byte of the scan code is the value returned as the ASCII code.

Table 5.2. Alphabetic Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	CAPS LOCK ON
A	1E61H	1E41H	1E01H	1E00H	1E41H
B	3062H	3042H	3002H	3000H	3042H
C	2E63H	3E43H	2E03H	2E00H	2E43H
D	2064H	2044H	2004H	2000H	2044H
E	1265H	1245H	1205H	1200H	1245H
F	2166H	2146H	2106H	2100H	2146H
G	2267H	2247H	2207H	2200H	2247H
H	2368H	2348H	2308H	2300H	2348H
I	1769H	1749H	1709H	1700H	1749H
J	246AH	244AH	240AH	2400H	244AH
K	256BH	254BH	250BH	2500H	254BH
L	266CH	264CH	260CH	2600H	264CH
M	326DH	324DH	320DH	3200H	324DH
N	316EH	314EH	310EH	3100H	314EH
O	186FH	184FH	180FH	1800H	184FH
P	1970H	1950H	1910H	1900H	1950H
Q	1071H	1051H	1011H	1000H	1051H
R	1372H	1352H	1312H	1300H	1352H
S	1F73H	1F53H	1F13H	1F00H	1F53H
T	1474H	1454H	1414H	1400H	1454H
U	1675H	1655H	1615H	1600H	1655H
V	2F76H	2F56H	2F16H	2F00H	2F56H
W	1177H	1157H	1117H	1100H	1157H
X	2D78H	2D58H	2D18H	2D00H	2D58H
Y	1579H	1559H	1519H	1500H	1559H
Z	2C7AH	2C5AH	2C1AH	2C00H	2C5AH

Keyboard

The nonalphabetic keys, identified in Figure 5.2, include the numbers 0 through 9, the common punctuation marks, and the special programming characters that make up the remainder of the printing ASCII character set. Since each of these keys has one character, 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 5.3, 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 with the CTRL or ALT key down. 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.

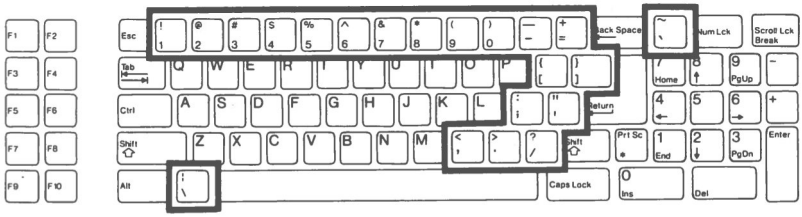

Figure 5.2. Nonalphabetic Keys

Table 5.3. Nonalphabetic Key Scan Codes

KEY TOP	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE
! 1	0231H	0221H		7800H
@ 2	0332H	0340H	0300H	7900H
# 3	0433H	0423H		7A00H
$ 4	0534H	0524H		7B00H
% 5	0635H	0625H		7C00H
^ 6	0736H	075EH	071EH	7D00H
& 7	0837H	0826H		7E00H
* 8	0938H	092AH		7F00H
(9	0A39H	0A28H		8000H
) 0	0B30H	0B29H		8100H
_ -	0C2DH	0C5FH	0C1FH	8200H
+ =	0D3DH	0D2BH		8300H
~ `	2960H	297EH		
{ [	1A5BH	1A7BH	1A1BH	
}]	1B5DH	1B7DH	1B1DH	
: ;	273BH	273AH		

Table 5.3 (continued). Nonalphabetic Key Scan Codes

KEY TOP	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE
" '	2827H	2822H		
< ,	332CH	333CH		
> .	342EH	343EH		
? /	352FH	353FH		
| \	2B5CH	2B7CH	2B1CH	

The common control keys are shown in Figure 5.3; the scan codes and the usual software function are listed in Table 5.4.

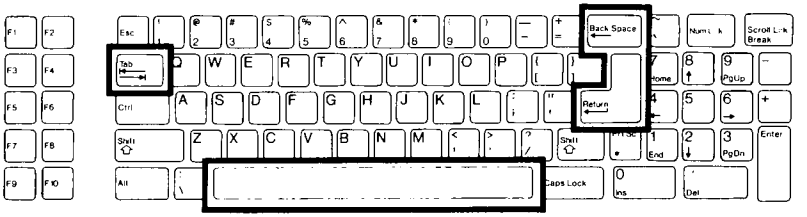

Figure 5.3. Common Control Keys

Table 5.4. Common Control Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	USUAL FUNCTION
BACK SPACE	0E08H	0E08H	0E7FH		Moves the cursor one position to the left. Often used to erase the character.
TAB	0F09H	0F00H			Moves the cursor to next tab column or to previous tab column with SHIFT key pressed.
RETURN	1C0DH	1C0DH	1C0AH		Returns cursor to 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).

The special function keys are identified in Figure 5.4. The scan codes and usual function (if any) are listed in Table 5.5. Note that the keypad special functions are shown in this figure and table, along with the codes for the numeric function, although they are not specifically called out in the Usual Function column.

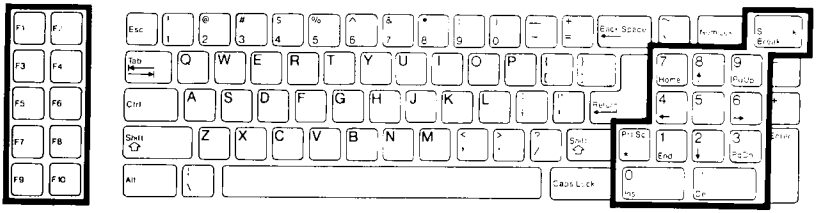

Figure 5.4. Special Function Keys

Table 5.5. Special Function Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	USUAL FUNCTION
F1	3B00H	5400H	5E00H	6800H	Function Key 1
F2	3C00H	5500H	5F00H	6900H	Function Key 2
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	4400H	5D00H	6700H	7100H	Function Key 10
SCROLL LCK BREAK			0000H	see function	CTRL-BREAK used to terminate a program. ALT-BREAK empties the keyboard buffer.
7 HOME	4700H	4737H	7700H	6667H	Moves the cursor to the home position—upper left-hand corner of the screen. Enters the 7 key when used with the ALT key.

Table 5.5. (continued) Special Function Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	USUAL FUNCTION
8 ↑	4800H	4838H		8	Moves the cursor up one line. Enters the number 8 when used with the ALT key.
9 PGUP	4900H	4939H	8400H	9	Moves to the previous page (word processing). Enters the number 9 when used with the ALT key.
4 ←	4B00H	4B34H	7300H	4	Moves the cursor one position to the left. Enters the number 4 when used with the ALT key.
5	n/a	4C35H		5	Enters the number 5 with the ALT key.

Table 5.5. (continued) Special Function Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	USUAL FUNCTION
6 →	4D00H	4D36H	7400H	6	Moves the cursor one position to the right. Enters the number 6 when used with the ALT key.
PRTSC *	372AH	see function	7200H		Enters the asterisk or when used in conjuction with either SHIFT key, sends the contents of the screen to the printer port. This last function is part of the monitor ROM and cannot be overridden.
1 END	4F00H	4F31H	7500H	1	Moves to the end of a file (word processing). Enters the number 1 when used with the ALT key.

Table 5.5. (continued) Special Function Key Scan Codes

KEY	NOT SHIFTED	(SHIFT) SHIFTED	(CTRL) CONTROL	(ALT) ALTERNATE	USUAL FUNCTION
2 ↓	5000H	5032H	n/a	2	Moves the cursor down one line. Enters the number 2 when used with the ALT key.
3 PGDN	5100H	5133H	7600H	3	Moves to the next page (word processing). Enters the number 3 when used with the ALT key.
0 INS	5200H	5230H		0	Enters the insert mode (word processing). Used to enter the number 0 when used with the ALT key.
DEL	5300H	532EH	n/a		Used to delete characters (word processing).

Keyboard

The control keys are identified in Figure 5.5. The ALT and CTRL keys do not generate a code but modify the codes of the other keys.

The ESC key generates ASCII code 1BH (Scan code 011BH). The ESC function is often used by software to stop the execution of a program or, when used in sequence with another key, as a means of entering escape codes required by some software.

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 and then enter the decimal equivalent of the hexadecimal code you wish to generate. Then release the ALT key. When the ALT key is released, the conversion takes place and the hexadecimal code is generated.

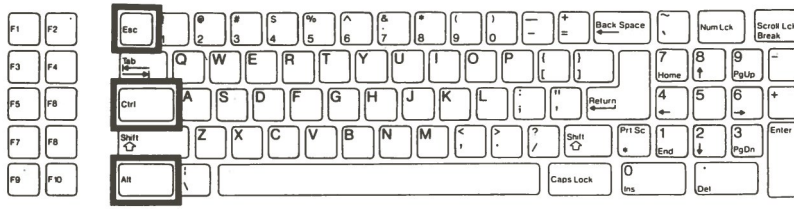

Figure 5.5. Control Keys

Figure 5.6 identifies the special keys that are used with the numeric keypad.

The NUM LCK key is used to "lock" the keypad into the shifted position. When so locked, the SHIFT key produces the "not shifted" codes in Table 5.4 for the 0 through the 9 and '.' keys of the numeric keypad. This allows the operator to use the keypad in a manner similar to a 10-key calculator. However, through software, it would be possible to use the keypad for other purposes.

The minus key (−) produces the scan code 4A2DH and the plus key (+) produces 4E2BH. The ENTER key produces the same scan codes as the RETURN key (1C0DH in the shifted and unshifted modes and 1C0AH when used with the CTRL key) and is usually used in the same way.

Figure 5.6. Keypad Special Keys

ESC ØIIBH

keypad

− 4A2D

+ 4E2B

Increasing Keyboard Buffer Size

You may increase the keyboard buffer size if you find fifteen characters are insufficient for your applications. To set a buffer size to some value other than these fifteen characters, first determine the size of the buffer that you wish to establish, then use the following procedure:

1. Set the word variable at F000:C8 to the segment address for the proper block of memory.
2. Set the word variable at 40:1A, 40:1C, and F000:CA to the starting offset of the new keyboard buffer.
3. Set the word variable at F000:CC to the offset of the last byte in the keyboard buffer.

NOTE: You cannot change the buffer size through the commands in the monitor ROM. Several of the instructions must be carried out before any input from the keyboard will be properly recognized. Therefore, you must use an assembler, such as the Microsoft Macro Assembler, available as part of the Programmer's Utility Pack from Zenith Data Systems.

The following assembly language example sets put aside the top 8K of unused Z-100 PC monitor RAM as a 4K character keyboard buffer.

```
MONITOR_SEGMENT SEGMENT AT (0F000H)     ;Z-100 PC monitor data segment
KEY_BUFF_SEGMENT    EQU  0C8H           ;Buffer segment
KEY_BUFF_START      EQU  0CAH           ;Start offset of buffer
KEY_BUFF_END        EQU  0CCH           ;End offset of buffer
FREE_MEM            EQU  2000H          ;Start of available memory
END_MEM             EQU  3FFFH          ;End of free memory
MONITOR_SEGMENT ENDS

DATA_SEGMENT SEGMENT AT (40H)           ;Compatible data segment
KEY_HEAD_PTR        EQU  1AH            ;Buffer head pointer
KEY_TAIL_PTR        EQU  1CH            ;Buffer tail pointer
DATA_SEGMENT ENDS

CODE SEGMENT PUBLIC

    ASSUME    CS:CODE,  DS:DATA_SEGMENT,    ES:MONITOR_SEGMENT

        ORG  100H                                   ;Start of  COM program
BGKBF:  MOV  AX,MONITOR_SEGMENT                     ;Point to monitor segment
        MOV  ES,AX
        MOV  AX,DATA_SEGMENT                        ;Point to data segment
        MOV  DS,AX
        MOV  ES:KEY_BUFF_SEGMENT,MONITOR_SEGMENT    ;Set buffer segment
        MOV  DS:KEY_HEAD_PTR,FREE_MEM               ;Set buffer head ptr
        MOV  DS:KEY_TAIL_PTR,FREE_MEM               ;Set buffer tail ptr
        MOV  ES:KEY_BUFF_START,FREE_MEM             ;Set start of buffer
        MOV  ES:KEY_BUFF_END,END_MEM                ;Set end of buffer
        RET                                         ;Done! -- exit.

CODE ENDS

        END BGKBF
```

To compile the above code using the Macro assembler, enter the following commands:

```
MASM BIGKBUFF;
LINK BIGKBUFF;
EXE2BIN BIGKBUFF.EXE .COM
```

The MASM file is only on the Programmer's Utility Pack distribution disk. The LINK and EXE2BIN files are both on the second distribution disk of MS-DOS. You should not see any errors in the MASM sequence or the EXE2BIN operation. You will see an error in the link sequence (missing stack segment), which is normal.

