Mindset Software Developers Guide

Table of Contents

Section 4
MINDSET-NATIVE GRAPHICS OPERATION

Section 5
CUSTOM SOUND-PROCESSOR OPERATION

List of Figures

List of Tables

Section 1

INTRODUCTION

THE MANUAL

This manual explains the operation of the Mindset Personal Computer from the point of view of an applications programmer. This manual refers to the applications programmer as "the user".

The descriptions in this manual are of basic operations in the Mindset computer. Many descriptions involve loading registers in the Intel 80186 microprocessor and interpreting bit fields in status registers. Familiarity with assembly language programming is therefore fundamental to understanding the descriptions in this manual.

You may find the following references useful while reading this manual:

1. Intel 80186 Hardware Reference Guide

2. Intel iAPX 86/88 186/188 User's Manual

3. Principles of Interactive Computer Graphics, by Newman and Sproull. (This is a standard graphics reference book, useful for defining terminology used in this manual.)

This manual emphasizes the use of software entry points (as opposed to direct hardware access) to avoid hardware-dependent programming. Hardware-dependent programming causes problems when the hardware is changed and upgraded. The use of software entry points protects against program obsolescence.

HOW TO USE THIS MANUAL

This manual is organized from general to specific. In each section and in the manual as a whole, introductory material comes first and operational details are described later.

Some conventions used in this manual are:

- Parentheses around the name of a register imply "the values in" that register. For example, (AX) means "the values in register AX". The register name by itself simply refers to the register.

- The least significant bit (LSB) of a byte or word is 0. The most significant bit (MSB) of a byte is 7 and of a word is 15.

Section 1 covers the theoretical bases of the design of the Mindset Personal Computer. This section includes an architectural overview and descriptions of the various components of the computer.

Section 2 is an introduction to ROM Basic Input/Output System (BIOS) operation. The ROM BIOS provides all the fundamental commands on the Mindset computer.

Section 3 describes the industry-standard BIOS functions that the Mindset computer provides. This section also describes the incompatibilities between the industry-standard and Mindset BIOS systems.

Section 4 describes Mindset-unique graphics operations. These operations go far beyond industry-standard graphics capabilities and support diverse graphics creations.

Section 5 describes custom sound-processor (CSP) operations in the Mindset computer. As with the graphics operations, Mindset provides a number of fundamental procedures in sound generation that support the production of a wide variety of sound creations.

Section 6 gives some basics on keyboard operations and features.

Section 7 describes the real-time clock and its operations.

Section 8 covers RAM/ROM cartridge features and operations in the Mindset computer.

Section 9 details the communications capabilities of the Mindset computer.

Section 10 covers miscellaneous BIOS commands offered on the Mindset computer.

Finally, Appendices A through G provide error messages, ASCII codes, keyboard scan codes, a system memory map, system stack usage information, a sound frequency table, and a sound generator sine-value table.

AN ARCHITECTURAL OVERVIEW

The Mindset computer system consists of a main system board/unit, the keyboard, an optional expansion unit, and a display (not supplied).

The main system board includes the Intel 80186 CPU, the display processor, the graphics coprocessor, two frame buffers, the custom sound-processor, the input/output (I/O) controller, the main read-only memory (ROM), the main random-access memory (RAM), and the bus interconnections structure.

The main system board is housed in the main System Unit. The main System Unit also houses two cartridge slots and the main double- and single-width I/O module ports. Cartridges can be plugged into the main unit to provide additional RAM and/or ROM features.

The keyboard provides the basic input to the system, along with optional joystick, mouse or other encoder-type input.

The Expansion Unit contains additional RAM, the one or two disk drives, and the expansion double- and single-width I/O module ports.

Modules that plug into the main unit or Expansion Unit include an RS-232-C Module, a Stereo Sound Module, a Printer Interface Module, and two types of Modem Modules.

The display for the Mindset computer can be a red-green-blue (RGB) monitor, a composite video monitor, or a television set.

Figure 1-1 shows the interconnections between the main system components.

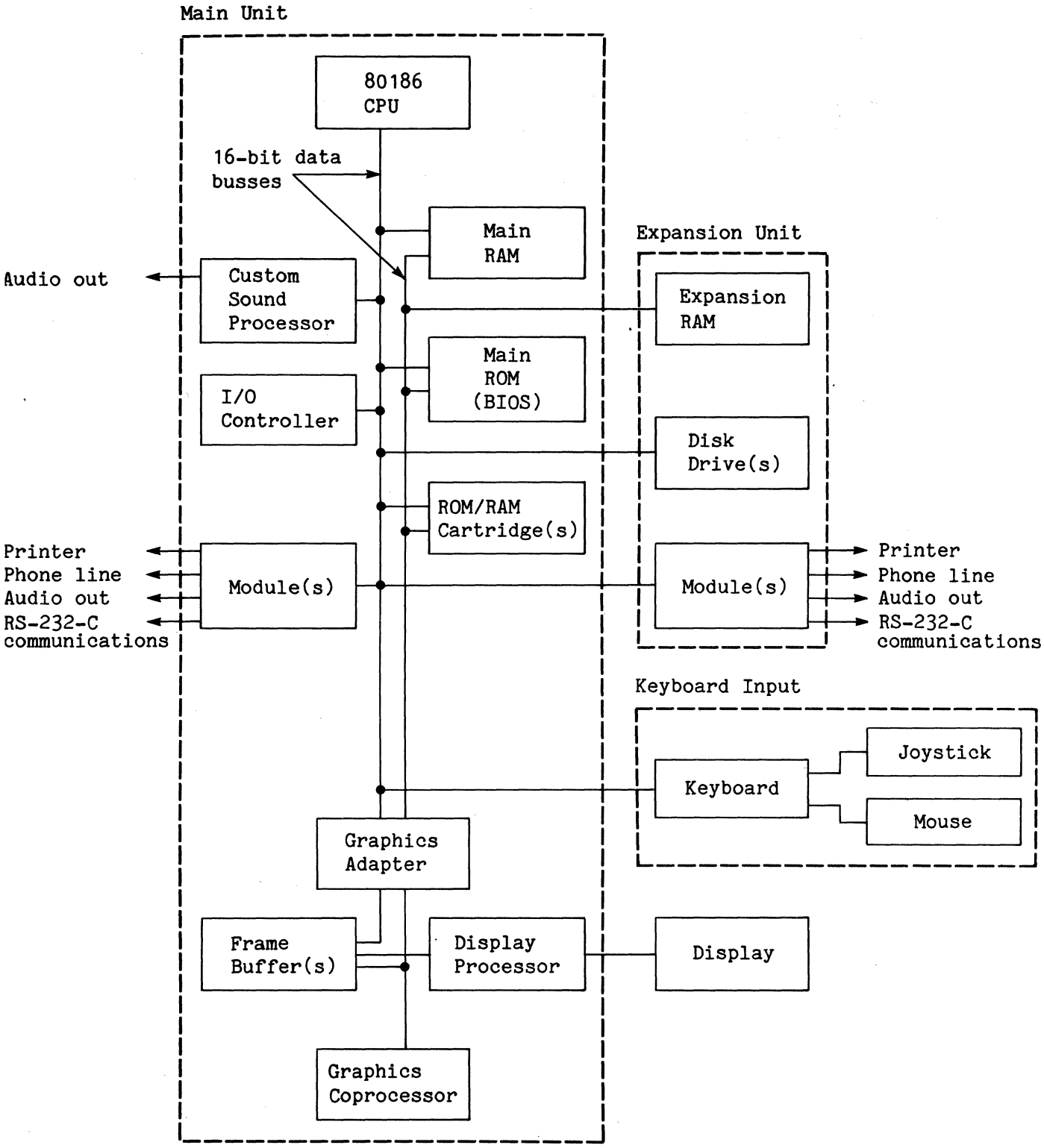


Figure 1-1: System block diagram

THE CPU, RAM, AND ROM

The CPU of a computer is its heart. It performs the basic operations necessary for reading intructions and executing them. The CPU performs program execution, logical comparisons, arithmetical operations, interrupt processing, and memory storage and retrieval. The CPU directs operations in the rest of the system, delegating some operations (such as input/output) to other devices as possible.

RAM is the main storage component in the system. For example, programs and data read into the system from disk are stored in RAM. It provides quick storage and access to data regardless of position in memory.

ROM provides system operations that are more complex than the basic operations of the CPU--the program routines in system ROM are, in fact, made up of the fundamental operations provided by the CPU.

MEMORY CARTRIDGES

Memory cartridges are options that provide extra RAM, extra ROM, or both. Extra RAM supports more complex programs and larger amounts of data, while extra ROM can provide new functions to the basic system set.

THE DISPLAY PROCESSOR AND GRAPHICS COPROCESSOR

The display processor performs the chores necessary for displaying bits of information on a cathode ray tube (CRT). The image on a CRT, which consists of hundreds of individual dots (pixels), must be refreshed or updated sixty times per second. If the CPU had to do this job, it would not have much time to do anything else. Therefore, the CPU delegates the job of maintaining a video display to the display processor.

The display processor in the Mindset computer also handles the two frame buffers. A frame buffer is a block of memory that can store the bits that define an entire video screen image. The second frame buffer gives the user the capability of creating an image in the inactive buffer and then switching buffers for instantaneous screen changes. Building images in an active frame buffer may be too slow or distracting for a particular application.

Three other display processor features are external video synchronization, transparent colors, and interlaced display. External video synchronization enables the Mindset computer to synchronize and superimpose its output with that of another video device such as a camera. The transparent-colors option allows an external video signal to show through a Mindset color. And interlaced display causes the Mindset computer to display 200 lines of data in 400 scan lines on the video screen. The result is a more filled-in look (higher apparent resolution) with less required data. Section 4, "Mindset-Native Graphics Operation", describes the display processor in more detail.

The graphics coprocessor (GCP) creates the graphic images in the frame buffers. It supports many powerful graphics operations such as block transfers, collision pattern and clip rectangle definitions, collision and clip detection, ellipse and polygon drawing, and bit selective data transfers. See Section 4 for descriptions of GCP capabilities.

THE CUSTOM SOUND-PROCESSOR

The custom sound-processor (CSP) is another special-purpose processor that both relieves the CPU of work and provides dedicated hardware for special operational features. The CSP provides four modes of operation that alternatively optimize music generation, sound effects and music generation, the largest number of voices, and direct digital-to-analog sound generation. You can specify different tones and noise masks to fine-tune your sound creations. The CSP even offers frequency- and amplitude-modulation controls and attack/decay controls for sound processing.

THE KEYBOARD PROCESSOR AND ASSOCIATED DEVICES

The keyboard processor (located in the keyboard unit) converts key strokes on the keyboard into the ASCII scan codes and specialized function codes used in the computer. In this way, the processor provides the basic input to the computer.

The real-time clock (RT clock), which is part of the keyboard processor, provides the time and date for application program uses. The RT clock also has an alarm that can be set to send a special interrupt at a predetermined time and date. This alarm can also "wake up" the system--turn it on and start preset operations without manual initiation.

Another feature convenient for custom programming is that of programmable system light-emitting diodes (LEDs). These LEDs (one green and one yellow) can be turned on and off as desired in an application to indicate status conditions.

A joystick or a mouse can be connected to the system as game or other application input.

I/O DEVICES

The RS-232-C Module provides numerous commands to facilitate RS-232-C communications between the Mindset computer and peripheral devices (such as a printer, external modem, or another computer, to mention a few).

Printer Interface Modules, when inserted into the Expansion Unit, support communications between the Mindset computer and printers.

Two Modem Modules are available that provide 300- and 1200-baud telecommunications capabilities for the Mindset computer.

Attaching a Stereo Sound Module adapter provides stereo sound effects when used in conjunction with the custom sound processor.

THE EXPANSION UNIT

As described above, the Expansion Unit supplies extra RAM (224 K-bytes), houses one or two disk drives, and provides additional I/O module ports.

Section 2

INTRODUCTION TO ROM BIOS OPERATION

HOW TO USE THE ROM BIOS ENTRY POINTS

The ROM BIOS (Read-Only-Memory Basic-Input/Output-System) provides operation routines that increase the usefulness of the Mindset Personal Computer. Each routine can be called as a basic function of the computer, just as simple addition is a basic function of a computer. To call a BIOS routine, the application program first specifies any parameters necessary for the routine and then calls the interrupt for the routine. The interrupt for a particular routine, which usually requires a specifying parameter in the AH register, is known as the entry point for that routine.

RECOMMENDED PROGRAMMING TECHNIQUES

The ROM BIOS constitutes a software interface between user applications and the system hardware. Rather than directly accessing hardware, the user should always employ the routines from the ROM BIOS to set up and control the operations of the Mindset computer. However, the frame buffers are an exception and may be accessed directly as desired.

INDUSTRY-COMPATIBLE FUNCTIONS

The Mindset computer offers a nearly complete set of industry-standard BIOS functions. These functions support programs written for industry-standard personal computers. Many industry-standard programs will run on the Mindset computer with little or no modification.

SYSTEM INTERRUPTS

Table 2-1 lists the BIOS interrupts of the Mindset computer system.