Chapter 6

System Memory

Introduction

This chapter contains information and descriptive material related to programming operations as they apply to the system memory (RAM) banks.

Refer to the information in Chapter 2 under ROM Diagnostics which applies to memory access and manipulation, specifically the Enter, Go, Move, Search, and Unassemble commands.

General

The memory cards contain the main user-accessible dynamic RAM and associated timing and multiplexing circuitry. These cards interface with the system through one of the parallel I/O connectors located on the backplane board. Connection with the cards is implemented through the use of a 64-pin edge connector which is an integral part of the cards. A configuration jumper on these cards enables up to two blocks (two cards) of memory to be designated as residing in the system.

Specifications

Your computer comes with a minimum of 128 kilobytes of RAM but may contain up to 640K if additional banks and a second memory card have been installed. One bank of memory consists of nine dynamic RAM integrated circuits. Eight of the chips store data, and one chip in each bank is used for a parity check of stored data.

As presently configured, memory may be added in increments of one bank of 64 kilobit chips (64 kilobits times 8 chips equals 64 kilobytes per bank). This allows 320 kilobytes to reside on one memory card if all five banks are installed, and an additional 320 kilobytes may reside on a second card.

When 256 kilobit RAM ICs become readily available, it will be possible to install the entire 640 kilobytes on one memory card. The system will accept this configuration only if U455, the memory address decoder, is replaced. This memory address decoder provides the address logic required by the higher capacity chips. The replacement U455 device will be made available when it is determined that reliable, economical, 256K devices are feasible.

The memory configuration is an 8-bit data structure, with odd parity generation and error detection, to detect even-numbered bit producing errors.

The RAM ICs feature on-chip refresh which is initiated when a given address is read. The card also provides the necessary circuitry to control wholesale refresh at required intervals, to arbitrate read, write, or refresh requests, and to properly time the various memory access operations.

The memory cards are designed to be completely compatible with the IBM PC and related software and firmware.

DIP Switches

DIP switches located on the CPU card are configured to reflect the amount of memory residing in your system. Refer to Chapter 4 for the correct settings of these switches for your system.

Configuration Jumpers

J401 is a configuration jumper located on the memory card used as a card select jumper. If only one memory card is installed, J401 should be positioned to the left (away from the edge connector) marked '1st'. If the system has two memory cards, position the jumper on the card to be designated as the first block of 320K memory, to '1st', and the jumper on the card containing the second block to the right marked '2nd'. If 256 kilobit RAM chips are installed, only one card is required, and J401 should be positioned to '1st'.

Refer to Figure 6.1 for the location of jumper J401.

J402 and J403 allow you to select the port address of the parallel interface. Refer to the Parallel Port Selection section in Chapter 8 for information pertaining to these jumpers.

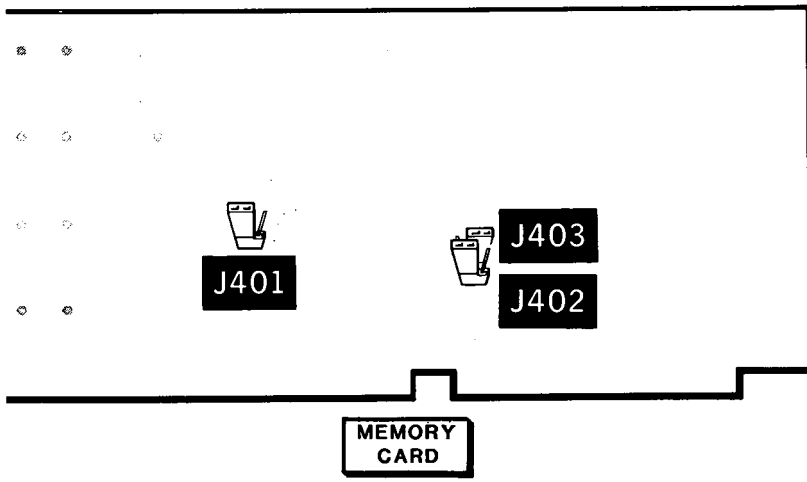

Figure 6.1. Memory Card Select Jumper Locations

Theory of Operation

In order to clarify exactly how memory is accessed and manipulated, the following description of memory circuit operation is provided.

Refer to the memory card block diagram in Figure 6.2 for the general functional description of the memory circuits which follows.

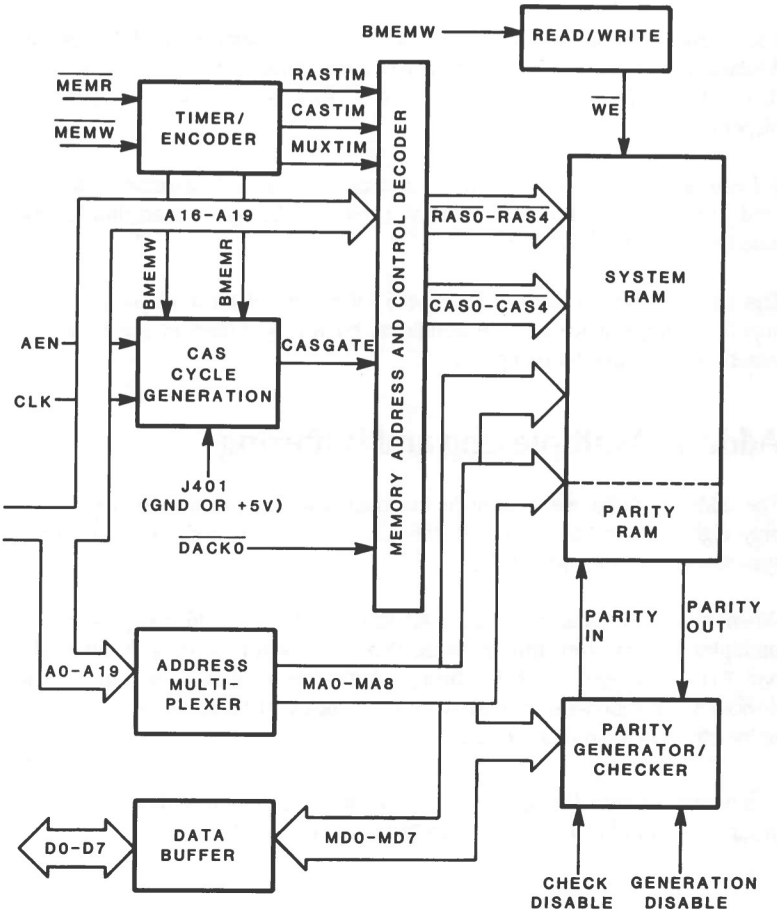

Figure 6.2. Memory Access Block Diagram

Memory Access Operations

A memory access cycle may be initiated by one of three requests: memory read, memory write, or refresh. The Address Enable Signal (AEN) must also be active for any of these requests to initiate access.

Contention between these requests, timing, and row and column address selection, is handled by the address logic. This circuitry also determines which memory card is being addressed if more than one card resides in the system.

The memory banks consist of nine 6665 or 4164 dynamic RAM ICs per bank. A total of five banks may reside on each of two memory cards. Each bit-addressable IC can hold 64K by 1 bit, for a total of 64 kilobytes per bank, plus one parity bit per byte.

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.

The parity RAMs are also 6665 or 4164 ICs. One of these chips in each bank supplies storage space for one additional bit which is used to check parity. The system is configured for odd parity.

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, A8 through A15 and A17 are present as MA0 through MA8. 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 or 4164 ICs, MA8 is only used when 256 kilobit chips are installed.

The heart of the memory card is the memory itself. The supporting circuitry serves to time and control access to a given memory address, direct the transfer of data, enable the parity generation, and check circuitry on the card containing the memory block being accessed.

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 was 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. If the address was to be read from, the ninth chip would contribute a parity bit as an error check of that location. The parity bit is the least-significant bit (LSB) in the byte.

To understand how memory is accessed, you must first understand how the memory chip itself is structured.

Each dynamic RAM IC is capable of storing 65,536 (64K) 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 (MSB) are retrieved to obtain the full 16-bit (or 18-bit for 256K chips) address word.

The low order bits also include A16, and the high order bits, A17, if 256K chips are installed.

From a functional standpoint, each memory chip can be considered a 256 by 256 bit matrix. The address byte is gated by the row address signal selecting the row and the column address signal selecting the column. During a read cycle, when all the bits from the same rows and columns of the eight data chips in a bank are combined, one data byte is formed.

Table 6.1 graphically illustrates the bank select logic.

Table 6.1. RAM Bank Select Logic

BANK	FIRST	A19	A18	A17	A16	RASTIM	MEMORY RANGE
First Block (Card 1)							
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
Second Block (Card 2)							
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

The card select jumper, J401, determines the logic state of the FIRST signal on the memory card, which determines whether the memory card is the first or second card in the system.

Data Buffer

The memory cards 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.

Parity Generation and Error Detection

When a write operation is being processed, the parity generator looks at the 8-bit data word and determines if an odd number of logical ones (1s) are present or not. If not, the parity generator places a logical 1 in the corresponding memory location of the parity RAM chip of the bank where that piece of data resides.

Upon a read retrieval, the 9-bit word residing at that address is again checked for odd parity. If a data loss which produces an even number of bits (such as an error condition) occurs, the parity generator outputs an error signal to interrupt the processor.

Parity Check/Generation Disabling

To determine the condition of the parity check circuitry, both parity generation and parity error checking can be disabled by software means by sending appropriate values to the parity disable port at 100H.

System Memory Map

Table 6.2 shows the location of user memory in the system memory scheme.

Table 6.2. System Memory Map

TYPE OF MEMORY	HEXADECIMAL ADDRESS	DECIMAL ADDRESS
1st RAM Card	00000-4FFFF	0000000-0327679
2nd RAM Card	50000-9FFFF	0327680-0655359
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 bit-mapped graphics video interface	C0000-EFFFF	0786432-0983039
MFM-150 Monitor Scratchpad RAM	F0000-F3FFF	0983040-0999423
Winchester drive buffer	F4000-F7FFF	0999424-1015807
System ROM	F8000-FFFFF	1015808-1048575

Memory addresses can range from 0 to 9FFFFH (0 to 655359 decimal), although your system may not contain that much memory. For instance, if you have a basic system, the addressable range for your system is 0 to 1FFFFH (0 to 131071 decimal) or 128K.

If you have one complete memory card filled, the range is 0 to 4FFFF. To have the complete range of memory, you must either use two full memory cards, or install 256K chips on a single card.

Programming User Memory

In general, programming user memory is very simple and straightforward. Most of the access operations can be performed directly from the built-in ROM routines or from assembly level programs written using the Programmer's Utility Pack.

Typical memory access operations include:

- Reading and writing to specific memory locations
- Allocating specific sections of memory to dedicated purposes, such as often used programming routines or user-defined character storage
- Storing repetitive character and numeric strings
- Rerouting useful interrupt vectors
- Stack operations
- Moving memory contents from one area to another
- Using RAM contents to control video graphics

In addition to normal RAM manipulation, you have the ability to disable parity generation and/or checking, either for diagnostic checks of the system, or any other purpose you might have which does not require, or for which you do not desire, parity checks to be made.

Memory Address Format

The notation recognized by the 8088 microprocessor for specific memory location access is a two-part parameter consisting of a 4-digit hexadecimal number called the segment, and a 4-digit hexadecimal number called the offset. The format for this value is:

XXXX:YYYY

The first four digits actually represent a 5-digit hexadecimal number, 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 44F01 hexadecimal (282369 decimal), since 3F3F shifted left equals 3F3F0 (259056 decimal) plus 5B11 (23313 decimal) equals 44F01H.

Since the least-significant digit of the segment is always 0, it should be apparent that more than one combination of segment and offset can point to the same memory address. For instance, the address arrived at in our previous example could also be defined by:

3F00:5F01

The highest and lowest usable segment value are determined by the memory address in question. For our 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.

Disabling the Parity Circuits

To disable the parity check and/or generation circuitry, send a value between 1 and 3 to the parity check and generation disable port at 100H. The disable port responds to values defined in Table 6.3.

Table 6.3. Parity Disable Output Port Values

VALUE	RESULT
0	Reset to normal (no disable)
1	Disable parity check
2	Disable parity generation
3	Disable both parity generation and checking

Chapter 7

Video Graphics Programming

Introduction

This chapter provides hardware and software programming information for the video card in your Z-100 PC Series Computer.

Among the features covered are:

- Character font and monitor synchronization selection
- Text mode programming
- Medium and high resolution graphics
- Foreground, background and border color selection
- General light pen interfacing information
- Video page selection
- Custom character set creation and use
- Scroll mode selection
- Cursor type and position selection
- Use of the monitor as a simple terminal
- Retrieval of screen memory information

The video card provides an interface between your computer and various display output devices such as an internal monochrome monitor, an external monochrome monitor, and/or an external RGB color monitor. An optional light pen also may be interfaced through this card. The external monochrome monitor is connected to an RCA phono jack, while the RGB, or color output, is connected to a 9-pin "D" type connector. These are both located on the video card, and project through the rear of the computer cabinet. The optional light pen interfaces through a 6-pin connector strip, P301, located on the video card.

The following are features, functions, and capabilities of the video card:

- Addressing, reading, and writing to the 16K video RAM with no timing restrictions or display interference
- Foreground, background, and border color selection
- Text mode or pixel addressable graphics mode selection
- Selection of one of four built-in 256-character ROM fonts
- Up to 128 user-defined characters
- 7 or 14 MHz video bandwidth, mode dependent
- Monochrome and/or color (RGB) output
- Internal and external vertical, horizontal and/or composite sync selection
- Light pen interface
- Cursor type selection (block, line, blinking, etc.)
- Cursor position selection

Configuration Jumpers

This section describes the optional functions obtainable through the positioning of the configuration jumpers located on the video card.

When using the following instructions, if you need to access and/or remove the video card, refer to the service manual for disassembly, card removal, and card replacement procedures.

CAUTION: This product contains Electrostatic-Sensitive Devices (ESD). Exercise extreme care when handling these devices to prevent damage.

Refer to Figure 7.1 for location of the hardware jumpers mentioned in the following paragraphs.

The eight configuration jumpers located on your video interface card can be used to set these initial power-up parameters: configure proper input polarity from a light pen device, select the desired character font when the power is turned on, and configure sync type and polarity to drive different types of monitors.

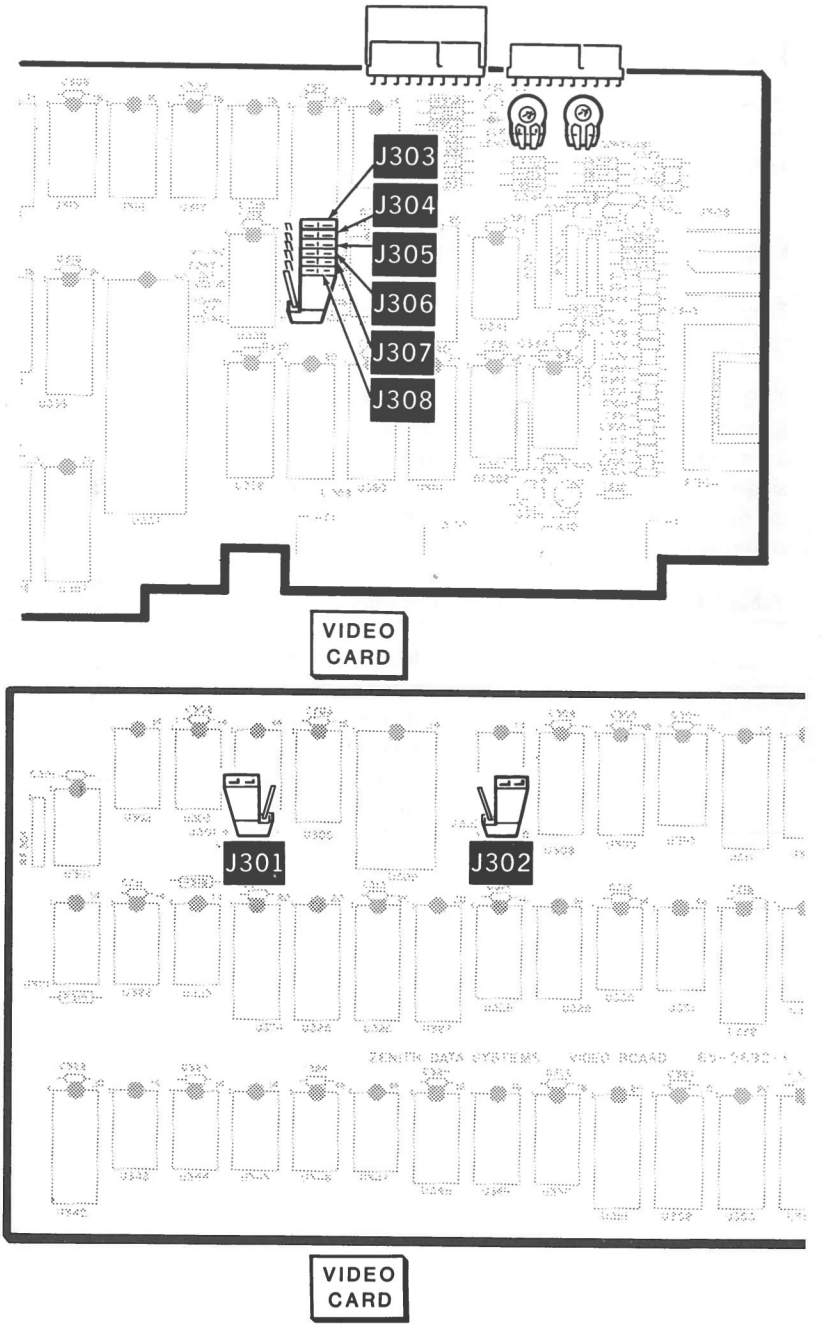

Figure 7.1. Video Card Jumper Locations

Light Pen Polarity

if you are using a light pen with a positive-going pulse output, position J301, the light pen jumper, to "+". If your pen outputs a negative-going pulse, position J301 to "−". The light pen option currently is not supported by Zenith Data Systems.

Character Font Selection

Figure 7.2 depicts the contents of the character font ROM, and the ASCII hexadecimal values which represent them.

J302 and J305 are the character-type selection jumpers. The condition of these jumpers is polled when you first turn on your computer and will determine whether the characters from the character font ROM are single-dot, double-dot, underlined single-dot, or underlined double-dot. Position the jumpers to the "0" or "1" markings on the card corresponding to the desired configuration. Table 7.1 outlines the character font selection.

Table 7.1. Power-Up Character Font Selection

J305	J302	CHARACTER FONT
1	0	Double-dot characters
1	1	Single-dot characters
0	0	Underlined double-dot
0	1	Underlined single-dot

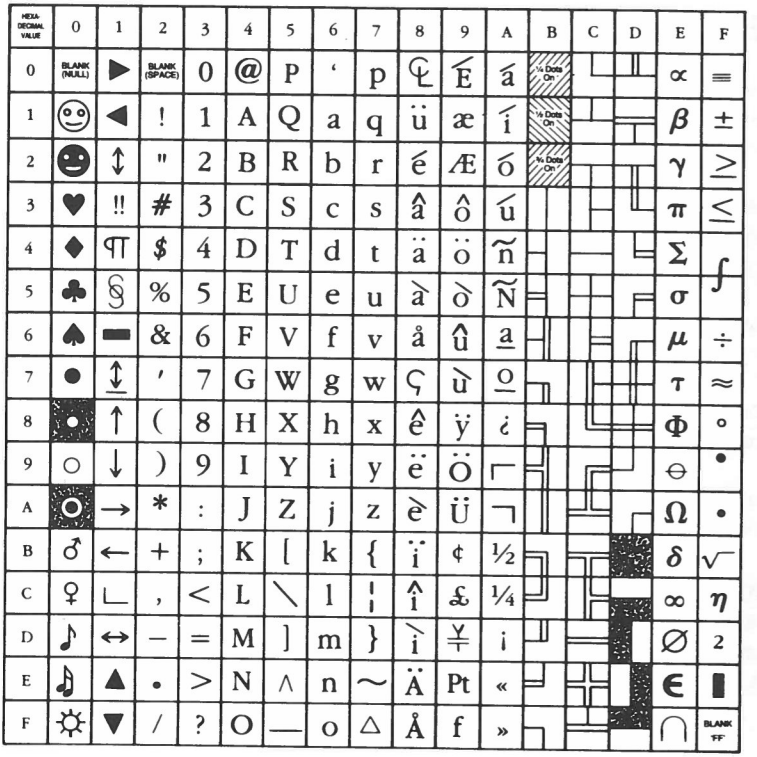

Figure 7.2. Character ROM Contents

Monitor Synchronization Selection

If you are using an external RGB monitor which requires composite (horizontal and vertical) sync, position J304 to "C". If your monitor requires separate horizontal and vertical sync, position J304 to "V".

Position J303 to the card markings corresponding to the horizontal sync polarity required by your internal monochrome monitor (Z-160 models), positive "+" or negative "−".

Position J306 to the markings corresponding to the vertical sync polarity required by your internal monitor (Z-160 models), positive "+" or negative "−".

Position J307 to the markings which correspond to the horizontal sync polarity required by your external RGB monitor, positive "+" or negative "−".

Position J308 to the markings which correspond to the vertical/composite sync polarity required by your external RGB monitor, positive "+" or negative "−".

NOTE: The jumpers are set at the factory for most Zenith Data Systems monitors. The jumpers J303 and J306 in the Z-160 models are properly set at the factory for the internal monitor's electronics so you shouldn't have to change these two jumpers. Jumpers J303 and J306 have no effect in the Z-150 models.

Major Components

The video circuitry is comprised of the following major components.

CRT Controller

A 6845 CRT controller IC provides the proper drive signals for a raster scan CRT. The device is highly programmable with regard to raster and character manipulation, and allows for the implementation of many display modes.

Mode Select and Status Registers

The mode select and status registers in the video interface are general-purpose programmable I/O registers which determine mode of operation and color on an RGB CRT display and levels of gray on a monochrome monitor.

Display Buffer

The display buffer, which resides in addressable memory at B8000H, incorporates 16 kilobytes of dynamic read/write memory. A dual port arrangement allows both the CPU and CRT controller equal access to this buffer in all modes of operation. In case of conflict, the CRT controller has higher priority in order to prevent display interference. This is invisible to the user program.

Monitor RAM

An additional 16K of RAM is contained on the video card and located at F0000 to F3FFF. This RAM is for use by monitor driven software only and should not be written to by the user. If data is inadvertently sent to this buffer, strange and unpredictable results may occur.

Character Generator

The ROM character generator on the video card, when in text mode, generates software-selectable characters from dot patterns stored in the ROM. Four character configurations are provided: 7 × 7 double-dot width, 5 × 7 single-dot width, and underlined versions of either of these character sets. You may install provided hardware jumpers to enable either of the character types at powerup, and/or use 8088 I/O OUT commands to program select your choice.

Timing Generator

The timing generator creates the timing signals for the 6845 CRT controller and the dynamic display memory. It also handles traffic control between the CPU and CRT controller for display buffer access.

Video Output

The video output circuitry routes the video information to the proper output connector. It sends color dots to the RGB connector and generates levels of gray for the monochrome output.

Modes of Operation

The video card can operate in one of two major modes: text mode and pixel addressable graphics mode. Additional submodes are available within each mode, for either color or monochrome display.

Text Mode

In text mode, 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 × 8 character matrix.

In either submode, the characters take up a 7 × 7 portion of this matrix space with one line for descenders in lowercase characters. Both uppercase and lowercase characters are available in all modes.