Table 2-1: Interrupt Vectors

Interrupt number (Hexadecimal)		80186 Definition	BIOS Definition
0		Divide Error	Dummy IRET
1		Single Step	Dummy IRET
2		Nonmaskable	Reserved
3	80186	Breakpoint	Dummy IRET
4	Interrupts	INT0 Detect Ovfl	Dummy IRET
5		Array Bounds	Print Screen
6		Unused Opcode	Dummy IRET
7		ESC Opcode	Dummy IRET
8		Timer 0	Timer Interrupt
9		Reserved	Dummy IRET
A		DMA 0	Refresh Interrupt
B	Hardware	DMA 1	Dummy IRET
C	Interrupts	INT0	Module IRQ
D		INT1	Late VBLANK (flyback) Interrupt
E		INT2	Display Interrupt
F		INT3	Keyboard/GCP
10		Unused	Video I/O
11		Unused	Equipment Check
12		Unused	Memory Size
13		Unused	Diskette I/O
14	Industry-Standard	Unused	RS-232-C
15	Software	Unused	Dummy IRET
16	Entry Points	Unused	Keyboard I/O
17		Unused	Printer I/O
18		Unused	Dummy IRET
19		Unused	Bootstrap
1A		Unused	Time-of-Day
1B		Unused	Keyboard Break
1C	User	Unused	Timer Tick
1D	Definable	Unused	Video Init Params
1E	Entry Points	Unused	Disk Parameters
1F		Unused	Video Alt Chars
E0-ED	Mindset-Unique	Unused	Mindset Reserved
EE	Entry Points	Unused	Mindset Unique
EF		Unused	Mindset Graphics

MINDSET-NATIVE ENTRY POINTS

The following two lists present the Mindset-unique ROM BIOS commands offered under interrupts EE hexadecimal and EF hexadecimal. ("Hexadecimal" is abbreviated to "hex" or "H" in this manual.)

The first list includes the commands under interrupt EEH, which is the entry-point interrupt for the general ROM BIOS functions. The value of register AH selects the specific command.

Command	Value of AH Decimal	Hex	Function
Write TTY String	00	00	Writes a character string to a screen buffer.
Write TTY String with Attributes	01	01	Writes a character string with character attributes to a buffer.
Set Display Device	02	02	Sets the display for either a television or an RGB monitor and reloads the color palette. Also enables or disables color output to a television display.
Set Screen Position	03	03	Sets the screen position within the screen border.
Get Screen Position	04	04	Returns the current position of the screen within the screen border.
Set Cursor Shape	05	05	Sets the shape of the cursor for character modes.
Set Display Sync Features	06	06	Enables or disables the use of genlock for transparent colors, enables or disables interlaced sync display, and enables or disables fixed-phase display.
Set Display Interrupt	07	07	Specifies the scan line at which the display processor causes the CPU to perform a user-defined interrupt service routine.
RT Clock Enable Wake-up/Alarm	08	08	Enables/disables the real-time clock wake-up/alarm options.
RT Clock Get Wake-up/Alarm Status	09	09	Returns the enabled/disabled status of the real-time clock wake-up/alarm options.

Command	Value of Decimal	AH Hex	Function
RT Clock Set Status	10	0A	Sets or clears the valid/invalid status of the real-time clock.
RT Clock Get Status	11	0B	Returns the valid/invalid status of the real-time clock.
RT Clock Read Time	12	0C	Returns the time from the real-time clock.
RT Clock Set Time	13	0D	Sets the time on the real-time clock.
RT Clock Read Date	14	0E	Returns the date from the real-time clock.
RT Clock Set Date	15	0F	Sets the date on the real-time clock.
RT Clock Read Alarm	16	10	Returns the time of the alarm-interrupt setting from the real-time clock.
RT Clock Set Alarm	17	11	Sets the time for the alarm interrupt.
RT Clock Set Int	18	12	Sets the interrupt pointer for the user-specified alarm-interrupt service routine.
Reserved	19	13	N/A
ROM/RAM Cart Status	20	14	Returns the operational data for the cartridge and reads the cartridge wait state specification into the system.
RAM Cart Format	21	15	Formats a RAM cartridge to accept information and support a directory.
RAM Cart Get Entry	22	16	Returns RAM cartridge file entry information from the cartridge directory.
RAM Cart Put Entry	23	17	Writes RAM cartridge file entry information to the cartridge directory.
RAM Cart Read	24	18	Reads blocks of data from a RAM cartridge file.
RAM Cart Write	25	19	Writes blocks of data to a RAM cartridge file.

Command	Value of AH Decimal	Hex	Function
RAM Cart Delete Block	26	1A	Deletes blocks of data from a RAM cartridge file.
Reserved	27	1B	N/A
Reserved	28	1C	N/A
Reserved	29	1D	N/A
Reserved	30	1E	N/A
Joystick Mouse I/O	31	1F	Reads the status of two joystick/ button, mouse, or other encoder-type controls.
Get Module ID Table	32	20	Returns a pointer to the table of the IDs of the installed modules.
Set Power Off	33	21	Turns off the power to the system.
Set LEDs	34	22	Turns the system LEDs on or off.
Print String	35	23	Prints a string of characters at the printer.
Set Sound Mode	36	24	Sets the operation mode of the custom sound processor(s).
Set Sound Register	37	25	Sets the registers that control sound processor operations.
Sound Data	38	26	Transfers sound data directly from the CPU to the system D/A converter.
Stereo Module Check	39	27	Checks for the presence of a Stereo Sound Module.
RS-232-C Send Character	40	28	Writes a character to the RS-232-C output buffer.
RS-232-C Get Character	41	29	Reads a character from the RS-232-C input buffer.
RS-232-C Send String	42	2A	Writes a character string to the RS-232-C output buffer.
RS-232-C Get Buffer Status	43	2B	Returns the available space in the RS-232-C input and output buffers.

Command	Value of AH Decimal	Hex	Function
RS-232-C Set Input Buffer	44	2C	Specifies the RS-232-C input buffer.
RS-232-C Set Output Buffer	45	2D	Specifies the RS-232-C output buffer.
RS-232-C Set Communications Control	46	2E	Specifies RS-232-C communications controls and interrupt enables.
RS-232-C Get Communications Control	47	2F	Returns RS-232-C communications modem controls and interrupt enables.
RS-232-C Get Modem Status	48	30	Returns information on modem installations.
RS-232-C Set Communications Break	49	31	Sends a break signal over the RS-232-C communications line.
Set Auxiliary Output Port	50	32	Sets the system auxiliary output port logic state.
Set Module Int	51	33	Specifies the interrupt pointer for the user-specified module interrupt service routine.
Set System Timer Rate	52	34	Changes the rate at which the system timer interrupts.
Get System Timer Rate	53	35	Returns the rate at which the system timer interrupts.
Enable/Disable Beeper	54	36	Enables or disables the system beeper.
Check Beeper Enable	55	37	Indicates whether the beeper is enabled.
Set Beeper	56	38	Turns the beeper on or off, provided it is enabled.

The second list presents the commands under interrupt EFH. These commands are the Mindset-unique extended graphics commands. The value of register AH selects a specific graphics command from the following list.

Command	Value of AH Decimal	Hex	Function
Set Screen Mode	00	00	Selects graphics resolution, number of colors, single or double buffering, and interlaced or non-interlaced operation.
Get Screen Mode	01	01	Returns the current screen mode parameters.
Set Transfer Mode	02	02	Specifies the transparent or opaque mode and the NOT, AND, OR, or exclusive-OR mode for GCP data transfers.
Get Transfer Mode	03	03	Returns the current GCP transfer mode parameters.
Set Destination Buffer	04	04	Specifies the address and size of the BLT destination buffer.
Get Destination Buffer	05	05	Returns the address and size of the current BLT destination buffer.
Set Write Mask	06	06	Specifies a 16-bit write mask that enables selective bit modification during data transfer operations.
Get Write Mask	07	07	Returns the 16-bit write mask.
BLT Copy	08	08	Copies a region of the source buffer to the destination buffer.
BLT Copy Word	09	09	Fills rectangular regions of the destination buffer with a user-specified 16-bit fill pattern.
Set Palette	10	0A	Specifies the screen border color and transfers color data from a user-specified area of memory to the color palette.
Get Palette	11	0B	Returns the value of the border color and transfers color data from the palette to a user-specified area of memory.
BLT Polypoint	12	0C	Draws a series of points of the same color into the destination buffer.

Command	Value of AH Decimal	Hex	Function
BLT Polyline	13	0D	Draws a series of lines of the same color into the destination buffer.
Get Buffer Info	14	0E	Returns the addresses and sizes of the system frame buffers.
Set Display Int Address	15	0F	Specifies the address of the user-defined display interrupt service routine that the system calls as part of each display interrupt.
Get Display Int Address	16	10	Returns the address of the user-defined display interrupt service routine.
Switch Active Buffer	17	11	Switches to the alternate frame buffer if double buffering is possible.
Set Collision Pattern	18	12	Specifies the pattern that, when found, indicates a collision.
Get Collision Pattern	19	13	Returns the current collision pattern.
Set Clip Rectangle	20	14	Specifies the clipping bounds for graphics commands.
Get Clip Rectangle	21	15	Returns the current clipping bounds.
Set Collision/Clip/ Done Detect	22	16	Sets the collision and clip enable flags and the address for the user-defined collision/clip/done-interrupt service routine.
Get Collision/Clip/ Done Detect	23	17	Returns the settings of the collision and clip enable flags and the address for the user-defined collision/clip/done-interrupt service routine.
GCP Wait	24	18	Waits until the block-transfer hardware completes its current task and then returns the collision/clip status.
BLT Polygon	25	19	Draws a filled polygon into the destination buffer.

Command	Value of AH Decimal	Hex	Function
BLT Filled Ellipses	26	1A	Draws a series of filled ellipses or ellipse sectors into the destination buffer.
BLT Hollow Ellipses	27	1B	Draws a series of hollow ellipses or elliptical arcs into the destination buffer.
Save GCP	28	1C	Saves the current state of the graphics coprocessor in the user-specified data area.
Restore GCP	29	1D	Restores data previously saved by the Save GCP routine.
Fill Dest Buffer	30	1E	Fills the destination buffer with specified data.
Set Font Pointer	31	1F	Sets a pointer to a user-defined font descriptor block.
Get Font Pointer	32	20	Returns the pointer to a user-defined font descriptor block.
BLT String	33	21	Transfers a string of characters into the destination buffer.
Set Param Block Mode	34	22	Specifies the representation for certain BLT command parameter blocks.
Get Param Block Mode	35	23	Returns the current parameter block representation mode.
Get GCP Status	36	24	Returns the current GCP status word.
Get Char Bitmap Address	37	25	Returns the addresses of the two system character bitmaps.
Get BLT Memory Bounds	38	26	Returns the address bounds of memory available for block transfers.

HOW THIS MANUAL GROUPS BIOS FUNCTIONS

This manual describes industry-compatible functions first, in Section 3. These functions provide industry-standard compatibility by performing operations common to industry-standard personal computers.

The second group of commands covered, in Section 4, are the Mindset-native graphics commands. These commands constitute the powerful graphics capabilities of the Mindset computer. The Mindset-unique graphics section describes the functions for both the display processor and the graphics coprocessor.

Section 5 in the manual describes the custom sound-processor (CSP) operations and capabilities. The CSP provides a fundamental set of sound-processing options that give you great creative opportunities.

Keyboard operations are covered in Section 6. This section discusses the industry-standard scan codes of the Mindset keyboard and the Mindset-unique keys also provided. In particular, the single-key-operated option is explained in detail.

The real-time clock is the subject of Section 7. This section covers the features of the real-time clock and details the corresponding commands.

Section 8 covers RAM/ROM cartridge operation. Again, the section discusses the operational features and details the corresponding commands.

Mindset-native communications commands are the subject of Section 9. This section enumerates the many commands the Mindset computer provides for versatile RS-232-C standard communications and describes the two modem modules and their commands that enable telecommunications.

Finally, Section 10 covers miscellaneous commands such as those that provide enhanced string operations, system wake-up capability, automatic power-off, and LED control. Also in this section are descriptions of joystick and mouse input commands, beeper control commands, system timer commands, and the auxiliary device control line.

Section 3

INDUSTRY-COMPATIBLE BIOS FUNCTIONS

OVERVIEW OF COMPATIBILITY AND INCOMPATIBILITY

In the Mindset Personal Computer, access to all hardware and devices occurs through the ROM BIOS; access to the ROM BIOS is by means of software interrupts. These software interrupts are generally entry-point compatible with industry-standard software interrupts. This compatibility means that most programs designed for an industry-standard personal computer will run on the Mindset computer with no special modifications.

The ROM BIOS provides a software interface between user applications and the system hardware. Rather than directly accessing hardware, the user should always use the routines from the ROM BIOS to set up and control the operations of the Mindset computer.

The Mindset computer offers features beyond those of the industry-standard computer. As an applications programmer, you should be aware of the extra features provided by the Mindset computer and how they affect compatibility with industry-standard operations. The remainder of Section 3 describes the compatibilities and incompatibilities between the Mindset and industry-standard personal computers.

MINDSET FEATURES THAT ARE INCOMPATIBLE WITH INDUSTRY STANDARDS

In some areas of operation, the Mindset BIOS routines differ from those of the industry-standard personal computer. These areas include the PRINT SCREEN interrupt command, the video I/O interrupt, the keyboard interrupt, the disk bootstrap interrupt, and the cassette interrupt.