On a monochrome monitor, reverse video, blinking, sixteen foreground and background levels of gray, and highlighting features can be obtained on an individual character basis. On an RGB monitor, sixteen 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. Since there is 16K of video memory, up to eight screens can be stored in video memory. Also, since all the video memory is directly addressable by the CPU, extreme flexibility in manipulation of screen contents is obtainable.

Text Mode Display Architecture

A 2-byte character and attribute format is used to define each character display position as shown in Figure 7.3.

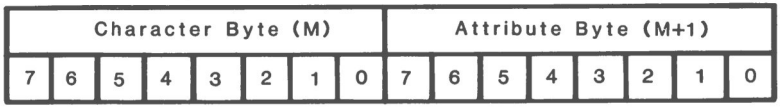

Figure 7.3. Character/Attribute Format

These two bytes are mapped into assigned locations in the screen memory buffer, which is part of the video interface not the system memory. Table 7.2 shows the codes assigned to the particular character attribute desired for a monochrome display. The starting address (character byte) must be an even numbered location.

Table 7.2. Monochrome Character/Attribute Selection

FUNCTION	ATTRIBUTE BYTE (BITS) 7	6	5	4	3	2	1	0
Normal video	B	0	0	0	I	1	1	1
Reverse video	B	1	1	1	I	0	0	0
Nondisplay (black)	B	0	0	0	I	0	0	0
Nondisplay (white)	B	1	1	1	I	1	1	1
Underline (normal, if enabled)	B	1	0	0	I	1	1	1

If B (bit 7) is logic 1, foreground blinks
If I (bit 3) is logic 1, foreground is intensified

Note that bits 0 through 2 and bits 4 through 6 are either all logic 1 (white) or all logic 0 (black) in the monochrome submode with the exception of the underline function. All other combinations of these bits are used to obtain the desired foreground and background colors on an RGB monitor or levels of gray on a monochrome monitor.

Table 7.3 shows the attribute byte breakdown. As before, bits 7 and 3 determine blink and intensity, respectively, of the foreground character.

Table 7.3. Color Attribute Byte Logic

COLOR	BACKGROUND BITS 6	5	4	FOREGROUND BITS 2	1	0
Black	0	0	0	0	0	0
Blue	0	0	1	0	0	1
Green	0	1	0	0	1	0
Cyan	0	1	1	0	1	1
Red	1	0	0	1	0	0
Magenta	1	0	1	1	0	1
Brown	1	1	0	1	1	0
Light gray	1	1	1	1	1	1

Intensified brown is yellow
Intensified light gray is white
Intensified black is dark gray

Using 25 × 80 format the same rules apply. But, since there are twice as many characters per row at this resolution, 4000 bytes of video memory are required to store one screen. This means only four screens may be stored at one time.

Graphics Mode

In graphics mode, two degrees of resolution are available: 320 × 200 and 640 × 200.

640 × 200 resolution is only obtainable as a monochrome display, since the full 16K of video memory is needed to define the ON or OFF states of a screenful of picture elements (pixels). If a color other than white is desired for the ON state of a given pixel, one of the 16 border colors may be selected. This is done by sending a value from 00H to 0FH (0 to 15 decimal) to the color select register at 3D9H after selecting 640 × 200 resolution.

In 320 × 200 resolution each pixel may be one of four colors. This color may be one of the sixteen background colors or one of three other colors software selected from the red/green/brown palette or the cyan/magenta/white palette.

Medium Resolution (320 × 200) Graphics

Pixel information is stored in video memory in two banks of 8000 bytes each, as shown in Figure 7.4. Address B8000 contains color selection data for the pixels in the upper left corner of the display area. Addresses B8000 to B9F3F contain information for pixels in even numbered rows 0 through 198. Addresses BA000 to BBF3F contain information for odd numbered rows 1 through 199.

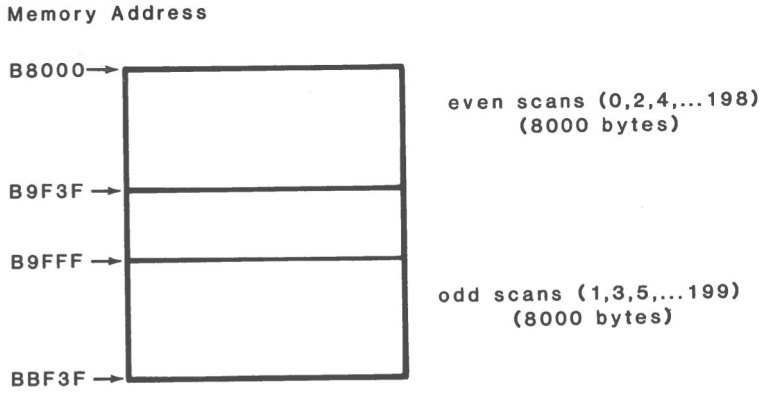

Figure 7.4. Graphics Storage Map

Each video memory byte contains color information for four pixels in 2-bit pairs as in the format shown in Figure 7.5.

7 6	5 4	3 2	1 0
C1 C0	C1 C0	C1 C0	C1 C0

Figure 7.5. Medium Resolution Mapping

For a given pixel bit pair composed of bits C0 and C1, the following logic, as shown in Tables 7.4 and 7.5, determines which color is selected from one of four colors: the current background color or one of the three colors from the current palette.

Table 7.4. 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.

Table 7.5. Palette Structure

COLOR #	SET 1	SET 2
1	Cyan	Green
2	Magenta	Red
3	White	Brown

The possible background colors are the same basic eight colors used in low resolution graphics, plus the lighter shades of these colors. This results when intensified, for a total of sixteen colors including black and white as shown in Table 7.3.

In order to obtain the lighter (intensified) shades, the monitor being used must be capable of recognizing the "I" bit.

High Resolution (640 × 200) Graphics

The high resolution (640 × 200) graphics submode may only be implemented in a monochrome format because of memory limitations. Addressing and mapping is the same as medium (320 × 200) 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 in Figure 7.6.

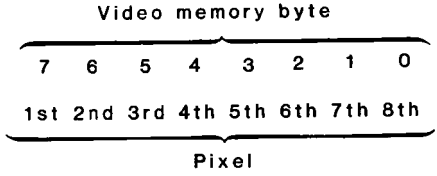

Figure 7.6. High Resolution Mapping

Programming Video Graphics

In the following descriptions all address and register values are given in hexadecimal notation. The term "set" in a given bit position indicates a logic 1 state. The term "reset" indicates a logic 0.

The memory used by the video interface is a self-contained 16 kilobyte bank with no parity bit. It is used in text mode for storage and retrieval of character and attribute information. In medium (320 × 200) resolution graphics mode, it contains data to define color attributes of the graphics pixels. In high resolution (640 × 200) graphics mode, it contains the ON or OFF status of the pixels. The buffer address starts at B8000H.

CRT Controller Signals

The following paragraphs define the functions and interrelationships of the signals appearing on the pins of the 6845 CRT Controller.

The signals fall into three categories:

- Signals which interface the controller with the microprocessor and system bus.
- Signals which interface the controller with the video memory bank and character generator logic.
- Signals which directly relate to controller/monitor interfacing.

System Bus Interface Signals

Table 7.6 defines the signals used to provide communication and data transfer between the video card and the rest of the system.

Table 7.6. Video Card to System Interface Signals

SIGNAL NAME	DESCRIPTION
A0 to A19	The 20 system address lines used to access memory locations.
D0 to D7	Bidirectional data lines which transfer information between the CPU and the internal registers of the 6845.
CS*	Chip Select signal typically generated by system address decoding logic. CS* must be low to read information from or write information to one of the 6845 internal registers.
RS	6845 register select signal. Rather than sacrifice five of the 40 available pins to 6845 register addressing, one of the registers is used as an address register to access the other 18 registers. When the RS signal is low, the address register is accessed and can then be loaded with the address of the internal register to be accessed next. When RS is high, the addressed register may be accessed.
R/W*	The read/write signal. Determines whether data is to be written into or read from an internal register. R/W* is low for a write operation, high for a read operation.

Table 7.6. (continued) Video Card to System Interface Signals

SIGNAL NAME	DESCRIPTION
E	Enables the internal I/O buffers and clocks data into and out of the internal registers via the data buffers. In your computer, the E input connects to the system ECLK signal.
CLK	Synchronizes the 6845's control signals. Derived from the system clock to become the character rate clock in text mode. Primary timing unit to the 6845.
RESET*	Initializes the 6845. When this pin goes low, the internal counters are cleared and the 6845 outputs go low, effectively stopping the video display operation. As the RESET* signal does not affect the program accessible counters within the 6845, display may continue when RESET* goes high.
Vcc (+5V) and Vss (Ground)	Standard TTL power levels which enable proper and operation of the video card ICs.

Screen Memory and Character Generator Signals

There are two sets of signals provided by the 6845 to implement screen memory and character generator logic interface. MA0 through MA13 are screen memory address outputs and RA0 through RA4 are the raster address signals sent to the character generator logic. The fourteen screen memory address lines allow the 6845 to address the 16K of video memory. The raster address lines are outputs from the 6845's scan line counter required by the character generator logic to determine which scan line of a character row is in progress.

CRT Monitor Control Signals

HSYNC and VSYNC are standard horizontal and vertical synchronization signals required by the CRT monitor for display stabilization.

DISPEN is the display enable signal which goes high whenever the video signal to the CRT is to be active. DISPEN is low during horizontal and vertical retrace and could also be called the video blanking signal.

CURSOR is the cursor enable signal which allows a steady stream of dots to be produced on the CRT screen as a cursor symbol.

LPSTB is the light pen strobe input signal which, in conjunction with ancillary circuitry, may be used to interface a light pen. A high on the LPSTB line signals the 6845 to save the contents of the screen memory address in one of the internal register sets so the microprocessor can subsequently determine the position on the monitor screen at which the light pen was detected.

CRT Controller Registers

The nineteen internal registers of the 6845 CRT controller can be software programmed to define display and control parameters of a raster scanned monitor. One of these registers, the address register, is used only as a pointer to the addresses of the other eighteen registers. The address register is write only and is loaded from the CPU using an I/O OUT instruction to the output port at 3D4. This register is loaded with the five LSBs of the OUT argument.

The other eighteen registers are selected by the pointer information in the address register. Then the CPU directs the information at address 3D5 to be loaded into the selected register. Transfers of address and parameter information to and from the registers is via data lines D0 through D7.

Programming the Registers

Programming Sequence

In general, you should observe the following sequence when selecting operating parameters for the video interface.

1. Determine desired mode of operation, text mode or pixel addressable graphics.

2. Reset video enable bit. Also reset video override bit 0 in port address 3DA if not already reset.

3. Program the CRT controller to desired mode.

4. Determine desired submode(s), then program mode and color select registers.

5. Set video enable bit. Also, set the video override bit, if desired.

Table 7.7 defines the individual registers in terms of values contained in them and the results obtained with these values. All values are expressed in hexadecimal.

Table 7.7. CRT Controller Register Functions

REGISTER ADDRESS	REGISTER NUMBER	REGISTER TYPE	(UNIT)	REGISTER STATUS	VALUES 40 × 25 TEXT	VALUES 80 × 25 TEXT	VALUES GRAPHICS MODE
0	R0	Horizontal Total	Character	Write	38	71	38
1	R1	Horizontal Displayed	Character	Write	28	50	28
2	R2	Horizontal Sync Position	Character	Write	2D	5A	2D
3	R3	Horizontal Sync	Character	Write	0A	0A	0A
4	R4	Vertical Total	Row	Write	1F	1F	7F
5	R5	Vertical Total Adjust	Scan Line	Write	06	06	06

Table 7.7. (continued) CRT Controller Register Functions

REGISTER ADDRESS	REGISTER NUMBER	REGISTER TYPE	(UNIT)	REGISTER STATUS	VALUES 40 × 25 TEXT	80 × 25 TEXT	GRAPHICS MODE
6	R6	Vertical Displayed	Row	Write	19	19	64
7	R7	Vertical Sync Position	Row	Write	1C	1C	70
8	R8	Interlace Mode	---	Write	02	02	02
9	R9	Maximum Scan Line Address	Scan Line	Write	07	07	01
A	R10	Cursor Start	Scan Line	Write	06	06	06
B	R11	Cursor End	Scan Line	Write	07	07	07
C	R12	Start Address	High	Write	00	00	00

Table 7.7 (continued). CRT Controller Register Functions

REGISTER ADDRESS	REGISTER NUMBER	REGISTER TYPE	(UNIT)	REGISTER STATUS	VALUES		
					40 × 25 TEXT	80 × 25 TEXT	GRAPHICS MODE
D	R13	Start Address	Low	Write	00	00	00
E	R14	Cursor Address	High	Read/Write	XX	XX	XX
F	R15	Cursor Address	Low	Read/Write	XX	XX	XX
10	R16	Light Pen	High	Read	XX	XX	XX
11	R17	Light Pen	Low	Read	XX	XX	XX

NOTE: XX is a hexadecimal number between 00H and FFH.