Interrupt 05H is normally the industry-standard PRINT SCREEN command on the Mindset computer. The 80186 central processor chip, however, uses interrupt 05H to signal that an array index is outside of program-specified bounds when checked with the BOUNDS instruction.

Because the BOUNDS instruction is not part of the instruction set of the 8088 (or 8086), it cannot be part of any program written for an industry-standard personal computer. The compatibility problem arises if a program for the Mindset computer uses the BOUNDS instruction. In this case, the program must replace the PRINT SCREEN vector with a vector to an exception-handling subroutine for the BOUNDS instruction.

The industry-standard video I/O interrupt (10H) is supported in both the industry-standard character modes and the Mindset-native graphics modes, but with the following differences:

- In the industry-standard character modes, the Mindset computer provides half the number of character pages (2 in 80-column mode and 4 in 40-column mode).

- Where double buffering is possible in Mindset graphics modes, the page number (0 or 1) is required for the character-read- and character-write-type commands.

- In Mindset graphics interlaced modes, characters are 16 scan lines high, not 8.

- The WRITE TELETYPE command of the Mindset computer, unlike that of the industry-standard personal computer, actually does use the page number passed in BH. The industry-standard personal computer restricts the output of this command to the active page.

Other industry-standard reserved interrupts handled in a nonstandard way by the Mindset computer are the keyboard interrupt (09H), the disk bootstrap interrupt (19H), interrupts 0AH through 0DH, and the cassette BASIC interrupt 18H.

For the industry-standard computer, interrupt 09H is the keyboard interrupt. The Mindset computer reserves interrupt 09H for future use.

Interrupt 19H, the disk bootstrap interrupt for the industry-standard personal computer, is the cartridge and disk bootstrap interrupt for the Mindset computer. The Mindset computer may try to boot from a ROM or ROM/RAM cartridge before it tries to boot from a diskette, depending on the priorities set on the system configuration screen.

Interrupts 0AH through 0DH are reserved (but not dedicated) by the BIOS of the industry-standard personal computer. The Mindset computer uses 0AH through 0DH for the 80186 hardware functions DMA 0, DMA 1, INT 0, and INT 1, respectively, as listed in Table 2-1.

The BIOS of the industry-standard personal computer specifies that interrupt 18H is for cassette I/O, but the BIOS of the Mindset computer replaces interrupt 18H with a dummy IRET instruction.

The Mindset computer uses interrupts EEH and EFH to provide Mindset-unique operations.

CHARACTER MODE OPERATION

Character mode operation of the Mindset computer is compatible with that of the industry-standard color/graphics adapter, with three exceptions:

1. The Mindset computer provides 2 pages of text storage for 80-column mode and 4 pages for 40-column mode. The industry standard is 4 pages and 8 pages, respectively. See Section 4, "Mindset-Native Graphics Operation", for screen format details.

2. The Mindset computer enables the user to redefine the cursor shape. The industry standard is a variable-height block cursor the width of one character. See Section 4 for a description of the cursor shape definition command.

3. The character modes of the Mindset computer provide two colors instead of the eight colors and two intensities of the industry-standard personal computer. See Section 4 for screen format details.

In character modes, there is a buffer of several pages of ASCII characters with attributes. There are 4 pages in 40-column modes, and 2 pages in 80-column modes. The buffer of ASCII characters with attributes is used during early VBLANK (vertical flyback) time to draw the display buffer.

Each display page has a cursor position. The cursor for the active page (the page currently displayed) is drawn on the screen. The cursor for any other page is the location at which characters are put or read; it is not displayed.

Page numbers start at 0 for the first page.

Rows begin at 0 for the top row on the screen, and go to 24. Columns begin at 0 for the left-most column on the screen, and go to either 79 or 39.

The format of the display page buffers in industry-standard character modes is an array of words. Each word contains an ASCII character code as the lower byte and an attribute as the upper byte. The attribute specifies blink/no-blink and colors for the character and background.

The format of the attribute byte is:

Bit(s)	Definition
7	is a 1 to enable blinking and 0 to disable blinking.
6,5,4	is a 1 to enable the red/green/blue palette for the background or a 0 to disable this palette for the background.
3,2,1,0	is a 1 to enable the intensity/red/green/blue option for the foreground or a 0 to disable this option for the foreground.

The following list shows how the attribute specifications are implemented on the Mindset computer:

Attribute	Display on Mindset Computer
Blink set (= 1)	Character blinks
Foreground = black, Background = anything	Character is black, background is white
Foreground = color, Background = anything	Character is white, background is black
Intensity set (= 1)	Not implemented

Only two colors are displayed, so reverse video and blink are the only attributes shown on the screen. To display other colors, use the Mindset SET PALETTE command to redefine a color to be displayed instead of black (color 0 in the frame buffer) and white (color 1 in the frame buffer).

GRAPHICS MODE OPERATION

The Mindset computer supports the industry-standard 320- and 640-pixel-wide graphics modes. In addition, the Mindset computer offers a number of graphics features that go far beyond the industry standard. The extended capabilities of Mindset's display processor/graphics coprocessor combination include the following:

- Flexible graphics modes:
  - A 16-color, 320-pixel-wide display mode
  - A 4-color, 640-pixel-wide display mode
  - Double-buffered modes with 2 or 4 colors for 320 pixels or 2 colors for 640 pixels
  - Interlaced modes for doubled vertical resolution
- An interrupt to synchronize program activity with the screen refresh
- A screen image that the user can relocate within the border area
- The ability to synchronize with and mix display with an external video input
- Support for the industry-standard personal computer 320- and 640-pixel-wide graphics modes

In industry-standard graphics modes, the Mindset computer has only one display buffer and only one page. Mindset-unique graphics modes offer double buffering, which corresponds to two pages. The first page is numbered 0 and the second is numbered 1.

Characters can be drawn in all graphics modes. In modes where there is only one display page, the Mindset computer ignores page parameters related to industry-standard video commands. The cursor position determines character positioning. For character I/O routines in graphics modes, row means a character row (8 pixel lines) and column means a character column (8 pixels wide).

For pixel referencing, columns begin at 0 for the left-most column and go to either 639 or 319. Rows begin at 0 and end at 199 for all industry-standard compatible modes; only in Mindset 400-scan-line modes does the row number go to 399. The system changes any illegal value for a pixel row or column to the maximum allowable.

In industry-standard graphics modes, the frame buffer picture is stored with even scan lines and odd scan lines separated. In other words, all the even scan lines (0,2,4,...) are stored first, and then all the odd scan lines (1,3,5,...). In Mindset-native graphics modes, the frame buffer is organized the same way it appears on the screen, starting with scan line 0 and continuing through scan line 199 or 399. Graphics operation with the Mindset computer is described in detail in Section 4.

DESCRIPTIONS OF INDUSTRY-COMPATIBLE INTERRUPTS

The industry-compatible interrupts are:

Interrupt Name	Interrupt Number
Print Screen	05H
Video I/O	10H
Equipment check	11H
Memory size check	12H
Disk I/O	13H
RS-232-C I/O	14H
Keyboard I/O	16H
Printer I/O	17H
Cassette I/O*	18H
System bootstrap	19H
Time-of-day	1AH
Keyboard break	1BH
Timer tick	1CH
Video parameters	1DH
Diskette parameters	1EH
Video graphics characters	1FH

*Interrupt 18H leads only to a dummy IRET instruction in the Mindset computer.

The following sections describe each interrupt.

Print Screen -- Interrupt 05H

The Mindset print screen interrupt is identical to the industry-standard print screen interrupt. Invoking this interrupt causes the contents of the screen to be sent to the printer.

In character modes, the print screen interrupt sends the entire screen of characters to the printer.

In graphics modes, the print screen interrupt sends only characters that are drawn in some foreground color to the printer. This interrupt sends any other graphics or reverse-video characters to the printer as spaces.

Video I/O -- Interrupt 10H

The Mindset video I/O interrupt is a software interrupt entry point identical in function to the industry-standard video I/O entry point. To apply this entry point, the user places a function code in register AH and other parameters in other registers as described subsequently, and calls interrupt 10H. Except where noted otherwise, all functions described below work the same in Mindset display modes as they do in industry-standard modes.

The functions provided by the video I/O interrupt are:

Command	Value of AH Decimal	Hex	Function
Set Mode	00	00	Specifies the video display mode.
Set Cursor Type	01	01	Specifies the appearance of the cursor.
Set Cursor Position	02	02	Specifies the cursor location.
Read Cursor Position	03	03	Returns the current cursor position.
Reserved	04	04	N/A
Select Active Display Page	05	05	Specifies the page to be displayed.
Scroll Active Page Up	06	06	Specifies upward movement of a part of the displayed page.
Scroll Active Page Down	07	07	Specifies downward movement of a part of the displayed page.
Read Attribute/ Character	08	08	Returns the attribute/character at the current cursor position.

Command	Value of AH Decimal	Hex	Function
Write Attribute/ Character	09	09	Writes the specified attribute/ character to the current cursor position.
Write Character Only	10	0A	Writes a character only to the current cursor position.
Set Color Palette	11	0B	Specifies the color palette for use in industry-standard display modes.
Write Dot	12	0C	Writes a single pixel with a specified color at a specified position.
Read Dot	13	0D	Returns the color of a specified pixel.
Write Teletype	14	0E	Writes a teletype (TTY) character to the display.
Current Video State	15	0F	Returns the current mode, number of character columns, and active display page.

All registers except those used to return values remain unchanged after a call.

Descriptions of the individual video I/O commands follow.

SET MODE -- Int. 10H, (AH) = 00H

Function: Specifies the video display mode.

Description: The SET MODE video I/O command selects one of eight display modes for use.

In graphics modes, the display processor initializes the display buffer by filling it with 0s. In character modes, the display processor fills the display buffer with spaces and attribute value 7 (white on black), sets the display page to 0, and initializes the cursor to lines 6 through 7 (see the SET CURSOR TYPE function, described next).

Input Parameters: (AL) is the mode selected. The modes are:

	Mode	Specification	Color or Black-and-White (B&W)
Character Modes	0	40 x 25	B&W
	1	40 x 25	Color (2 colors)
	2	80 x 25	B&W
	3	80 x 25	Color (2 colors)
Graphics Modes	4	320 x 200	Color (4 colors)
	5	320 x 200	B&W
	6	640 x 200	B&W
CRT mode	7	80 x 25	B&W

Output Parameters: None.

SET CURSOR TYPE -- Int. 10H, (AH) = 01H

Function: Specifies the height of the cursor.

Description: The SET CURSOR TYPE video I/O command specifies the appearance of the cursor, which appears in character modes only. The cursor has a fixed width (one character wide) but variable beginning and end lines that are set with this command. For example, if you place the start and end lines at the same position, the cursor appears as a blinking horizontal line. If you put the start line at 0 and set the end line at 7, the cursor appears as a blinking box the height and width of a standard video character. (Line 0 is the top line and line 7 is the bottom line of the character position.)

The cursor blink function cannot be altered.

The system saves the cursor parameters in the BIOS data area. The parameters can be returned using the CURRENT VIDEO STATE or READ CURSOR POSITION command.

Input Parameters:

(CH) is the cursor start line, 0 to 15 decimal. Using a number greater than 7 results in no cursor display.

(CL) is the cursor end line, 0 to 15 decimal. Using a number greater than 7 results in no cursor display.

Output Parameters: None.

SET CURSOR POSITION -- Int. 10H, (AH) = 02H

Function: Specifies the cursor location.

Description: The SET CURSOR POSITION video I/O command specifies the row, column, and page number for the cursor. A (row, column) value of (0,0) indicates the upper left corner.

Input Parameters:

(BH) is the page number. In all modes, the display processor masks (BH) to a legal page value.

(DH) is the row number, automatically limited by the SET CURSOR POSITION routine to the maximum for the current mode.

(DL) is the column number, automatically limited by the SET CURSOR POSITION routine to the maximum for the current mode.

Output Parameters: None.

READ CURSOR POSITION -- Int. 10H, (AH) = 03H

Function: Returns the current cursor position.

Description: The READ CURSOR POSITION video I/O command returns the current row number, column number, start line, and end line for the cursor.

Input Parameters:

(BH)	is the page number, ignored if currently in a mode with only one page. In all modes, the display processor masks (BH) to a legal value.

Output Parameters:

(CH)	is the cursor start line.
(CL)	is the cursor end line.
(DH)	is the row number of the cursor.
(DL)	is the column number of the cursor.

SELECT ACTIVE DISPLAY PAGE -- Int. 10H, (AH) = 05H

Function: Specifies the page to be displayed.

Description: The SELECT ACTIVE DISPLAY PAGE video I/O command selects one of up to 2 pages (modes 2 and 3) or 4 pages (modes 0 and 1) for display in character modes.

In Mindset-native graphics modes that allow double buffering, the SELECT ACTIVE DISPLAY PAGE command does not have the effect of the SWITCH ACTIVE BUFFER command--it cannot switch the active buffer for display. The SELECT ACTIVE DISPLAY PAGE routine does, however, select the active page for other video I/O commands such as SCROLL ACTIVE PAGE UP, WRITE DOT, and READ DOT.

Input Parameter:

(AL) is the new page number: 0 or 1 for Mindset-native graphics modes, 0 to 3 for industry-standard modes 0 and 1, or 0 to 1 for industry-standard modes 2 and 3. (AL) has no effect in graphics modes with only 1 page. The SELECT ACTIVE DISPLAY PAGE routine masks (AL) to a legal value.

Output Parameters: None.

SCROLL ACTIVE PAGE UP -- Int. 10H, (AH) = 06H

Function: Specifies upward movement of a part of the displayed page.

Description: The SCROLL ACTIVE PAGE UP video I/O command specifies the number of lines, the upper left corner, the lower right corner, and the blank line attribute for use in the upward scroll procedure.

The upward scroll moves the specified number of lines up within the specified rectangular area in the active page and fills the area at the bottom with spaces having a specified attribute (in character modes), or with a specified fill byte (in graphics modes).

Input Parameters:

(AL) is the number of lines to be moved at the top of the rectangular region (window) specified in (CH, CL) and (DH, DL) below, with blank line replacements at the bottom. A value of 0 for (AL) causes the active window to be filled with blank lines.

(BH) is the attribute (in character modes) or fill byte (in graphics modes) used to fill the blank lines that enter at the bottom of the scroll window.

(CH) is the row of the upper left corner of the scroll window.

(CL) is the column of the upper left corner of the scroll window.

(DH) is the row of the lower right corner of the scroll window.

(DL) is the column of the lower right corner of the scroll window.

Output Parameters: None.

SCROLL ACTIVE PAGE DOWN -- Int. 10H, (AH) = 07H

Function: Specifies a downward movement of a part of the displayed page.

Description: The SCROLL ACTIVE PAGE DOWN video I/O command specifies the number of lines, the upper left corner, the lower right corner, and the blank line attribute for use in the downward scroll procedure.

The downward scroll moves the specified number of lines down within the specified rectangular area in the active page and fills the area at the top with spaces having a specified attribute (in character modes), or with a specified fill byte (in graphics modes).

Input Parameters:

(AL) is the number of lines to be moved at the bottom of the rectangular region (window) specified in (CH, CL) and (DH, DL) below, with blank line replacements at the top. A value of 0 for (AL) causes the active window to be blanked.

(BH) is the attribute (in character modes) or fill byte (in graphics modes) used to fill the blank lines that enter at the top of the scroll window.

(CH) is the row of the upper left corner of the scroll window.

(CL) is the column of the upper left corner of the scroll window.

(DH) is the row of the lower right corner of the scroll window.

(DL) is the column of the lower right corner of the scroll window.

Output Parameters: None.

READ ATTRIBUTE/CHARACTER -- Int. 10H, (AH) = 08H

Function: Returns the attribute/character at the current cursor position.

Description: In character modes, the READ ATTRIBUTE/CHARACTER video I/O command returns the character and attribute, corresponding to the current cursor position, which are in the ASCII/attribute buffer. Input for this command is the page number of the current display.

In graphics modes, the READ ATTRIBUTE/CHARACTER command compares the contents of the graphics buffer to the 8 x 8 pixel character representations stored in either BIOS ROM or user-defined bitmaps. The character may consist of any combination of foreground colors on a background color. The first character that matches the contents of the graphics buffer is returned in AL. The character search starts with character 0.

The READ ATTRIBUTE/CHARACTER routine returns no attribute in graphics modes.

Input Parameter:

(BH) is the display page number (valid for character modes and some Mindset-native graphics modes).

Output Parameter:

(AH) is the attribute of the character at the current cursor position (valid for character modes and some Mindset-native graphics modes).

(AL) is the character at the current cursor position (or 0 in graphics modes if no match is found).

WRITE ATTRIBUTE/CHARACTER -- Int. 10H, (AH) = 09H

Function: Writes an attribute/character to the current cursor position.

Description: In character modes, the WRITE ATTRIBUTE/CHARACTER video I/O command puts the specified attribute and character into the ASCII/attribute buffer.

In graphics modes, BL specifies the color. If bit 7 of BL is set, the character will be exclusive-ORed into the frame buffer. The WRITE ATTRIBUTE/CHARACTER command draws the character(s) from an 8 x 8 pixel representation stored in BIOS ROM, or from a user-defined alternate character set if the character has a numerical equivalent greater than 127.

In 400-pixel line graphics modes, the character is drawn as 8 pixels wide by 16 pixels high so that it looks the same as the characters in other modes.

Input Parameters:

(AL) is the character this command is to write.

(BH) is the display page number (valid for character modes and some Mindset-native graphics modes).

(BL) is the attribute of the character to be written (in character modes) or the color of the character (in graphics modes). If bit 7 of BL is 1 in graphics modes, then the color value specified is exclusive-ORed with the current color of the cursor position.

(CX) is the number of times the character is to be written. (All characters must remain on the same row.)

Output Parameters: None.

WRITE CHARACTER ONLY -- Int. 10H, (AH) = 0AH

Function: Writes a character to the current cursor position.

Description: The WRITE CHARACTER ONLY video I/O command displays the specified character at the current cursor position.

In graphics modes, the WRITE CHARACTER ONLY command is equivalent to the WRITE ATTRIBUTE/CHARACTER command. BL must specify a color for the character.

In character modes, the WRITE CHARACTER ONLY command replaces the character only, leaving the corresponding attribute unchanged.

Input Parameters:

(AL) is the character this command is to write.

(BH) is the display page number (valid for character modes and some Mindset-native graphics modes).

(BL) is the color of the character (for graphics modes only). If bit 7 of BL is 1, then the color value specified is exclusive-ORed with the current color of the cursor position.

(CX) is the number of times the character is to be written. (All characters must remain on the same row.)

Output Parameters: None.

SET COLOR PALETTE -- Int. 10H, (AH) = 0BH

Function: Specifies the color palette for use in industry-standard display modes.

Description: The SET COLOR PALETTE video I/O command specifies the foreground, background, and/or border colors, or the color palette, for use in industry-standard graphics and character modes. The result of the SET COLOR PALETTE depends on the values placed in BH and BL, as described below. This command is not flexible or useful as the Mindset-unique SET PALETTE command; it only provides industry-standard compatibility.

Input Parameters:

(BH) is the identification (ID) number of the palette color to be set. For graphics modes, an ID number of 0 indicates the background and border color and an ID number of 1 selects a foreground color. In character modes, an ID number of 0 indicates the border color (an ID number of 1 is not used).

(BL) is the color value for the foreground, background, and/or border as determined by the identification number in BH and the current display mode. The color value placed in BL when (BH) = 0 must follow the intensity/red/green/blue format described under Color Palette Control in Section 4, "Mindset-Native Graphics Operation". When (BH) = 1, (BL) specifies one of two color palettes.

(BL) specifies the foreground, background, and/or border colors or the foreground color palette as follows: When (BH) = 0, (BL) specifies the border color for character modes 0, 1, 2, and 3, the background and border color for industry-standard graphics modes 4 and 5, and the foreground color for industry-standard graphics modes 6. When (BH) = 1, (BL) specifies the foreground color palette for industry-standard graphics modes 4 and 5 only. In this case, (BL) = 0 specifies the red/green/yellow palette and (BL) = 1 specifies the cyan/magenta/white palette.

Output Parameters: None.

WRITE DOT -- Int. 10H, (AH) = 0CH

Function: Writes a single pixel with a specified color at a specified position.

Description: The WRITE DOT video I/O command displays a dot (single pixel) with the specified color at the specified location on the active page.

Input Parameters:

(AL) is the desired color value. If bit 7 of AL is a 1, then the color value is exclusive-ORed with the current color at the pixel specified by (CX) and (DX).

(CX) is the column number.

(DX) is the row number.

Output Parameters: None.

READ DOT -- Int. 10H, (AH) = 0DH

Function: Returns the color of a specified pixel.

Description: The READ DOT video I/O command returns the color value of a pixel specified by row and column numbers. This command has no effect in character modes.

Input Parameters:

	(CX)	is the column number.
	(DX)	is the row number.

Output Parameters:

	(AL)	is the color value of the pixel.

WRITE TELETYPE -- Int. 10H, (AH) = 0EH

Function: Writes a teletype (TTY) character to the display.

Description: The WRITE TELETYPE video I/O command writes a specified character to a specified display page using the WRITE CHARACTER ONLY routine and then updates the cursor position.

If the cursor position was at the right-most column on the page, the WRITE TELETYPE routine sends the character (and the cursor position) to the next line.

The WRITE TELETYPE command implements four control characters (carriage return, line feed, bell, and backspace) as follows:

1. A carriage return returns the cursor to column 0 of the current line.

2. A line feed sends the cursor to the next line, remaining in the current column, and scrolls the entire display page up if necessary.

3. A bell causes the beeper to beep if the beeper is enabled and causes the custom sound-processor (CSP) to emit a tone as specified by the CSP sound registers.

4. A backspace moves the cursor back one column with no erase of the character, unless the cursor is already at column 0. In column 0, no backspacing is possible.

If the character is a line feed, and the cursor is already on the last line of the page, the routine scrolls the page up one line and fills the new line with blanks (in character modes) or with zeroes (in graphics modes).

As implemented on industry-standard personal computers, this call is limited to operating on the active page. On the Mindset computer, the call works for any valid page number passed in BH.

Input Parameters:

(AL) is the character to be written.

(BH) is the display page number (valid for character modes and some Mindset-native graphics modes).

(BL) is the foreground color in graphics modes. If bit 7 is 1, the WRITE TELETYPE routine exclusive-ORs the character color with the background color.

Output Parameters: None.

CURRENT VIDEO STATE -- Int. 10H, (AH) = 0FH

Function: Returns the status of the current video state.

Description: The CURRENT VIDEO STATE video I/O command returns the current mode, number of character columns, and active display page.

Input Parameters: None.

Output Parameters:

(AL) is the current mode. (See the SET MODE video I/O command for a description of video modes.) If the current mode is not an industry-standard mode, the CURRENT VIDEO STATE routine returns FFH in the AL register.

(AH) is the number of character columns on the screen.

(BH) is the active display page.

Equipment Check -- Interrupt 11H

The equipment check interrupt returns data that indicates whether optional devices are attached to the system. This interrupt returns information on the number of printers attached, whether or not a game I/O device is attached, the number of RS-232-C modules attached, the number of diskette drives, the initial video mode, and the RAM size.

Input Parameters: None.

Output Parameters: (AX) returns status bits as follows:

Bit(s)	Definition
0	is a 1 if one or more disk drives are attached to the system, or is 0 if no disk drives are attached.
1	is unused.
3,2	indicate the system board RAM size as follows:

Bit Values	RAM Size
0,0	16K bytes
0,1	32K bytes
1,0	48K bytes
1,1	64K bytes

5,4 indicate the initial video mode:

Bit Values	Video Mode
0,0	(unused)
0,1	40 x 25 B&W using the color card
1,0	80 x 25 B&W using the color card
1,1	80 x 25 B&W using the B&W card

Bit(s)	Definition
7,6	indicate the number of disk drives (provided that bit 0 is a 1):

Bit Values	Number of Drives
0,0	1
0,1	2
1,0	3
1,1	4

8	is unused.
11,10,9	represent the number of RS-232-C modules attached to the system (in binary).
12	is a 1 if a game I/O device is attached to the system or a 0 if otherwise.
13	is unused.
15,14	represent the number of printers attached to the system (in binary).

Memory Size Check -- Interrupt 12H

The memory size check interrupt returns the number of contiguous 1K-byte blocks of memory in the system. The system memory includes the memory in the main System Unit and in the Expansion Unit. The memory size returned does not include RAM in any installed cartridges.

Input
Parameters: None.

Output
Parameters: (AX) is the number of contiguous 1K-byte blocks of RAM.

Disk I/O -- Interrupt 13H

The disk I/O interrupt provides six functions relating to disk operations that are selected by the value in register AH when the interrupt is called. The six functions are:

Command	Value of AH Decimal	Hex	Function
Reset Disk System	00	00	Resets the disk drive.
Read Disk System Status	01	01	Returns the disk drive status.
Read Sectors	02	02	Retrieves sectors of data.
Write Sectors	03	03	Writes sectors to disk.
Verify Sectors	04	04	Verifies sectors of disk storage.
Format Track	05	05	Formats a disk track.

The RESET DISK SYSTEM and READ DISK SYSTEM STATUS commands have no input parameters and return status bits in register AH. The status values and corresponding meanings are:

Name	Value	Meaning
(none)	00H	Successful operation
Bad Command	01H	Bad command passed to disk I/O
Bad Addr Mark	02H	Address mark not found
Write Protect	03H	Write attempted on a write-protected disk
Record Not Found	04H	Requested sector not found
Bad DMA	08H	DMA (direct memory access) overrun on operation
DMA Boundary	09H	Attempt to perform a DMA across a 64K boundary
Bad CRC	10H	Bad CRC (cycle redundancy check) on disk read
Bad Printer	20H	Printer interface failed
Bad Seek	40H	Seek operation failed
Time Out	50H	Attachment (disk drive) failed to respond

Also, register CY contains 0 after a successful disk drive operation (register AH is 0) or 1 after a failed disk drive operation (AH has an error reason as described above).