The first group of registers, R0 through R3, establishes the horizontal format and timing parameters.

Registers R4 through R9 determine vertical format and timing parameters.

The remaining registers, R10 through R17, deal with cursor characteristics, screen memory addressing, and the light pen interface.

Typically, registers R0 through R11 are loaded when the system is turned on and should not need to be accessed thereafter. The other six registers, R12 through R17, will probably need to be accessed on an ongoing basis during display operations.

R12 and R13 establish the 14-bit starting (top of page) address of screen memory. The contents of these registers may be manipulated to perform scrolling of the screen contents.

R14 and R15 establish the 14-bit cursor address which establishes the cursor symbol position on the screen.

R16 and R17 are only used if a light pen is interfaced.

Horizontal Timing and Format Registers

The contents of register R0 determines the total time allotted for one scan line in terms of character clock (CLK) cycles (total of displayed and undisplayed characters minus one) per horizontal line, thus determining the horizontal sync (HSYNC) frequency.

The character row register, R1, is loaded with the total number of characters minus one in character clock (CLK) units.

The horizontal sync register, R2, establishes the point at which the HSYNC signal makes its negative to positive transition, specified in terms of clocks. The reference point for the beginning of the HSYNC pulse is the left-most character position displayed on the scan line.

R3, the HSYNC width register, uses only the four LSBs of data to establish the duration of the HSYNC pulse in the range of 1 to 16 character clocks. This allows you to adjust the HSYNC pulse duration to the requirements of your respective CRT.

Vertical Timing and Format Registers

Registers R4 through R9 are normally loaded at system startup and are not normally changed later. The point of reference for these registers is the top-most character position displayed on the monitor screen.

The vertical total register, R4, and the vertical sync adjust register determine the total number of scan line times in a frame, including vertical retrace, establishing the overall frame rate or VSYNC frequency.

R4 is a 7-bit register loaded with the total number of character rows. Since a character row can consist of up to 32 scan lines, it may be difficult for you to establish a refresh frequency close to the line frequency. Register R5, a 5-bit VSYNC adjust register, may then be used to fine tune the VSYNC frequency. R5 is loaded with a value representing scan line times.

The character rows displayed register, R6, is a 7-bit register which allows you to select the number of rows of characters to be displayed, up to 128. What you specify for this register does not determine the position of the VSYNC pulse, but the point at which display enable (DISPEN) will be reset for vertical retrace.

R7, the vertical sync (VSYNC) position register, determines the point at which the VSYNC signal makes its negative to positive transition to initiate vertical retrace. VSYNC position is determined by the character row times, measured from the first character row on the monitor screen. The VSYNC pulse always has a duration of 16 scan lines. Since the scan line frequency will vary from one application to another, and since you cannot change the VSYNC pulse duration, external circuitry may be needed to achieve a pulse that is compatible with your CRT.

The interlace mode register, R8, determines whether interlaced or noninterlaced scanning is used.

The value in register R9 determines the number of scan lines per character row. You may program this 5-bit register for a character row of up to 32 scan lines. The value you use will dictate the maximum count output by the raster address signals (RA0 through RA4) to the character generator logic. Load R9 with the desired scan line count minus 1.

The cursor formatting registers are comprised of the cursor start and cursor stop registers, R10 and R11, respectively. The five LSBs of each register determine the scan lines within a character row where the cursor symbol is to be displayed.

The scan line specified in R10 is the first scan in which the CURSOR signal is to be set and it will remain set until the scan line specified in R11 has been completed. Accordingly, if you want the cursor symbol to occupy a single scan line, the same value must be loaded into both registers. If different values are loaded, a block-type cursor will be formed.

In interlaced sync and video mode both registers must be loaded with either odd or even values. Bits 5 and 6 determine whether or not the cursor is to blink, and if so, at either 1/16th or 1/32th of the field rate.

Primary Operating Registers

The remaining six registers, R12 through R17, are considered the primary operating registers, since, in the course of display programming, you will undoubtedly change the contents of them on an ongoing basis rather than loading them one time when the system is turned on. The six registers are arranged as three 14-bit register pairs (bits 6 and 7 of each most-significant byte are not used).

R12 and R13 comprise the top of page register which specifies the screen memory address containing the first character from the top left corner to be displayed. At the end of each vertical retrace, the first screen memory address generated will be the one contained in these registers. Since the 6845 addresses memory serially rather than on a row/column basis, scrolling can be performed on either a character by character or row by row basis.

The cursor position registers, R14 and R15, generate a cursor signal when the 6845 generates a screen memory address on MA0 through MA13 that matches the contents of this register pair, and when the scan line counter (RA0 through RA4) outputs fall within the limits established by registers R10 and R11. It may be necessary to delay the cursor signal with external logic circuitry to achieve display at the desired position.

Cursor movement on the screen is accomplished by loading new values into R14 and R15. These registers are the 6845's only read/write registers so they can also be used to keep track of the cursor position rather than copying this information from another memory location.

R16 and R17 are the light pen register pair that will be loaded with the screen memory address corresponding to the screen position at which the LPSTB signal was detected. Since there are several critical timing parameters involved with this signal, be sure to carefully study the documentation supplied with the light pen you are using.

Raster Scan Signals

The raster scan outputs, RA0 through RA4, provide the interface between the 6845 and the character generator logic. These outputs may represent scan line counts of from 0 to 31, or up to 32 scan lines per character row. Register R9, the scan lines/row register, determines the maximum count from the scan output registers before reset occurs. Scan line counts are incremented at the HSYNC rate but are also dependent on the interlace format selected.

Video Card Input/Output Devices

Table 7.8 defines the I/O registers contained on the video interface and the port assignments.

Table 7.8. Video Card Input/Output Port Assignments

PORT ADDRESS	REGISTER
3D0	6845
3D1	6845
3D8	Mode Control
3D9	Color Select
3DA	Status/Mode Input/Output
3DB	Light Pen Latch CLEAR
3DC	Light Pen Latch PRESET

To address a given port, the address lines are programmed as defined in Table 7.9.

Table 7.9. Video Input/Output Port Selection

PORT	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
3D0	1	1	1	1	0	1	0	X	X	0
3D1	1	1	1	1	0	1	0	X	X	1
3D8	1	1	1	1	0	1	1	0	0	0
3D9	1	1	1	1	0	1	1	0	0	1
3DA	1	1	1	1	0	1	1	0	1	0
3DB	1	1	1	1	0	1	1	0	1	1
3DC	1	1	1	1	0	1	1	1	0	0

X = Don't care

The 6845 CRT controller occupies two I/O port addresses. The RS (register select) line is connected to the least-significant address bit, A0. When A0 is reset, the address register is accessed to load it with the register address of the desired parameter which would then be accessed by setting A0 (RS).

A typical register access operation would then consist of two consecutive device write cycles, or a write cycle followed by a read cycle. These cycles would be directed to consecutive memory or I/O locations.

Color Select Register

The color select register, located at address 3D9, is a 6-bit output only device (it cannot be read) that can be written to using the I/O OUT command. Only the six LSBs of the instruction are used, as shown in Table 7.10.

Table 7.10. Color Select Register Logic

BIT	DESCRIPTION
0	B (blue) border color select (text mode) or 640 × 200 graphics dot color.
1	G (green) border color select (text mode) or 640 × 200 graphics dot color.
2	R (red) border color select (text mode) or 640 × 200 graphics dot color.
3	I (intensity) intensifies border color (text mode) or 640 × 200 graphics dot color.
4	Selects alternate intensified color palette.
5	320 × 200, graphics color palette selection.

Bits 6 and 7 are not used

In text mode, bits 0, 1, 2, and 3 determine the screen border color. In 320 × 200 graphics submode, these bits select the screen background color (C0 and C1). In 640 × 200 graphics submode, these bits select the pixel color. In this mode, the background will always be black, so selecting 0000 (black foreground) will render the display invisible.

Bit 4, when set, selects an alternate, intensified palette of background colors in text mode, if bit 5 in the mode select register, I/O port 3D8, is reset.

Bit 5 is only used in the 320 × 200 graphics submode to select the active palette of screen display colors. If bit 5 is a logical 1, color is determined by the logic of Table 7.11.

Table 7.11. Palette #1 Selection

C1	C0	BACKGROUND COLOR
0	0	Defined by bits 0-3 of port 3D9
0	1	Cyan
1	0	Magenta
1	1	White

If bit 5 is a logical 0, color is determined by the logic of Table 7.12.

Table 7.12. Palette # 2 Selection

C1	C0	BACKGROUND COLOR
0	0	Defined by bits 0-3 of port 3D9
0	1	Green
1	0	Red
1	1	Brown

Mode Select Registers

The mode select registers are also write-only registers. One is a 6-bit register at port address 3D8; the other is an 8-bit register at 3DA. Both can be written to using I/O OUT commands. The output register at 3D8 functions according to the logic of Table 7.13.

Table 7.13. Mode Select Port 3D8 Logic

BIT	DESCRIPTION
0	A logical 1 selects 80 × 25 text submode. A logical 0 selects 40 × 25 submode.
1	A logical 1 selects 320 × 200 graphics submode. A logical 0 selects text mode.
2	A logical 1 selects monochrome submode. A logical 0 selects color submode.
3	Disables the video signal during mode changes. The video signal should be reenabled following a mode change A logical 1 enables the video. This signal can be overriden by port 3DA Bit 0 (see Table 7.14).
4	A logical 1 selects 640 × 200 monochrome graphics submode. Use I/O port 3D9 to select one of eight colors when using a direct drive monitor in this submode.
5	A logical 1 selects blinking feature in alphanumeric mode (normal setting). A logical 0 intensifies whichever of the 16 background colors is selected to yield a lighter shade of that color.

Bits 6 and 7 are not used

The output register at 3DA functions according to the logic of Table 7.14.

Table 7.14. Mode Select Port 3DA Logic

BIT	DESCRIPTION
0	A logical 1 overrides the VIDEO ENABLE signal from 3D8. Eliminates monitor display flicker when in text mode.
1	Character font selection (see Table 7.15).
2	Character font selection (see Table 7.15).
3	Reserved for video diagnostics. Used to light the LED located on the video card.
4	A logical 1 disables the color video interface and connects video output from an alternate video card located within your central computing unit to output connectors on the resident video card.
5	A logical 1 enables fast hardware scrolling in the 80 × 25 submode only.
6 & 7	Used to select one of four pages of display text from the 16K video RAM, when bit 5 is set, according to the following logic:

7	6	Screen Page Select
0	0	1st page
0	1	2nd page
1	0	3rd page
1	1	4th page

Table 7.15 further defines the manner in which character font selection is determined.

Table 7.15. Character Font Selection

			MODE SELECT PORT 3DA		STATUS PORT 3DA	
J305	J302	*	BIT 1	BIT 5	BIT 6	FONT
1	0	0	0	0	0	Power-up double-dot
1	0	0	1	0	1	Single-dot
1	0	1	0	1	0	Underlined double-dot
1	0	1	1	1	1	Underlined single-dot
1	1	0	0	0	0	Power-up single-dot
1	1	0	1	0	0	Double-dot
1	1	1	0	1	1	Underlined single-dot
1	1	1	1	1	0	Underlined double-dot
0	1	0	0	1	1	Power-up underlined single-dot
0	1	0	1	1	0	Underlined double-dot
0	1	1	0	0	1	Single-dot
0	1	1	1	0	0	Double-dot
0	0	0	0	1	0	Power-up underlined double-dot
0	0	0	1	1	1	Underlined single-dot
0	0	1	0	0	0	Double-dot
0	0	1	1	0	1	Single-dot

* = Result of ANDing mode-select port 3DA bit 2 and AT6 (attribute bit 6, red background)

Status Register

The status register, an 8-bit read only buffer, resides at input port address 3DA. Table 7.16 summarizes the logical operation of this register.