The READ SECTORS and WRITE SECTORS commands transfer data from disk to memory and memory to disk respectively. The VERIFY SECTORS command checks the parity of data in the specified sectors of the disk. The FORMAT TRACK command writes format information on the disk. This command requires that the buffer address (ES:[BX]) point to the collection of desired address fields for the track. An address field consists of four bytes: the track number (C), the head number (H), the sector number (R), and the number of bytes per sector (N), where

0 indicates 128 bytes/sector, 1 indicates 256 bytes/sector, 2 indicates 512 bytes/sector, and 3 indicates 1024 bytes/sector. There must be an address field for every sector on the track; the address field information is used to find the requested sector during read/write accesses.

For the READ SECTORS, WRITE SECTORS, VERIFY SECTORS, and FORMAT TRACK commands, the input parameters are:

(AL) is the number of sectors to read, write, verify, or format. (The value of this parameter is not checked before use.)

(CH) is the track number, 0 to 39. (Value not checked.)

(CL) is the sector number. (Value not checked.)

(DH) is the head number, 0 or 1. (Value not checked.)

(DL) is the disk drive number, 0 or 1. The value of this parameter is checked when used.

(ES:[BX]) is the buffer address (not required for the VERIFY SECTORS command).

RS-232-C I/O -- Interrupt 14H

The RS-232-C I/O interrupt consists of four commands selected by the value in register AH. The four commands are:

Command	Value of AH Decimal	Hex	Function
Initialize Com Port	00	00	Specifies the baud rate, parity, stop bit, and number of data bits per character for the communications port.
Send Character	01	01	Sends the character in AL over the communications port.
Receive Character	02	02	Reads a character from the communications port into AL.
Get Com Port Status	03	03	Returns the status of the communications port.

The commands are described individually on the following pages.

INITIALIZE COM PORT -- Int. 14H, (AH) = 00H

Function: Specifies initialization parameters for a selected communications port.

Description: The INITIALIZE COM PORT RS-232-C I/O command specifies the baud rate, parity, stop bit, and number of data bits per character for the communications port.

Input Parameters: (AL) specifies the parameters according to the following bit definitions:

Bit(s)	Definition

7,6,5 select the baud rate as follows:

Bit Values	Baud Rate
0,0,0	110 baud
0,0,1	150 baud
0,1,0	300 baud
0,1,1	600 baud
1,0,0	1200 baud
1,0,1	2400 baud
1,1,0	4800 baud
1,1,1	9600 baud

4,3 select the parity:

Bit Values	Parity
0,0 or 1,0	none
0,1	odd
1,1	even

2 selects the number of stop bits:

Bit Value	Stop Bits
0	1 stop bit
1	2 stop bits

1,0 select the number of data bits per character:

Bit Value	Data Bits/Character
1,0	7 bits
1,1	8 bits

(DX) selects an RS-232-C module for use (0, 1, 2, or 3).

Output
Parameters: (AX) holds the communications port status (see the GET COM PORT STATUS command later in this section for a description of the status bits).

All other registers remain unchanged.

SEND CHARACTER -- Int. 14H, (AH) = 01H

Function: Sends the character in register AL to the communications port.

Description: The SEND CHARACTER RS-232-C I/O command transmits a character to the selected communications port.

Before sending the character, the SEND CHARACTER routine sets the data-terminal-ready (DTR) and request-to-send (RTS) RS-232-C status lines. If the receiving device (at the far end of the RS-232-C connection) responds with the data-set-ready (DSR) and clear-to-send (CTS) status signals within approximately 500 milliseconds, SEND CHARACTER sends the character. If the receiving device does not issue the DSR and CTS signals within approximately 500 milliseconds, the SEND CHARACTER routine sets the timeout-error status bit and does not send the character.

Input Parameters:

(AL) is the character to be sent.

(DX) indicates which RS-232-C module to use (0, 1, 2, or 3).

Output Parameters:

(AH) is the status of the operation. Bit 7 is set if the SEND CHARACTER routine was unable to send the byte over the line. The rest of the AH bits are set as in a GET COM PORT STATUS call (see the GET COM PORT STATUS command later in this section).

(AL) is preserved.

All other registers remain unchanged.

RECEIVE CHARACTER -- Int. 14H, (AH) = 02H

Function: Reads a character from the selected communications port.

Description: The RECEIVE CHARACTER RS-232-C I/O command reads a character from the communications port and places it in register AL.

Initially, the RECEIVE CHARACTER routine sets the data-terminal-ready (DTR) RS-232-C status line. If the device at the far end of the RS-232-C connection responds with a data-set-ready (DSR) status signal within approximately 500 milliseconds, the RECEIVE CHARACTER routine waits (if necessary) until it can read a character from the communications port. If the far-end device does not send the DSR signal within approximately 500 milliseconds, the routine sets the timeout-error status bit and does not read a character.

Input Parameters:

(DX) indicates which RS-232-C module to use (0, 1, 2, or 3).

Output Parameters:

(AH) is the current line status, as set by the GET COM PORT STATUS command, except that only the error bits (7, 4, 3, 2, 1) of the status word are affected. Therefore, (AH) is nonzero after a RECEIVE CHARACTER command call only if an error occurred. Also, in this case, if the timeout bit (bit 7) is set, it means that the data-set-ready signal was not received.

(AL) is the character byte received.

All other registers remain unchanged.

GET COM PORT STATUS -- Int. 14H, (AH) = 03H

Function: Returns the status of the selected communications port.

Description: The GET COM PORT STATUS RS-232-C I/O command returns the status of the selected communications port.

The GET COM PORT STATUS routine clears the error and status conditions after placing the corresponding bits in AH and AL.

Input Parameters: (DX) indicates which RS-232-C module to use (0, 1, 2, or 3).

Output Parameters: (AH) contains the line control status:

Bit	Meaning (if set)
0	Data ready
1	Overrun error occurred
2	Parity error occurred
3	Framing error occurred
4	Break occurred
5	Transmission holding register is empty
6	Transmission shift register is empty
7	Timeout occurred

(AL) holds the modem status:

Bit	Meaning (if set)
0	Clear-to-send (CTS) changed since the last time it was read
1	Data-set-ready (DSR) changed since the last time it was read
2	Trailing edge of ring detected
3	Receive line signal changed since the last time it was read
4	Clear to send
5	Data set ready
6	Ring indicated
7	Received line signal

All other registers remain unchanged.

Keyboard I/O -- Interrupt 16H

The keyboard I/O interrupt provides three commands for keyboard I/O. The commands are:

Command	Value of AH Decimal	Hex	Function
Read Next Char	00	00	Reads the next character struck on the keyboard.
Keyboard Status	01	01	Indicates if an ASCII character is available to be read.
Get Shift Status	02	02	Returns the current shift status.

The commands are described individually on the following pages.

READ NEXT CHAR -- Int. 16H, (AH) = 00H

Function: Reads the next character struck on the keyboard.

Description: The READ NEXT CHAR keyboard I/O command returns the next ASCII character and scan code received from the keyboard.

Input Parameters: None.

Output Parameters:

(AH) is the scan code.

(AL) is the character.

All other registers remain unchanged.

KEYBOARD STATUS -- Int. 16H, (AH) = 01H

Function: Indicates if an ASCII character is available to be read.

Description: The KEYBOARD STATUS keyboard I/O command specifies the Z flag to indicate whether a scan code from the keyboard is available. If a scan code is available, the KEYBOARD STATUS routine copies the corresponding character into (AX), leaving the original character in the buffer.

Input Parameters: None.

Output Parameters:

(AX) is as described above.

(ZF) is a 1 if no code is available or a 0 if a scan code is available.

All other registers remain unchanged.

GET SHIFT STATUS -- Int. 16H, (AH) = 02H

Function: Returns the current shift status.

Description: The GET SHIFT STATUS keyboard I/O command returns the current shift status of the keyboard.

Input Parameters: None.

Output Parameter: (AL) is the current shift status as follows:

Bit(s)	Name	Meaning (if 1)
0,1,2	--	(unused)
3	Hold State	The PAUSE key has been toggled.
4	Scroll Shift	The SCROLL LOCK key is depressed.
5	Num Shift	The NUM LOCK key is depressed.
6	Caps Shift	The CAPS LOCK key is depressed.
7	Ins Shift	The INSERT key is depressed.

All other registers remain unchanged.

Printer I/O -- Interrupt 17H

The printer I/O interrupt provides three commands for communications with the printer. The commands are:

Command	Value of AH Decimal	Hex	Function
Print Character	00	00	Sends a character to the printer.
Init Printer Port	01	01	Opens the printer port.
Get Printer Status	02	02	Returns printer status bits.

Three printers can be used (numbered 0, 1, or 2). The base address for each printer module must be set in the printer base data area.

The commands are described individually on the following pages.

PRINT CHARACTER -- Int. 17H, (AH) = 00H

Function: Sends a character to the printer.

Description: The PRINT CHARACTER printer I/O command sends a character to the printer.

Input Parameters:

(AL) is the character to be printed.

(DX) is the printer to be used (0, 1, or 2).

Output Parameters:

(AH) has a value of 1 if the character was not printed due to a timeout. Otherwise, AH has one of the following status values:

Bit(s)	Meaning (if 1)
0	Timeout occurred (printer did not respond).
1,2	(unused).
3	I/O error occurred.
4	Printer is selected.
5	Out of paper.
6	Acknowledge. (0 means data has been received and the printer is ready to accept other data.)
7	Not busy, printer can receive data. (0 means the printer is busy and cannot receive data.)

All other registers remain unchanged.

INIT PRINTER PORT -- Int. 17H, (AH) = 01H

Function: Opens the printer port.

Description: The INIT PRINTER PORT printer I/O command opens the printer port and returns the printer status.

Input Parameters: (DX) is the printer to be used (0, 1, or 2).

Output Parameters: (AH) is one of the following printer status bits:

Bit(s)	Meaning (if 1)
0	Timeout occurred (printer did not respond).
1,2	(unused).
3	I/O error occurred.
4	Printer is selected.
5	Out of paper.
6	Acknowledge. (0 means data has been received and the printer is ready to accept other data.)
7	Not busy, printer can receive data. (0 means the printer is busy and cannot receive data.)

All other registers remain unchanged.

GET PRINTER STATUS -- Int. 17H, (AH) = 02H

Function: Returns printer status bits.

Description: The GET PRINTER STATUS printer I/O command returns status bits for the printer.

Input Parameters: (DX) is the printer to be used (0, 1, or 2).

Output Parameters: (AH) is the printer status as follows:

Bit(s)	Meaning (if 1)
0	Timeout occurred (printer did not respond).
1,2	(unused).
3	I/O error occurred.
4	Printer is selected.
5	Out of paper.
6	Acknowledge. (0 means data has been received and the printer is ready to accept other data.)
7	Not busy, printer can receive data. (0 means the printer is busy and cannot receive data.)

All other registers remain unchanged.

System Bootstrap -- Interrupt 19H

The system bootstrap interrupt starts the system by loading track 0, sector 1 of a disk, if a disk and disk drive are available, or by transferring control to a ROM or ROM/RAM cartridge if a disk and disk drive are not available. If a hardware error occurs while the system is trying to read the disk, the system bootstrap routine transfers control to a ROM or ROM/RAM cartridge. If no suitable ROM or ROM/RAM cartridge is installed, the system displays the rotating Mindset logo.

Note also that the Mindset computer may try to boot from a ROM or ROM/RAM cartridge before it tries to boot from a disk, depending on the priorities set on the system configuration screen.

There are no input or output parameters for this interrupt.

Time of Day -- Interrupt 1AH

The time-of-day interrupt provides two commands to read and set the time-of-day clock. An input value of 0 for AH, which invokes the READ T.O.D. CLOCK command, returns the high portion of the count (hours) in CX and the low portion of the count (minutes) in DX. READ T.O.D. CLOCK also returns 0 in AL if 24 hours have not passed since the last time the clock was read, or it returns a nonzero value in AL if 24 hours have passed since the last time the T.O.D. clock was read.

An input value of 1 for AH invokes the SET T.O.D CLOCK command. The high portion of the count goes in CX and the low portion of the count goes in DX.

Keyboard Break -- Interrupt 1BH

This interrupt is identical to the corresponding industry-standard interrupt.

Timer Tick -- Interrupt 1CH

This interrupt is identical to the corresponding industry-standard interrupt.

Video Parameters -- Interrupt 1DH

This interrupt is identical to the corresponding industry-standard interrupt.

Diskette Parameters -- Interrupt 1EH

The diskette parameters interrupt is a pointer to a group of parameters required for diskette operation. To modify the parameters, you must make a new parameter block and reference it separately.

Video Graphics Characters -- Interrupt 1FH

The video graphics characters interrupt provides a way to reference user-defined character sets in graphics modes. To use this interrupt, the user must supply a vector to the desired block of character data in this interrupt location. A user-defined character set goes beyond the standard ROM-supplied 127-character set to define characters 128 through 255.

Each character in the user-defined set consists of a bitmap that specifies the 64 pixels in the 8 x 8 matrix that constitutes a character block.