Table 7.16. Status Port 3DA Logic

BIT	DESCRIPTION
0	Logical 1 indicates that a horizontal or vertical retrace is in progress. Used to update the video memory during screen refresh intervals.
1	Logical 1 indicates that a positive going light pen signal has set the light pen trigger. This trigger is reset at powerup and may also be cleared by issuing an I/O OUT command to port address 3DB. The reset is address activated; no specific data needs to be output.
2	Logical 0 indicates that the light pen switch is on. Switch signal is not latched nor debounced.
3	Logical 1 indicates that a vertical retrace is in progress.
4	LSB of the character font number.
5	MSB of the character font number.

Bits 6 and 7 are not used

Programming Interface Information

The information in this section repeats much of the foregoing, but is arranged in a format designed to clarify and link together related concepts and operations. Specific instructions are presented here, which, together with the previous data, will enable you to perform specific programming operations.

The 2-letter registers referenced here are 8088 microprocessor registers.

Video Interrupt Vector

Any of the the video features may be programmed by the use of the specific video function parameters, followed by an 8088 instruction call using the INTerrupt command. The necessary parameters for a desired function are first placed in the appropriate registers and the function code for that operation is placed in the AH register. Then the INT instruction for the video I/O, interrupt vector 10 hexadecimal, 16 decimal, is called.

The video I/O interrupt vector will allow several modes of operation to be implemented:

- Text displays using the RGB compatibility mode. All of these displays allow the use of multiple independent video pages.
- 80 character by 25 row text mode on an IBM monochrome display adapter card. Allows a single video page only.
- Graphics displays using RGB color compatibility mode. 128-character graphics character set contained in character ROM (on the video card) is used. An additional 128 user-defined characters may be created and used.

Table 7.17 defines the function codes (AH value) recognized by the system.

Table 7.17. Video Input/Output Function Codes

FUNCTION CODE	REGISTERS AFFECTED	OPERATION
0	AL	Set screen mode
1	CH, CL	Set cursor type
2	DH, DL	Set cursor position
3	DH, DL, CH, CL	Read cursor position
4	AH, DH, DL, CH, BH, BL	Read light pen position
5	AL	Select active display page
6	AL, BH, CH, CL, DH, DL	Scroll an area of screen up
7	AL, BH, CH, CL, DH, DL	Scroll an area of screen down
8	AH, AL, BH	Read cursor position
9	AL, BH, BL, CH, CL	Send character/attribute to screen
10	AL, BH, CH, CL	Send character to screen
11	BH, BL	Set graphics foreground color
12	AL, CH, CL, DH, DL	Write graphics pixel
13	AL, CH, CL, DH, DL	Read graphics pixel
14	AL, BH, BL	Dumb terminal display
15	AH, AL, BH	Returns current video state
100	AL	Set scroll mode

The following paragraphs describe the detailed use of the previously defined video functions.

Function Code 0—Set Video Mode

To set the desired video mode, a value from 0 to 7 is placed in the AL register before implementing the interrupt command. The recognized modes and their function codes are presented in Table 7.18.

Table 7.18. Video Mode Function Codes

FUNCTION CODE	VIDEO MODE DESCRIPTION
0	40 characters × 25 rows, monochrome display at the RGB output.
1	40 characters × 25 rows, color display at the RGB output.
2	80 characters × 25 rows, monochrome display at the RGB output. Individual video pages may be scrolled without affecting other video pages.
3	80 characters × 25 rows, color display at the RGB output. Individual text pages may be scrolled.
4	320 × 200 pixel resolution, color at the RGB output. Text emulation capability for 40 × 25.
5	320 × 200 pixel resolution, monochrome at the RGB output. 40 × 25 text emulation.
6	640 × 200 pixel resolution, monochrome at the RGB output. All three scrolling modes available (see scrolling selection).
7	80 × 25 text display on a monochrome display adapter card.

Function Code 1—Select Cursor Type

To set cursor type, load bits 0 to 4 of CH with the starting scan line value and load bits 0 to 4 of CL with the ending scan line value. While it is possible to load a value between 0 and 31, characters are normally eight scan lines tall, so it is advised that values in the range 0 to 7 be used. Also, the starting scan value may not be greater than the ending scan value.

Function Code 2—Select Cursor Position

Load DH with the desired cursor row number, and DL with the column number. DH = 0 and DL = 0 places the cursor in the uppermost left-hand position of the screen. Load BH with the desired video page number, consistent with the video mode selected. Also see information on setting the video page display functions.

Function Code 3—Read Cursor Position

Upon return from this call, DH will contain the cursor row, and DL the cursor column position.

Function Code 4—Read Light Pen Position

Upon return, AH will contain the light pen trigger/switch status, 0 or 1. If AH is 0, the light pen has either not been triggered, or not been switched active. If AH is 1, then DH will contain the row, and DL the column where the light pen was detected. The scan line number (0 to 199) will be in CH, and BX will hold the pixel column (0 to 319 or 0 to 639, depending on graphics mode).

Function Code 5—Select Active Display Page

In any text mode, the unused portion of video memory, if any, may be used for storage of video pages in excess of the current page. In video modes 0 and 1, the valid page numbers are 0 to 7; in modes 2 and 3, allowable values are 0 to 3. In the graphics modes, only page 0 is allowed. Load the page number into AL.

Function Code 6—Scroll an Area of the Screen Up

This code allows you to define an area of the screen to be scrolled upward a desired number of lines. CX is loaded with the upper left-hand coordinates (row in CH, column in CL), and DX with the lower right-hand coordinates. Load the attribute byte for blank lines in BH. Load AL with the number of lines to scroll. If AL is 0, the entire designated window will be cleared.

NOTE: If hardware or smooth scrolling are enabled when this function is implemented, they will be activated by this call if the entire screen is scrolled. It is not possible to use hardware or software scrolling on a portion of the screen.

Function Code 7—Scroll an Area of the Screen Down

This code allows you to define an area of the screen to be scrolled downward a desired number of lines. Load registers the same as for scrolling up.

Function Code 8—Read Character and Attribute

This function code will return the character and attribute byte codes for the character which resides at the current cursor position. Before calling, load the desired video page in BH for text modes. Upon return, AL will contain the character code, and AH the character attribute code.

Function Code 9—Write Character/Attribute to Screen

This code allows you to output a desired character and attribute to the current cursor position. Load BH with the desired page number, CX with a value representing the number of times to repeat the character, AL with the character code, and BL with the attribute code.

Function Code 10—Write Character Only to Screen

To use this function, load BH with the page number, CX the repeat count, and AL with the character code.

Function Code 11—Select Current Graphics Palette

This function is available only in 320 × 200 graphics mode. Load BH with a number from 0 to 127 to represent the identification of the palette. Load the actual color desired (0 to 4) value in BL.

If BH is even, the current background color will be used for the foreground graphics color, normally a value between 0 and 31. Values above 15 will select the intensified level of the 0 to 15 colors. See Table 7.3.

If BH is odd, one of the two available palettes is selected by the value in BL. If BL is 0, the green (1), red (2), yellow (3) palette is selected; if BL is 1, the cyan (1), magenta (2), white (3) palette is selected. The values in parentheses represent the color numbers used by the graphics calls.

Function Code 12—Write Graphics Pixel

This mode allows you to output a single pixel to a graphics coordinate on the current screen page. Load DX with a vertical value in the range 0 to 199 and CX to a horizontal value in the range of 0 to 319 or 0 to 639, depending on whether 320 × 200 or 640 × 200 video mode has been selected. Load AL with the color of the pixel (0 to 3) if in 320 × 200 graphics or (0 to 1) if in 640 × 200 mode.

If the MSB of AL is set, the color value given will be exclusively ORed with the current color value. This permits simple animation to be implemented.

Function Code 13—Read Graphics Pixel

This operation returns the color of a pixel in a given location, by returning the color value in AL. Before calling, load DX with the vertical position value, and CX with the horizontal value of the pixel.

Function Code 14—Dumb Terminal Display

Use of this function allows the user to display information on the monitor as if it were a simple terminal. Load the character to be written in AL, the foreground color in BL (for graphics modes), and the display page number in BH.

This routine will check and act upon the backspace (08H), carriage return (0DH), line feed (0AH), and bell (07H) as commands rather than characters, for screen formatting.

If a character occurs at the end of a screen line, the cursor is positioned to the start of the next line.

If a line feed is performed in the last display line on the screen, or if a character occurs in the last position of this line, the screen will be scrolled up one line. When scrolling is performed, the attribute for blanked rows in text mode is determined by the attribute at the cursor position on the previous line, prior to the scroll.

Function Code 15—Return Current Video State

Returns to the current video status. AL will hold the current video mode (function code 0), AH the screen width in character columns, and BH the currently active video page.

Function Code 100—Select Scrolling Mode

This routine allows you to select one of the three available scrolling modes. Load AL with a value between 0 and 2.

- If AL is 0, software scrolling is selected
- If AL is 1, hardware "jump" scrolling is selected
- If AL is 2, hardware "smooth" scrolling is selected

NOTE: Hardware scrolling is not permitted while in 40 × 25 text mode and smooth scrolling may only be used in the graphics modes.

Custom Character Creation

Normally when in graphics mode, characters called from the computer's ROM are the first 128 characters of the character font.

In addition to these standard characters, you may create your own 128 custom character set using the following procedure.

First allocate a 1 kilobyte section of user RAM to hold your characters. Eight bytes will be needed for each character you create.

Next design each character using an 8 × 8 matrix as shown in the example of Figure 7.7.

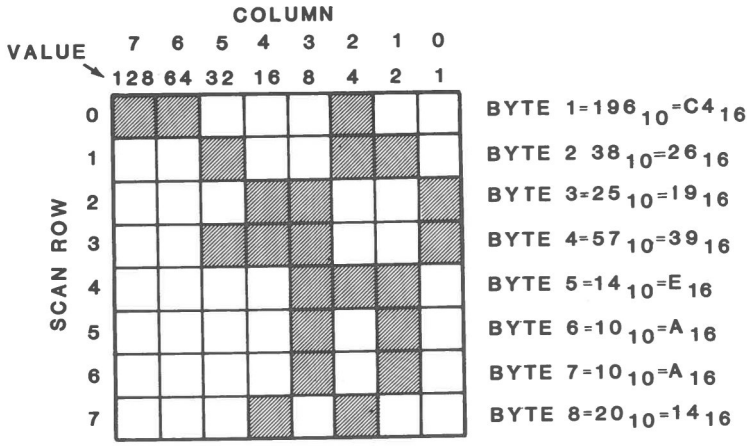

Figure 7.7. Character Design Matrix

Place a dot or fill in each location in the matrix which will be a solid part of your character. Then add up the binary weights of the spaces used in each row of the character grid. Note that the very first space is the MSB of each row. When you have values for all eight rows, you have all the data needed for the first character in your set. This data will be loaded into the first 8 bytes of the memory you have allocated for your character set.

Proceed in this manner to define each character in your set.

When you have all the characters defined and placed in memory, the first character will have a hexadecimal code of 80 and the last a code of FF.

All that remains now is to reset the pointer at interrupt vector 1F to the start of the memory section where your characters reside.

From now on, whenever you call a character code between 80 and FF in graphics mode, the corresponding character from the set you created will be used in place of the firmware character normally used.

Early Video Cards

If your computer is an early system, it may use a different video card instead of the present design. The two cards are easily distinguished from each other. The early card has seven configuration jumpers instead of eight, and the two connectors which project from the top rear of the present card are installed on the card field on the earlier design.

If your system uses the early design video card, you will need to use the following information to properly set the configuration jumpers for the proper operation of your system. Corresponding jumper numbers on the present video card do not correspond and perform the same functions on the early design video card.