In all graphics modes, the character bitmaps must be stored as 8 x 8 bit arrays, even if the mode calls for more than 1 bit per pixel. If these bitmaps are not supplied, the standard character bitmaps stored in the BIOS ROM are used to draw the characters. In character modes, any user-defined set of characters is ignored.

An industry-standard personal computer displays garbage for any character between 128 and 255 in graphics modes if a character set pointer is not supplied. The Mindset computer, by contrast, displays characters from its own extra character set if no pointer is supplied. READ-CHARACTER-type calls also employ the user-defined character set, if any, to match bit patterns.

HARDWARE INTERRUPT PRIORITIES

Assigning the proper priorities to interrupts is important in achieving tolerable interrupt latency and response times. Table 3-1 illustrates the hardware interrupt priorities in the Mindset Personal Computer system, where 0 is the highest priority and 7 is the lowest priority.

Table 3-1: Hardware Interrupt Priorities

Priority Level	Interrupt Name	Processing Time	Maximum Tolerable Latency
0	Unassigned		
1	Refresh Interrupt	50 µsec	500 µsec
2	Keyboard/GCP	200 µsec	1500 µsec
3	Module Interrupt	100 µsec	100 msec
4	Timer 0 Interrupt	100 µsec	50 msec
5	Late VBLANK Interrupt	50 µsec	60 msec
6	Display Interrupt	5000 µsec	60 msec
7	Unassigned		

Section 4

MINDSET-NATIVE GRAPHICS OPERATION

OVERVIEW OF GRAPHICS OPERATION

The graphics operation of the Mindset Personal Computer provides many extended features. Powerful graphics commands, resident in the Mindset ROM BIOS, simplify the programming of multilayered, multicolored graphics images.

The graphics mode operation of the Mindset computer uses pixels, scan lines, and optional interlacing to display objects on the screen.

A pixel (picture element) is the smallest addressable screen element in graphics mode. The number of bits in a pixel definition determines the number of colors available in the current graphics mode. A pixel may be represented by 1, 2, or 4 bits, which provide 2, 4, or 16 distinct colors, respectively.

Two Processors

Two separate processors are used to generate graphics displays in the Mindset computer. The two processors are the display processor and the graphics coprocessor (GCP). The display processor uses the data in the active frame buffer to generate the screen display. (Frame buffers are described later in this section.) The GCP can access all memory to produce and manipulate graphics images. Thus the GCP prepares graphics images, offering many high-level operational commands, and the display processor maps the images to the screen.

The extended capabilities of the display processor/GCP combination include the following:

- Flexible graphics modes:
  - A 16-color, 320-pixel-wide display mode
  - A 4-color, 640-pixel-wide display mode
  - Double-buffered modes with 2 or 4 colors for 320 pixels or 2 colors for 640 pixels
  - Interlaced modes for doubled vertical resolution
- An interrupt to synchronize program activity with the screen refresh
- A screen image that the user can relocate within the border area
- The ability to synchronize with and mix display with an external video input
- Support for the industry-standard personal computer 320- and 640-pixel-wide graphics modes

The GCP and ROM BIOS of the Mindset computer also enable the user to perform efficient block transfers (BLTs) of objects anywhere within memory. In addition to block transfers, the graphics commands of the ROM BIOS provide points, lines, filled polygons, and filled and hollow ellipses and circles.

Screen Orientation

In all references in this section, the pixel at the upper left corner of a rectangular region is location (0,0). Distances measured to points below and to the right of (0,0) are positive values. For example, the pixel at location (5,3) is 5 pixels to the right and 3 pixels below location (0,0).

DISPLAY PROCESSOR OPERATION

This section has two parts. The first part discusses display processor operations and capabilities; the second part describes the corresponding commands in detail.

Screen Format Control

The Mindset computer provides seven graphics modes that vary in screen resolution, number of colors available, single or double buffering, and interlaced or non-interlaced operation. You can specify an RGB monitor or a color television as the display device. You can position the active screen within the borders on the display screen.

Graphics Display Resolution. A scan line is a horizontal line of pixels. The graphics mode resolution describes the number of pixels per scan line and the number of scan lines on the screen. (For example, 640 x 200 means that there are 640 addressable pixels on each of 200 scan lines on the screen.)

The Mindset computer offers 320 x 200 or 640 x 200 resolution for non-interlaced graphics operation and 320 x 400 or 640 x 400 resolution for interlaced graphics operation.

Interlaced and Non-Interlaced Display. Interlacing is a technique that doubles the vertical resolution of the screen from 200 to 400 scan lines. In interlaced modes, the system draws the odd-numbered scan lines the first time it writes the screen and then "interlaces" the even-numbered scan lines between the odd-numbered scan lines the next time it writes the screen. Thus, interlaced modes use twice as many scan lines as non-interlaced modes, and the gun makes two top-to-bottom passes to write an entire screen.

Interlacing requires that the phosphor in the cathode ray tube (CRT) of the display device has enough persistence to retain the pixels on the odd-numbered scan lines while the system writes the even-numbered scan lines.

The highest-resolution modes use interlacing to double the vertical resolution of the non-interlaced modes. The interlaced modes are for televisions and composite video monitors only.

Screen Format Commands. The commands for specifying screen format are listed below and described more fully later in this Section.

Command Interrupt	Value of AH Decimal	Hex	Function
Set Screen Mode (EFH)	00	00	Selects graphics resolution, number of colors, single or double buffering, and interlaced or non-interlaced operation.
Get Screen Mode (EFH)	01	01	Returns the current screen mode parameters.
Set Display Device (EEH)	02	02	Sets the display for either a television or an RGB monitor and reloads the color palette. Also enables or disables color output to a television display.
Set Screen Position (EEH)	03	03	Sets the screen position within the screen border.
Get Screen Position (EEH)	04	04	Returns the current position of the screen within the screen border.

Double Buffering

The area of memory that the display processor displays on the screen is called a frame buffer. A double-buffering feature is available in certain display modes. Double buffering enables the user to display from either of two frame buffers.

The ROM BIOS of the Mindset computer system provides many commands that produce graphics images. The user can, however, write directly to the screen buffer. The ROM BIOS command GET BUFFER INFO returns the addresses of the frame buffers in native graphics modes so that the user can write information directly into them.

The ROM BIOS command GET SCREEN MODE returns a flag to indicate which frame buffer is active. When using double buffering, the SWITCH ACTIVE BUFFER command instructs the system to activate the hidden frame buffer and hide the active frame buffer.

The commands associated with double buffering are listed below and described more fully later in this section.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Get Buffer Info (EFH)	14	0E	Returns the addresses and sizes of the system frame buffers.
Switch Active Buffer (EFH)	17	11	Switches to the alternate frame buffer if double buffering is possible.

Color Palette Control

The Mindset computer uses a color index value to specify the color placed on the screen. The color index is an offset into a lookup table of 16 user-defined colors; this table is called the color palette. The user should define the color palette to obtain the desired colors for the selected mode.

The color palette specifies the actual color corresponding to each numeric color value. Each 16-bit word in the color palette specifies the content of red, green, and blue (RGB) for a single color. The 4 most significant bits of each word define the color for an RGB color monitor, while the 9 least significant bits define the color for a color television (see Figure 4-1).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Device:	M	M	M	M	U	U	T	T	T	T	T	T	T	T	T	T
Color (Key):	I	R	G	B	U	U	K	B	B	B	G	G	G	R	R	R

M = a bit used to define a monitor color
T = a bit used to define a television color
R = red
G = green
B = blue
I = intensity
K = key onto external video
U = unused

Figure 4-1: Definition of bits within a color palette entry

When using an RGB color monitor, the system uses bits 12, 13, and 14 to define the blue, green, and red components of a color and bit 15 to define high or low intensity for that color. For example, a value of 0000 for these 4 bits produces black, 0100 produces red, and 1111 produces intense white.

The color palette specifications for a monitor enable the user to turn primary colors on and off only. In contrast, the color palette specifications for television operation enable the user to specify a value from 0 to 7 for the contribution of each primary color.

When using a television set, the system uses bits 0 through 8 to define the red, green, and blue components of a color and bit 9 to key onto an external video signal. If genlock is enabled, only those colors with the key bit set

are displayed. A typical use for this color transparency feature is to turn off the background color so that an external video signal can provide the background. (See "External Video Synchronization" and the SET DISPLAY SYNC FEATURES command description, both later in this section, for more details.) For a television, bits 0 through 2 specify the contribution of red, bits 3 through 5 specify the contribution of green, and bits 6 through 8 specify the contribution of blue. For example, a value of 0 for bits 0 through 8 produces black and a value of 1 produces white.

A user program should always use the SET PALETTE command from the ROM BIOS to set the screen border color and the color values for the system.

The color index values select corresponding palette locations: a two-color mode having color indexes 0 and 1 uses palette locations 0 and 1, a four-color mode uses palette locations 0, 1, 2, and 3, and so on. The color index values and corresponding palette locations depend on the screen mode selected with the SET SCREEN MODE command. The border color, however, can be any of the 16 palette locations regardless of the current screen mode.

The commands associated with specifying the color palette are listed below and described more fully later in this section.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Set Palette (EFH)	10	0A	Specifies the screen border color and transfers color data from a user-specified area of memory to the color palette.
Get Palette (EFH)	11	0B	Returns the value of the border color and transfers color data from the palette to a user-specified area of memory.

Display Interrupt Control and VBLANK Operations

To enable the user to synchronize CPU and GCP operations, the Mindset computer provides a display interrupt. This interrupt occurs each time the screen displays a specified scan line. If the user does not specify a different scan line, the display interrupt occurs immediately after the system displays the last scan line of the screen (before starting to display the bottom border--see Figure 4-2).

The purpose of the display interrupt is to provide a way to synchronize the execution of a user-defined routine with screen display operations. Thus, a user-defined interrupt service routine can be called when the display interrupt occurs. See the SET DISPLAY INTERRUPT and SET DISPLAY INT ADDRESS command descriptions for details on how to implement a display interrupt service routine.

Synchronization between user-defined routines and display operations is required for smooth animation. If a program changes the display while the display processor is updating the screen, flickering may occur. To avoid flickering, a user-defined routine that changes the display should execute in the time between screen updates. This time is known as VBLANK time.

VBLANK stands for vertical blanking. During VBLANK, the electron beam of the CRT paints the bottom border, returns to the top of the screen (vertical flyback), and paints the top border (see Figure 4-2). The VBLANK period is about 4 milliseconds long. The screen-update cycle is 16.7 milliseconds long.

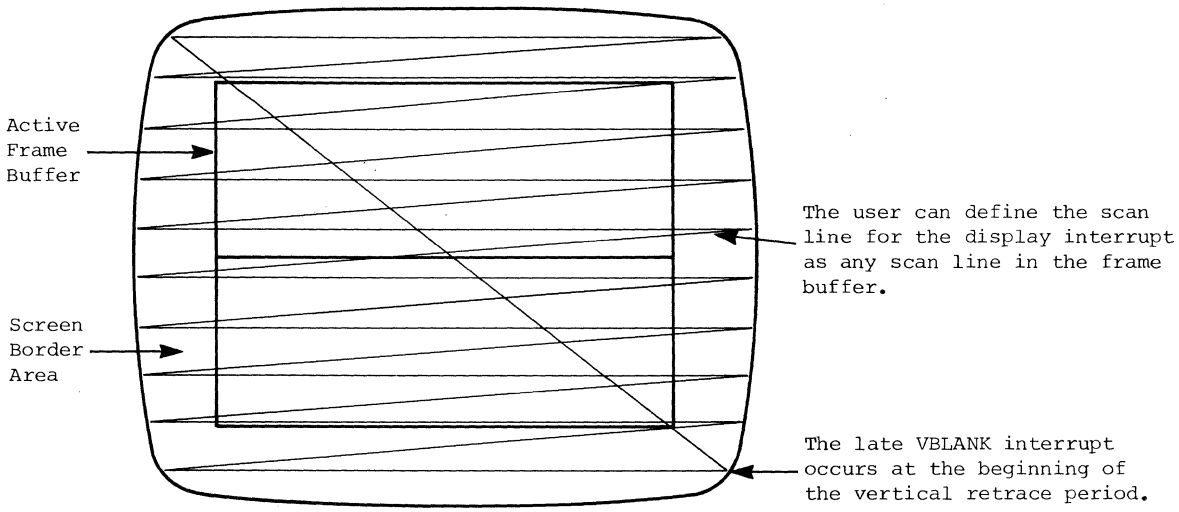


Figure 4-2: Screen layout of the Mindset Personal Computer

Early VBLANK starts when the system begins drawing the bottom border. The late VBLANK (vertical flyback) interrupt is a system interrupt that starts when the system finishes drawing the bottom border.

Unless the user selects a scan line for the display interrupt, this interrupt occurs at early VBLANK time, which follows the painting of the lowest scan line in the frame buffer. The address defined with the SET DISPLAY INT ADDRESS command directs the system to the user-defined interrupt service routine.

When the display interrupt occurs, the system first performs its housekeeping chores; then it calls the user routine, if any. The system performs the operations in the following order:

1. Clears the interrupt.

2. Checks for character mode. If in character mode, the system updates the screen according to the active page buffer.

3. If not in character mode, the system checks for the existence of a user-defined display interrupt service routine.

4. If the user called the SWITCH ACTIVE BUFFER command, the system sets a flag to cause the active buffer switch during the late VBLANK interrupt.

5. The system calls the user-defined display interrupt service routine if one exists.

6. Returns from the interrupt.

The system performs the following tasks as part of the late VBLANK (vertical flyback) interrupt service routine.

1. Clears the interrupt.

2. If the user has redefined any portion of the color palette, the system updates the palette.

3. If the user switches the Mindset computer from internal video signal synchronization to external synchronization, the display processor performs this switch.

4. If the user changes modes before the VBLANK period, the system disables the video signal to the screen when the mode change occurs, sets up for the new mode, and then reenables the video signal to the screen at late VBLANK time.

5. The display processor begins to display from the appropriate frame buffer.

The commands for specifying display interrupt operation are listed below and described more fully later in this section.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Set Display Interrupt (EEH)	07	07	Specifies the scan line at which the display processor causes the CPU to perform a user-defined interrupt service routine.
Set Display Int Address (EFH)	15	0F	Specifies the address of the user-defined display interrupt service routine that the system calls as part of each display interrupt.
Get Display Int Address (EFH)	16	10	Returns the address of the user-defined display interrupt service routine.

External Video Synchronization

The Mindset computer provides the capability of mixing text and graphics with the video signal from an external source. This feature can be used to display text and graphics superimposed on a picture from a video camera.

The command for specifying external video synchronization is listed below and described more fully later in this section.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Set Display Sync Features (EEH)	06	06	Enables or disables the use of genlock for transparent colors, enables or disables interlaced sync display, and enables or disables fixed-phase display.

DESCRIPTIONS OF DISPLAY PROCESSOR BIOS COMMANDS

The user invokes the following graphics commands with software interrupt EEH or EFH. The description for each command explains the register usage and the functions of all parameters.

SET SCREEN MODE -- Int. EFH, (AH) = 00H

Function: Sets the graphics display mode for both frame buffers and all BLTs.

Description: The SET SCREEN MODE command controls only the Mindset graphics display modes. Software interrupt 10H sets the industry-compatible screen modes for the Mindset computer.

The number of bits per pixel selected with the SET SCREEN MODE command applies to all GCP operations.

The seven available graphics display modes are listed in Table 4-1.

Table 4-1: Graphics Display Modes

Screen Mode	Resolution	Colors	Buffering
0	320 x 200 x 1	2	Single or double
1	320 x 200 x 2	4	Single or double
2	320 x 200 x 4	16	Single only
3	640 x 200 x 1	2	Single or double
4	640 x 200 x 2	4	Single only
5	320 x 400 x 2	4	Single only, interlaced
6	640 x 400 x 1	2	Single only, interlaced

SET SCREEN MODE performs the following functions:

1. Sets the screen resolution according to the value in register AL.

2. Zeros the frame buffers.

3. Sets the first frame buffer as the GCP destination buffer.

4. Disables collision detection.

5. Sets the clipping bounds to include the entire frame buffer.

6. Sets the clipping and collision indicators to zero.

7. Enables clipping.

8. Does not zero the user interrupt vectors for the scan line interrupt and the collision/clip interrupt.

9. Resets the display interrupt to occur when the system completes the painting of the visible frame buffer (that is, at early VBLANK time).

10. Sets the cursor position to location (0,0) for all pages.

11. Disables transparency.

12. Sets the transfer mode to replace the destination with the source.

13. Sets the source AND mask and the destination write mask to OFFFFH.

14. Resets the parameter block mode to "contiguous".

Input Parameters: (AL) selects the screen mode according to the screen mode values listed in Table 4-1.

Output Parameters: None. All registers remain unchanged.

GET SCREEN MODE -- Int. EFH, (AH) = 01H

Function: Returns current screen mode parameters.

Description: The GET SCREEN MODE command returns the number of the currently selected screen mode, the number of bits per pixel in this mode, and a series of status flags. Some of these flags are used only by the system and do not provide information that is useful to user programs.

Input Parameters: None.

Output Parameters:

(AH) is undefined.

(AL) is the number of the screen mode as defined in SET SCREEN MODE.

(BX) returns the following flags:

Bit	Definition (when bit = 1)
0	System flag.
1	Industry-standard APA mode is enabled.
2	Industry-standard character mode is enabled.
3	Double buffering is permitted.
4	50 Hz display refresh rate in effect (otherwise 60 Hz).
5	System flag.
6	Interlaced mode is enabled.
7	System flag.
8	External sync is enabled.
9	Genlock is enabled.
10	Second frame buffer is on display.
11	80-column character mode is enabled.
12	System flag.
13	Phase alternate line (PAL) operation is enabled (otherwise National TV Standards Committee (NTSC) operation).
14	TV operation is enabled (otherwise monitor).
15	Black-and-white TV signals are being generated (otherwise color).

(CX) is the number of bits per pixel.

All other registers remain unchanged.

SET DISPLAY DEVICE -- Int. EEH, (AH) = 02H

Function: Sets the display for either a television or an RGB color monitor and then reloads the color palette.

Description: The SET DISPLAY DEVICE command changes the current display setup to that for a color television, black-and-white television, or color monitor. After the system changes the setup, it reloads the color palette for the selected device.

Input Parameters: (AL) sets the display setup according to the following bit definitions:

Bit	Definition
0	is a 0 for television operation or a 1 for monitor operation.
1	is a 0 for color television or monitor or a 1 for black-and-white television only.

Output Parameters: None. All registers remain unchanged.

SET SCREEN POSITION -- Int. EEH, (AH) = 03H

Function: Sets the position of the screen within the screen border.

Description: The SET SCREEN POSITION command uses a single parameter to define the position of the screen within the screen border. This 1-byte parameter uses the most significant 4 bits to specify the vertical (Y-axis) location of the screen and the least significant 4 bits to specify the horizontal (X-axis) location.

Setting (AL) to 00H places the screen at the upper left corner of the border and 0FFH places the screen at the lower right corner of the border.

The SET SCREEN POSITION command affects only the current setting. The system configuration screen defines the default setting.

Input Parameters: (AL) is the location of the screen. Bits 0 through 3 specify the X-coordinate of the screen, and bits 4 through 7 specify the Y-coordinate.

Output Parameters: None. All registers remain unchanged.

GET SCREEN POSITION -- Int. EEH, (AH) = 04H

Function: Returns the position of the screen within the screen border.

Description: The GET SCREEN POSITION command returns a single parameter that reports the current position of the screen within the screen border. This 1-byte parameter uses the most significant 4 bits to specify the vertical (Y-axis) location of the screen and the least significant 4 bits to specify the horizontal (X-axis) location.

A value of 00H in (AL) causes the screen to appear at the upper left corner of the border and 0FFH causes the screen to appear at the lower right corner of the border.

Input Parameters: None.

Output Parameters:

(AL) is the location of the screen. Bits 0 through 3 specify the X-coordinate of the screen, and bits 4 through 7 specify the Y-coordinate.

All other registers remain unchanged.

SET DISPLAY SYNC FEATURES -- Int. EEH, (AH) = 06H

Function: Enables or disables the use of genlock for transparent colors, enables or disables interlaced sync display, and enables or disables fixed-phase display.

Description: The SET DISPLAY SYNC FEATURES command performs three distinct but related functions: enabling genlock, enabling interlaced sync display, and enabling fixed-phase display. Enabling genlock makes certain colors transparent so that an external video signal can show through.

Normally, the Mindset computer provides its own sync signal for video operation (internal video synchronization is the default). Using genlock, the Mindset computer can alternatively synchronize its output with the output from another video device such as a camera or video tape recorder. The genlock enable bit determines whether the Mindset computer provides its sync signal independently or matches its sync signal with the sync signal of an external device.

Genlock is normally disabled. When the user enables genlock, it affects only the colors that do not have the key bit set within their color palette definition; enabling genlock makes these colors transparent. This feature is most useful for making the background transparent so that an external video signal provides the background.

The interlaced sync display is normally disabled. Enabled, this feature uses 200 lines of display data, but displays data with an interlaced sync using 400 scan lines. Using an even/odd scan line pair to display each row of pixels creates a more filled-in look on the display.

Fixed-phase synchronization, when enabled, provides more flexibility in pixel-by-pixel color mixing and prevents flashing on the display. However, it also changes the television display so that some pictures may appear distorted.

Input Parameters: (AL) sets display sync features according to the following bit definitions:

Bit	Function
0	is a 1 to enable genlock or a 0 to disable genlock.
1	is unused.

Bit	Function
2	is a 1 to enable interlaced sync or a 0 to disable interlaced sync.
3	is a 1 to enable fixed phase or a 0 to disable fixed phase.

Output
Parameters: None. All registers remain unchanged.

SET DISPLAY INTERRUPT -- Int. EEH, (AH) = 07H

Function: Specifies the scan line at which the display processor causes the CPU to perform a user-defined interrupt service routine. This command also enables and disables a diagnostic marker that appears on the specified scan line.

Description: The SET DISPLAY INTERRUPT command enables the user to synchronize animation with the drawing of the screen by the display processor. This command controls a marker to assist the user in tracing the operation of the display interrupt. This diagnostic marker is a horizontal line that appears on the scan line that triggers the display interrupt.

The display interrupt operates only in the graphics mode of the Mindset computer.

A scan line is comprised of a row of pixels. The top scan line on the screen is line 0 and the bottom scan line is line 199. In interlaced graphics modes, the bottom scan line is effectively the 398th (on even line passes) or 399th (on odd line passes) line displayed, but is still referred to as line 199 in this command because it is the 199th line scanned on each pass.

If the user does not define a scan line with the SET DISPLAY INTERRUPT command, the system uses line 199 as the default scan line for calling the user-defined interrupt routine.

Use the SET DISPLAY INT ADDRESS command to specify the address of the user-defined service routine (see "Graphics Coprocessor Operation" later in this section). The default value for this vector is zero. If the vector is zero, the system does not call a user-defined interrupt service routine.

Input Parameters:

(AL) is the number of the scan line for the display interrupt.

(BL) is 1 to enable the diagnostic marker or 0 to disable the diagnostic marker.

Output Parameters: None. All registers remain unchanged.

SET PALETTE -- Int. EFH, (AH) = 0AH

Function: Specifies the contents of the color palette and selects the color of the screen border.

Description: The SET PALETTE command specifies all or part of the contents of the color palette and selects the screen border color. The user must supply a color data array in memory that the SET PALETTE command copies into the color palette.

The SET PALETTE command includes a BORDER color parameter, an offset into the color palette, a count of the number of words to be copied to the color palette (N), and an ARRAY parameter that is a pointer to the first word of the user-supplied color data.

The BORDER parameter of the SET PALETTE command is an index that selects an entry from the color palette as the screen border color. The system ignores BORDER values greater than decimal 15.

The offset parameter is an index value from 0 to 15 that specifies a word within the color palette. Beginning with this word, the system copies the user-specified color data into the color palette.

The N parameter specifies the number of words to copy from the user array to the color palette. This value must be from 0 to 16. The system does not alter the color palette if the value of N is 0.

The ARRAY parameter is the address of the first word to be copied from the user array to the color palette.

Input Parameters:

(AL) is the BORDER color index into the color palette.

(CX) is N, the number of words copied from the user-specified color data to the color palette.

(DX) is the offset into the color palette at which to begin writing the user-specified color data.

(ES:[BX]) is the address of the user-specified color palette data.

Output Parameters:

(AX) is undefined.

All other registers remain unchanged.

GET PALETTE -- Int. EFH, (AH) = 0BH

Function: Returns the value of the screen border color and the contents of the color palette.

Description: The GET PALETTE command returns all or part of the contents of the color palette and the index into the color palette that selects the screen border color. The user must provide an area in memory to receive the color data from the color palette.

The input parameters for the GET PALETTE command include an offset into the color palette, a count of the number of words to be copied from the color palette (N), and an ARRAY parameter that is a pointer to the area of memory receiving the color palette data.

The offset parameter is an index value from 0 to 15 that specifies a word within the color palette. Beginning with this word, the system copies the color palette data into the user-specified area of memory.

The N parameter specifies the number of words to copy from the color palette to the user array. This value must be from 0 to 16. The system does not copy any color palette data if the value of N is 0.

The ARRAY parameter is the user-specified address of the area that is to receive the color palette data.

The output parameters include a BORDER color parameter and the user-specified area of memory that receives the requested data from the color palette.

The BORDER parameter of the GET PALETTE command returns the color index of the border. The ARRAY parameter is a pointer to the data area that receives the color palette data.

Input Parameters:

(CX) is N, the number of words copied from the color palette to the user-specified area of memory.

(DX) is the offset into the color palette at which to begin reading data.

(ES:[BX]) is the address of the user-specified area of memory that is to receive the data from the color palette.

Output Parameters:

(AH) is undefined.

(AL) is the color index of the border color.

All other registers remain unchanged.

GET BUFFER INFO -- Int. EFH, (AH) = 0EH

Function: Returns the addresses and size of the system frame buffers.

Description: The GET BUFFER INFO command returns the addresses and size of the system frame buffers.

The parameters returned by this command are FRAME BUFFER 1, FRAME BUFFER 2, and SIZE.