Character Font Selection

Refer to Table 7.19 for the logic governing power-up character font selection, controlled by configuration jumpers J302 and J304.

Table 7.19. Character Font Selection—Early Video Card

J304	J302	CHARACTER FONT
0	0	Double-dot
0	1	Reserved
1	0	Single-dot
1	1	Reserved

Selection of a reserved option will result in a white screen or selection of a character type that is not wanted.

Light Pen Polarity

If you are using a light pen with your system, position J301 to "+" for a light pen with a positive going pulse output, or to "−" for a negative going signal.

Monitor Synchronization

If you are using an RGB (color) monitor which requires composite (horizontal and vertical) sync, position J303 to "C". If your monitor requires separate horizontal and vertical sync, position J303 to "V".

Position J305 to the video card markings corresponding to the vertical sync polarity required by your monochrome monitor, positive "+", or negative "−".

Position J306 to the markings corresponding to the horizontal sync polarity required by your RGB (color) monitor, "+" for positive sync, "−" for negative sync.

Position J307 to the markings corresponding to the vertical composite sync polarity required by your RGB (color) monitor, positive "+", or negative "−".

Chapter 8

Serial and Parallel Input/Output

Introduction

As with most of the other programmable features of the Z-100 PC Series Computers, the easiest and most direct means of manipulating the flow of serial and parallel data to a real world device (usually a printer), is by means of the routines provided in the monitor ROM, built into the machine.

However, certain applications may be more efficiently handled by using the information provided here, in conjunction with assembly level programming instructions.

Contained in this chapter is information for the serial and parallel device interrupts, optional configuration data, and tables showing locations of formatting parameters and layout architecture for device mapping.

Configuration

Configuration hardware jumpers are provided on the floppy disk controller card for the serial I/O ports and on the standard memory card for the parallel interface.

Serial Port Configuration

Serial port #1 (COM1), physically the top connector on the rear of the floppy disk controller card, is configured to interrupt request #4 (IR4). The lower connector, COM2, comes configured as interrupt request #3 (IR3) by section 2 of jumper block J502. You may configure this port to any other interrupt number desired in the range from 2 to 7, by moving the jumper at section 2 to one of the other sections of J502 corresponding to the desired interrupt number. You must bear in mind, however, that certain of these interrupts are already in use by other devices and confusion may result when using more than one of these devices at the same time. For instance, IR7 is the interrupt assigned to the parallel device port but may also be selected for COM2 if the jumper is placed at section 6 of J502.

NOTE: The disk controller I/O card may have only one RS-232 serial port. If this is the case, all references to COM2 may be ignored.

Parallel Port Configuration

Two hardware jumpers on the standard memory card, the one with the DB-25 connector installed, allow you to select the port address of the parallel interface. It is suggested that you normally use the configuration for port address 378H for a parallel printer, since the 3BCH address corresponds to the IBM PC monochrome port. Bear in mind also that two combinations of the configuration jumpers, J402 and J403, will disable the parallel interface. Table 8.1 shows the possible selections obtainable using these jumpers.

Table 8.1. Parallel Port Address Selection

J403	J402	PORT ADDRESS
0	0	3BC
0	1	378
1	0	Disable port
1	1	Disable port

Serial/Parallel Input/Output Tables

Figure 8.1 graphically illustrates the format for the serial and parallel device parameter mapping.

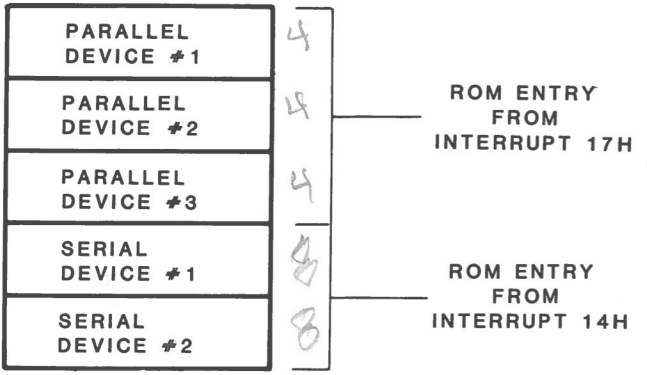

Figure 8.1. Serial and Parallel Device Layout

This is the table vectored to when a device is configured using the MS-DOS **CONFIGUR** program, but the parameters contained in this area of memory may also be changed by the user to suit his own unique device requirements.

A default value is contained in ROM to cause program branching to this table when a parallel interrupt, 17H, or a serial interrupt, 14H, is received. However, both the jump vector and the location of the table may be changed if desired.

Up to three parallel devices, and two serial devices, may be configured. In addition, a parallel device may be mapped to a serial device.

Parallel Format

The format for each of the three parallel maps is shown in Table 8.2.

Table 8.2. Parallel Map Format

BYTE	PARAMETER DESCRIPTION
1	Bits 0-3 perform a map Bit 4 strips parity on the output Bit 6 maps lowercase to uppercase
2	Pad character after carriage return
3	Number of pad characters to issue
4	ROM data segment time-out value for a parallel device

Serial Format

The format for the two serial maps is shown in Table 8.3.

Table 8.3. Serial Map Format

BYTE	PARAMETER DESCRIPTION
1	Type of handshake used by device
2	Attributes of transmission
3	Pad character after carriage return
4	Number of pad characters to issue
5	Flag for burst count for ETX/ACK
6	Counter for ETX/ACK
7	Device initialization
8	Reserved

Table 8.4 further defines byte #1 by bit.

Table 8.4. Serial Byte #1 Breakdown

BIT	PARAMETER DESCRIPTION
0	Compatibility bit. If logic 1, IBM-compatible DTR and RTS high. If logic 0, handshake determined by bit 1.
1	Protocol bit. If logic 0, hardware handshake. If logic 1, software handshake.
2	If bit 1 is logic 0, bit 2 is logic 1 if DTR handshake active, logic 0 if not. If bit 1 is logic 1, bit 2 is logic 1 if DC1/DC3 handshake, logic 0 if ETX/ACK handshake.
3	If bit 1 is logic 0, bit 3 is polarity of DTR if DTR active. If bit 1 is logic 1, bit 3 is logic 1 if waiting for handshake character, logic 0, if not.
4	If bit 1 is 0, bit 4 is logic 1 if RTS handshake active, 0 if not.
5	If bit 1 is logic 0, bit 5 is polarity of RTS if RTS active, 0 if low, 1 if high.

Table 8.5 further defines byte #2.

Table 8.5. Serial Byte #2 Breakdown

BIT	PARAMETER DESCRIPTION
0	If logic 0, do not strip parity on input; if logic 1, strip parity.
1	If logic 0, do not strip parity on output; if logic 1, strip parity.
2	If logic 1, map lowercase to uppercase on input; if logic 0, do not.
3	If logic 1, map lowercase to uppercase on output; if logic 0, do not.

Table 8.6 further defines byte #7.

Table 8.6. Serial Byte #7 Breakdown

BIT	PARAMETER DESCRIPTION
0 & 1	Word size
2	Number of stop bits
3 & 4	Define parity; yes/no odd/even
5, 6, & 7	Baud rate

Interrupts

The interrupts recognized by the system for the serial and parallel devices are 17H for the parallel I/O ports and 14H for the serial I/O ports. The following operation codes, when placed in the 8088 AH register, will result in specific actions as defined in the following paragraphs.

Parallel Port

Table 8.7 defines the results obtainable using an operational code followed by an INT 17H instruction.

Table 8.7. Parallel Device Operation Codes

OPERATION CODE	DESCRIPTION OF ACTION
0	Print the next character in AL. If the parallel device does not return a 'READY' status within a reasonable amount of time, AH will be set to 1, otherwise it will be 0.
1	Initialize the parallel port. The returned AH value will contain the device status. See code 2
2	Returns device status in AH as follows: Bit 0 – Time out error, device not ready Bit 1 – not used Bit 2 – not used Bit 3 – I/O error Bit 4 – Device on line Bit 5 – Out of paper signal Bit 6 – Character acknowledge Bit 7 – Device is busy or in an error state

Serial Port

The following operations are recognized using the serial interrupt, 14H. As before, AH is loaded with the operation code corresponding to the desired function. DX is loaded with the serial device designator, 0 or 1, before the interrupt is executed.

Table 8.8 defines the operation codes recognized by the system when using the serial interrupt.

Table 8.8. Serial Device Operation Codes

CODE	DESCRIPTION OF ACTION
0	Initialize the serial I/O port
1	Send character in AL to the serial port
2	Receive character in AL from serial port
3	Read communications status

The following paragraphs further define these serial device operations.

Function Code 0—Initialize the Serial I/O Port

The mode of the selected port is determined by the bit-mapped value in AL as shown in Table 8.9.

Table 8.9. Mode Select Byte Breakdown

BIT	DESCRIPTION OF PARAMETER
0	With bit 1, sets word length (See Table 8.10).
1	With bit 0, sets word length (See Table 8.10).
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 8.11).
4	With bit 3, sets parity selection (See Table 8.11).
5	With bits 6 and 7, sets baud rate (See Table 8.12).
6	With bits 5 and 7, sets baud rate (See Table 8.12).
7	With bits 5 and 6, sets baud rate (See Table 8.13).

Table 8.10. Word Length Selection

BIT 1	BIT 0	WORD LENGTH
0	0	Don't care
0	1	Don't care
1	0	7 bits
1	1	8 bits

Table 8.11. Parity Selection

BIT 4	BIT 3	SELECTION
0	0	No parity
0	1	Odd parity
1	0	Don't care
1	1	Even parity

Table 8.12. 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

Upon return from the initialization function, the status of the serial port will be in AX; AH will contain the line control status and AL the modem control status. See Function Code 3—Read Communications Status description for details.

Function Code 1—Send Character to the Serial Port

Upon return from this call, the MSB of AH will contain the transmit status. If the called routine could not send the character placed in AL, this bit is logic 1. The remaining AX bits will reflect the status as defined under Function Code 3.

Function Code 2—Receive Character from the Serial Port

Upon return from this call, bits 7, 4, 3, 2, and 1 of AH will contain the data transfer status. Again, refer to Function Code 3. If AH is logic 0, the routine reads a character properly into AL. If AH is not 0, some type of error occurred. In a case such as this, time out errors refer to either the absence of the Data Set Ready (DSR) or Clear To Send (CTS) signals.

Function Code 3—Read Communications Status

This function permits interrogation of the communications interface status, the condition of which is returned as a bit-mapped value in AX. AH contains the line control status and AL the modem control status, as shown by Tables 8.13 and 8.14.

Table 8.13. Line Control Status (Register AH)

BIT	STATUS
0	Received data ready.
1	Overrun error. The CPU did not read the previous character and a new character has arrived.
2	Parity error. Parity was enabled, but the parity of the incoming character did not match.
3	Framing error. An incorrect start/stop bit was received.
4	Break detect.
5	Transmitter holding register empty. Serial I/O channel is ready for another character to transmit.
6	Transmitter shift register empty. The serial I/O channel is not currently transmitting.
7	Time-out error. Device did not respond within a reasonable period of time.

Table 8.14. Modem Control Status (Register AL)

BIT	STATUS
0	CTS line has changed state.
1	DSR status has changed.
2	End of ringing pulse detector.
3	Carrier Detect (CD) signal has changed state.
4	Status of CTS line.
5	Status of DSR line.
6	Ringing indicator.
7	Carrier detect status.

Chapter 9

Disk Drives

Introduction

This chapter contains hardware and software programming information for the floppy disk drives, Winchester drives, and the drive controller cards.

Refer also to the section on the Central Processing Unit for the correct DIP switch settings for the desired system configuration, including: the selection of the auto-boot, drives present, and number of drives option.

Use the appropriate portions of this chapter for whichever drive type you have installed: Shugart or Mitsubishi floppy disk drives and/or the Seagate, Computer Memories (CMI) or MiniScribe Winchester hard disk drive.

Configuration

Various configuration jumpers and a switch on the Winchester drive controller are employed on the drive controller and drive signal separator cards to implement optional operational modes. This section will discuss the positioning of those jumpers for your particular application.

Winchester (Hard Disk) Controller

If you have a Winchester (hard disk) drive installed, refer to Figure 9.1 for the location of the two jumpers resident on the Winchester drive controller card.

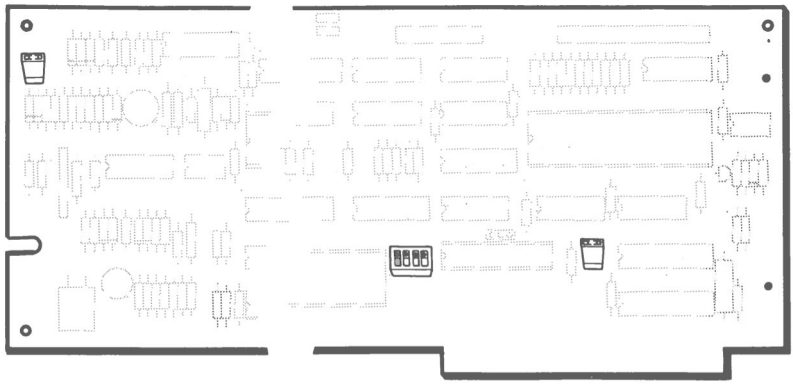

Figure 9.1. Winchester Controller Configuration

The jumper between E17 and E18 is provided for test purposes and is not normally jumpered.

The jumper on E21 is an address select jumper used to determine whether address C8000 or F4000 is used for the Winchester controller information. As factory configured (not jumpered), C8000 is selected. If E21 is jumpered, F4000 is selected.

Also located on the Winchester controller card is a 4-section switch, S1, which selects the number of hard disk heads/cylinders in use. Table 9.1 defines the head and cylinder selection for the section settings. The ON position of a section represents a logic 1 and OFF represents a logic 0. Sections 1 and 2 make the selection for drive 1, and sections 3 and 4 make the selection for drive 2.

Table 9.1. Winchester Head/Cylinder Selection

S1 SWITCH SECTION					
DRIVE 1		DRIVE 2			
1	2	3	4	HEADS	CYLINDERS
0	0	0	0	6	306
0	1	0	1	4	480
1	0	1	0	2	612
1	1	1	1	4	306

The last option is standard factory configuration

Floppy Disk Drive Controller

There are no jumpers on the floppy disk controller card associated with the floppy drives.

Mitsubishi Floppy Disk Drive Configuration

If your system is equipped with one or more Mitsubishi drives, use this section to configure your system. Refer to Figure 9.2 and Table 9.2 to configure the drive(s) by jumpering the appropriate pins. Remove the termination jumpers on all the drives except for those on the last drive on the cable.

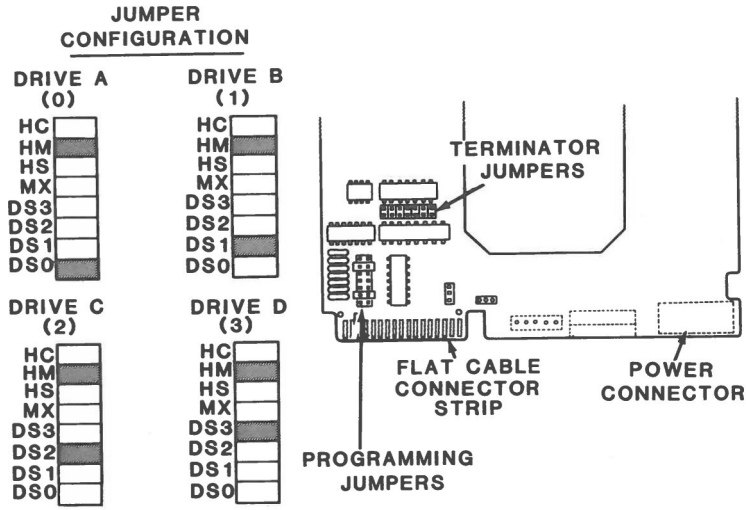

Figure 9.2. Mitsubishi Floppy Disk Drive Configuration

Table 9.2. Mitsubishi Floppy Disk Drive Configuration

	JUMPER POSITIONS							
DRIVE	DS0	DS1	DS2	DS3	MX	HS	HM	HC
A (0)	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF
B (1)	OFF	ON	OFF	OFF	OFF	OFF	ON	OFF
C (2)	OFF	OFF	ON	OFF	OFF	OFF	ON	OFF
D (3)	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF

Shugart Floppy Disk Drive Configuration

If your system is equipped with one or more Shugart floppy disk drives, use this section to configure your system. Refer to Figure 9.3 and Table 9.3 to configure the drive(s) by jumpering the appropriate pins. Remove the terminating resistor pack on all drives except for the last drive on the cable.

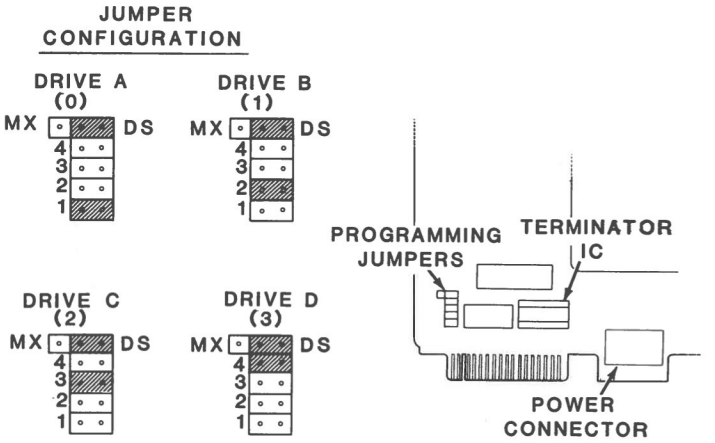

Figure 9.3. Shugart Floppy Disk Drive Configuration

Table 9.3. Shugart Floppy Disk Drive Configuration

	JUMPER POSITIONS				
DRIVE	1	2	3	4	MX-DS
A (0)	ON	OFF	OFF	OFF	Set to DS side of jumper
B (1)	OFF	ON	OFF	OFF	Set to DS side of jumper
C (2)	OFF	OFF	ON	OFF	Set to DS side of jumper
D (3)	OFF	OFF	OFF	ON	Set to DS side of jumper

Drive Function Calls

Table 9.6 defines the drive functions recognized by the system.

Table 9.6. Drive Functions

CODE	FUNCTION
0	Reset disk system.
1	Read disk status.
2	Read specified sectors into a buffer.
3	Write buffer contents to desired sectors.
4	Verify that selected sectors can be read without error. No information is actually transferred using this call.
5	Format track.
6	Flag selected track as a bad track.
7	Format drive starting with specified track.
8	Return current drive parameters.
9	Initialize drive type characteristics.
10	Read sectors including 4 ECC bytes at end of each sector.
11	Write sectors including ECC bytes.
12	Seek a desired track.
13	Reset Winchester controller only.
14	Read contents of controller's sector buffer into user- specified buffer.
15	Write user's buffer into sector buffer.
16	Test to see if specified drive is ready.
17	Recalibrate drive to track 0.
18	Execute controller RAM diagnostics.
19	Execute drive test diagnostics.
20	Execute internal controller diagnostics.

NOTE: Codes 6 through 20 apply only to Winchester drives

The following function codes further define the more elaborate functions.

Function Code 0—Reset Disk System

This call resets the disk software for each drive in the system. For floppy drives, interrupt vector 1EH should point to a drive parameter block as defined by Table 9.7.

Table 9.7. Drive Parameter Block

BYTE	PARAMETER
0	Disk controller mode byte 1.
1	Disk controller mode byte 2.
2	Number of clock ticks until motor is turned off following disk I/O.
3	Bytes per sector: 00 – 128 01 – 256 10 – 512 11 – 1024
4	Last sector of track.
5	Gap length in bytes.
6	Data length:. 80H if 128 bytes per sector FFH if not
7	Gap length when formatting.
8	Fill data used during formatting.
9	Head settling time (milliseconds).
10	Motor start time (.25 second increments).

Function Code 1—Read Disk Status

Returns the status in AL. This call is used by an applications program which requires the disk status generated by the last disk I/O operation. The status will be the same as that returned by the last I/O call (function codes 2 thru 5).

Table 9.8 defines the parameters required by function codes 2 through 5.

Table 9.8. Required Parameters for Function Codes 2 through 5

REGISTER	PARAMETER
DL	Drive ID Code as previously defined.
DH	Head (side) number. 0 to 1 for floppy drives 0 to 7 for Winchester drives
CH	Track number.
CL	Sector number. On Winchester I/O calls, the two MSBs of CL contain the two MSBs of the track number.
AL	Number of sectors to be transferred.
ES:BX	Pointer to the disk buffer; segment in ES. Offset in BX not required for function code 4.

Function Code 5—Format Track

When formatting a floppy disk, the buffer pointer, ES:BX, should point to a list of sector headers, containing one sector header for every sector on the track. AL should contain the number of sectors per track. The format of each sector header is:

Track Number
Head (disk side) Number
Sector Number
Bytes per Sector (see Function Code 0)

For Winchester drives, the data written to disk must first be loaded into the controller's sector buffer using function code 15.

Function Code 8—Return Current Drive Parameters

When completed, DL contains the number of active Winchester drives, DH contains a number representing maximum usable heads, CH contains a number representing maximum usable cylinders, and CL contains a number representing maximum usable sectors with the two MSBs representing the two MSBs of the maximum usable cylinder number.

Function Code 9—Initialize Drive Type Characteristics

Interrupt vector 41H should point to the new drive type parameter table, organized as four sets (for the four possible drive types) of information. This is formatted as follows:

Number of Tracks—Word
Number of Heads—Byte
Start Cylinder for current reduced write—Word
Start Cylinder for write precompensation—Word
Maximum number of bits to correct on data errors—Byte
Controller option byte:
    Bit 7—Disable retries
    Bit 6—Disable Error Correction Code (ECC)
    Bit 0 to 2—Step rate:
        000—3 m per step
        100—200 μsec per step
        101—approximately 60 μsec per step
        110—3 m per step
        111—3 msec per step
Time-out value for read/write—Byte
Time-out for format—Byte
Time-out for disk diagnostics—Byte
Reserved—4 Bytes

Error Status Codes

All of the drive function calls will return a status code in register AH upon completion. AH will either contain 0 if the carry flag is not set, or a value representing a specific error if the carry flag is set. Table 9.9 defines the error codes.

Table 9.9. Drive Error Codes

CODE	ERROR
01	Bad command. Illegal command was given to I/O routine.
02	Bad address mark. The controller did not find an address mark.
04	Write protect. A write or format operation was attempted but disk was write protected.
05	Winchester controller reset failed.
07	Controller did not accept drive parameters.
08	DMA overflow. DMA controller could not keep up with disk data and information was lost. Usually results from excessive DMA access from other devices on the da bus or a bus interface hardware failure.
09	DMA boundary error. Hardware is incapable of transferring sector information across 64K boundaries. Can be corrected by reducing sector count in AL or by changing buffer pointer so that entire transfer area can reside within a single segment.
10	Bad CRC on diskette. Specified record was found, but CRC for the data did not match what was calculated by the controller. On Winchester drives, flags that an error occurred which could not be rectified by the ECC circuitry.
11	Winchester drives only. Indicates that an ECC error occurred but controller was unable to reconstruct lost data.
20	Disk controller card is defective.
40	Bad seek. Controller attempted to move read/write head to specified track, but could not find sector header on that track which matched the correct track number.
80	Time-out error. Occurs when a command has been issued to the controller, but is n completed within a reasonable amount of time.
BB	Undefined error, usually bad controller.
FF	Winchester only. Indicates that a sense drive status operation failed.

Booting an Operating System

An operating system may be booted from a specified drive. Register DL should contain the drive number, and if the drive is a Winchester, AL should contain a partition number between 0 and 3 expressed in ASCII (30H to 33H) with 0 representing the default partition.

The routine will attempt to read from track 0, sector 1 of the specified device, and execute the code it finds. If the boot fails, either because the drive did not exist or because of a hardware failure, an error message is displayed and control is passed to the monitor ROM (see Chapter 2).

Interrupt vector 19H executes the boot function.

Index

NOTE: To locate a subject in this manual, first check the Contents for the location of general subjects or illustrations.

Index

Index

Index

Index

Index