FRAME BUFFER 1 and FRAME BUFFER 2 are the paragraph addresses of the two system frame buffers; these buffers are always paragraph aligned.

In some cases, FRAME BUFFER 1 and FRAME BUFFER 2 may not be contiguous. Therefore, you should not use the difference between the two buffer addresses to calculate the buffer size.

SIZE is the length in bytes of each frame buffer. SIZE is a constant value that is independent of the current display mode. The SIZE value is one half of the actual buffer size for the graphics modes that do not permit double buffering.

In display modes where double buffering is not possible (such as 320 x 200 x 4), the FRAME BUFFER 1 address is the address of the single frame buffer. The actual size of the single frame buffer is 2*SIZE.

Input Parameters: None.

Output Parameters:

(AX) is indeterminate.

(BX) is the FRAME BUFFER 1 paragraph address.

(DX) is the FRAME BUFFER 2 paragraph address.

(CX) is the SIZE of each buffer (or one half of the actual size for display modes restricted to single buffering).

All other registers remain unchanged.

SET DISPLAY INT ADDRESS -- Int. EFH, (AH) = 0FH

Function: Specifies the address of the user-defined display interrupt service routine.

Description: The SET DISPLAY INT ADDRESS command specifies the paragraph and offset addresses of a user-defined interrupt service routine. The system calls this routine each time it finishes writing the frame buffer image to the screen, or when the GCP reaches the scan line selected by the SET DISPLAY INTERRUPT command (see SET DISPLAY INTERRUPT earlier in this section). Specifying an address of 0 informs the system that there is no user-defined interrupt routine to service.

The last scan line on the screen is the default scan line for the display interrupt. This scan line coincides with early VBLANK. In this case, the user's interrupt routine has approximately 1 millisecond before late VBLANK occurs. The system interrupts the user's routine at this time to perform its late VBLANK procedures, and then returns to the user's routine.

The system uses the late VBLANK interrupt to change the color palette and to switch the active frame buffer. Therefore, if the user-defined display interrupt service routine makes changes to the color palette or switches the active frame buffer, these changes appear following the next late VBLANK interrupt.

Because the system interrupt handler invokes the user-defined service routine with a FAR CALL instruction, this routine must terminate with a FAR RET instruction instead of an IRET instruction.

The user's main program must disable the display interrupt vector before the main program terminates. Otherwise, the next display interrupt would cause the system to jump to a nonexistent service routine.

Input Parameters: (ES:[BX]) is the address of the user-defined display interrupt service routine. A value of (0:[0]) directs the system to not call a user-defined display interrupt service routine.

Output Parameters: (AX) is undefined.

All other registers remain unchanged.

GET DISPLAY INT ADDRESS -- Int. EFH, (AH) = 10H

Function: Returns the current address of the user-defined display interrupt service routine.

Description: The GET DISPLAY INT ADDRESS command returns the paragraph and offset addresses last set by the SET DISPLAY INT ADDRESS command.

If the address of the display interrupt routine is other than zero, the system uses this address as the beginning of a user-defined display interrupt service routine. If this address is zero, the system does not call a user-defined interrupt routine when the display interrupt occurs.

Input Parameters: None.

Output Parameters: (ES:[BX]) is the paragraph and offset addresses of the user-defined display interrupt service routine. A value of (0:[0]) directs the system not to call a user-defined display interrupt service routine.

All other registers remain unchanged.

SWITCH ACTIVE BUFFER -- Int. EFH, (AH) = 11H

Function: Causes the system to switch active frame buffers.

Description: The SWITCH ACTIVE BUFFER command operates in modes with double buffering to switch the active frame buffer with the hidden frame buffer. This command has no effect in modes restricted to single buffering such as 320 x 200 x 4.

Input Parameters: None.

Output Parameters: (AX) is undefined.

All other registers remain unchanged.

EFFECT OF DISPLAY PROCESSOR COMMANDS ON GRAPHICS COPROCESSOR OPERATION

The SET SCREEN MODE and SWITCH ACTIVE BUFFER Commands

Certain modes set using the display processor SET SCREEN MODE command allow single buffer operation only. The single buffer modes are 2, 4, 5, and 6 (see Table 4-2). This means that the SWITCH ACTIVE BUFFER command has no effect when in modes 2, 4, 5, or 6.

The SET SCREEN MODE command initializes a number of other parameters that affect GCP operation. The following is a list of the SET SCREEN MODE initialization effects.

1. Sets the screen resolution according to the value in register AL.
2. Zeros the frame buffers.
3. Sets the first frame buffer as the GCP destination buffer.
4. Disables collision detection.
5. Sets the clipping bounds to include the entire frame buffer.
6. Sets the clipping and collision indicators to zero.
7. Enables clipping.
8. Does not zero the user interrupt vectors for the scan line interrupt and the collision/clip interrupt.
9. Resets the display interrupt to occur when the system completes the painting of the visible frame buffer (that is, at early VBLANK time).
10. Sets the cursor position to (0,0) for all pages.
11. Disables transparency (genlock).
12. Sets transfer mode to replace destination with source.
13. Sets the source AND mask and the destination write mask to OFFFFH.
14. Resets the parameter block mode to "contiguous".

The SET PALETTE Command

Color is an important option in graphics images. The SET PALETTE command specifies colors for use in image development. See "Descriptions of Display Processor BIOS Commands" earlier in this section for a complete description of the SET PALETTE command.

The Display Interrupt

The display interrupt facilitates synchronization between CPU and GCP operations and is thus important in animation and other graphics operations. See "Display Processor Operation" earlier in this section for details on the display interrupt.

GRAPHICS COPROCESSOR OPERATION: IMAGE CREATION AND BLOCK TRANSFER OPERATIONS

The commands in this category control the operation of the graphics coprocessor. Each GCP operation takes data from one area of memory (the "source") and combines it with the data in another area of memory (the "destination").

The GCP does not simply overwrite the information in the destination with the source data. Instead, it combines each pixel of information according to the user-specified transfer mode. The user can also implement a write mask to protect certain bits in each word of the destination during a GCP operation.

The GCP does not modify the source data during a transfer operation unless the source and destination areas overlap.

Collision and Clip Detection

The collision and clip specification commands enable the user to establish how collisions and clips are detected. The collision and clip detection commands enable the user to determine if the preceding block transfer (BLT) or draw command caused a collision or a clip.

The user program can define a clipping rectangle within the destination buffer. The ROM BIOS provides clipping detection to notify the user when a clip has occurred during a previous BLT operation. Likewise, the ROM BIOS provides collision detection to notify the user when two overlapping objects satisfy user-defined collision criteria.

The commands for image creation and transfer operations are listed below and described more fully later in this section.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Set Transfer Mode (EFH)	02	02	Specifies the transparent or opaque mode and the NOT, AND, OR, or exclusive-OR mode for GCP data transfers.
Get Transfer Mode (EFH)	03	03	Returns the current GCP transfer mode parameters.
Set Destination Buffer (EFH)	04	04	Specifies the address and size of the BLT destination buffer.
Get Destination Buffer (EFH)	05	05	Returns the address and size of the current BLT destination buffer.

Command (Interrupt)	Value of AH Decimal	Hex	Function
Set Write Mask (EFH)	06	06	Specifies a 16-bit write mask that enables selective bit modification during data transfer operations.
Get Write Mask (EFH)	07	07	Returns the 16-bit write mask.
BLT Copy (EFH)	08	08	Copies a region of the source buffer to the destination buffer.
BLT Copy Word (EFH)	09	09	Fills rectangular regions of the destination buffer with a user-specified 16-bit fill pattern.
BLT Polypoint (EFH)	12	0C	Draws a series of points of the same color into the destination buffer.
BLT Polyline (EFH)	13	0D	Draws a series of lines of the same color into the destination buffer.
Set Collision Pattern (EFH)	18	12	Specifies the pattern that, when found, indicates a collision.
Get Collision Pattern (EFH)	19	13	Returns the current collision pattern.
Set Clip Rectangle (EFH)	20	14	Specifies the clipping bounds for graphics commands.
Get Clip Rectangle (EFH)	21	15	Returns the current clipping bounds.
Set Collision/Clip/ Done Detect (EFH)	22	16	Sets the collision and clip enable flags and the address for the user-defined collision/clip/done interrupt service routine.
Get Collision/Clip/ Done Detect (EFH)	23	17	Returns the settings of the collision and clip enable flags and the address for the user-defined collision/clip/done interrupt service routine.
GCP Wait (EFH)	24	18	Waits until the block-transfer hardware completes its current task and then returns the collision/clip status.
BLT Polygon (EFH)	25	19	Draws a filled polygon into the destination buffer.

Command (Interrupt)	Value of Decimal	AH Hex	Function
BLT Filled Ellipses (EFH)	26	1A	Draws a series of filled ellipses or ellipse sectors into the destination buffer.
BLT Hollow Ellipses (EFH)	27	1B	Draws a series of hollow ellipses or elliptical arcs into the destination buffer.
Save GCP (EFH)	28	1C	Saves the current state of the graphics coprocessor in the user-specified data area.
Restore GCP (EFH)	29	1D	Restores data previously saved by the Save GCP command.
Fill Dest Buffer (EFH)	30	1E	Fills the destination buffer with specified data.
Set Param Block Mode (EFH)	34	22	Specifies the representation for certain BLT command parameter blocks.
Get Param Block Mode (EFH)	35	23	Returns the current parameter block representation mode.
Get GCP Status (EFH)	36	24	Returns the current GCP status word.
Get BLT Memory Bounds (EFH)	38	26	Returns the address bounds of memory available for block transfers.

Custom-Character-Set Operations

The user can define a set of graphics characters for use in graphics creations. These characters can be written to the display using the BLT STRING command in any transfer mode and using any combination of the GCP features. The user can specify a proportional font, with each character sized individually, or a fixed font, with all characters the same size. See the description of the SET FONT POINTER command for further details.

The commands for custom-character-set operations are listed below and described more fully later in this section.

Command (Interrupt)	Value of Decimal	AH Hex	Function
Set Font Pointer (EFH)	31	1F	Sets a pointer to a user-defined font descriptor block.
Get Font Pointer (EFH)	32	20	Returns the pointer to a user-defined font descriptor block.

Command (Interrupt)	Value of AH Decimal	Hex	Function
BLT String (EFH)	33	21	Transfers a string of characters into the destination buffer.
Get Char Bitmap Address (EFH)	37	25	Returns the addresses of the two system character bitmaps.

GRAPHICS TECHNIQUES

Initialization and Mode Selection

The ROM BIOS of the Mindset computer provides service routines that interface to the hardware that controls the video modes of the computer. The application program should always use these ROM BIOS service routines to set up the modes and to initialize the computer for that application. By using the ROM BIOS to control the hardware of the Mindset computer, the user program remains independent of potential hardware modifications.

The screen display of the Mindset computer system in graphics mode consists of a frame buffer image and a border. The frame buffer is the area of memory containing data that the display processor interprets as the refresh bitmap display. The border is a single color boundary that surrounds the frame buffer image.

The Mindset ROM BIOS includes SET SCREEN POSITION and GET SCREEN POSITION commands. These commands enable the user to match the screen display with the overscan characteristics of a particular television. The user specifies the border color with the SET PALETTE command.

There are 7 graphics modes, each with differing characteristics. One mode can display up to 16 colors simultaneously. In other modes, the user can implement double buffering to change the entire screen display instantaneously. The 640 x 400-pixel resolution of mode 6 allows extremely detailed graphics images.

Table 4-2 lists the 7 graphics modes of the Mindset computer system.

Table 4-2: Mindset Graphics Modes

Graphics Mode	Screen Resolution	Possible Colors	Bits per Pixel	Characteristics
0	320 x 200	2	1	Single or double buffering
1	320 x 200	4	2	Single or double buffering
2	320 x 200	16	4	Single buffering
3	640 x 200	2	1	Single or double buffering
4	640 x 200	4	2	Single buffering
5	320 x 400	4	2	Interlaced, single buffering
6	640 x 400	2	1	Interlaced, single buffering

Double-Buffered Video RAM Operation

Three of the Mindset computer graphics modes support double buffering. Double buffering is the technique of using two frame buffers--one active (visible) and one hidden. While the system is displaying the active frame buffer, the user can update the hidden frame buffer without disturbing the screen. The user can then switch frame buffers when the hidden frame buffer is ready for display.

The user program can construct an image in the nondisplayed frame buffer and then switch frame buffers so that the new image appears "instantaneously". The SWITCH ACTIVE BUFFER command changes the entire screen image in a single frame (1/60 second). The construction of a complex multiple-object image in the visible frame buffer could take many times this long and distract the viewer.

The maximum length of each frame buffer in modes that support double buffering is 16,000 bytes. The maximum length of the single frame buffer in modes that do not support double buffering is 32,000 bytes.

The ROM BIOS includes the GET BUFFER INFO command that returns the addresses and length of the two frame buffers.

Object Definition, Location, and Status

The ROM BIOS of the Mindset computer provides service routines to move objects to or within the destination buffer. The destination buffer may be a frame buffer or any user-defined buffer in memory. The user program defines the rectangular region of a source buffer that contains the object and the rectangular region of the destination buffer that is to receive the object. The user program must maintain the definition, location, and status of all objects.

Collision and Clip Detection

The ROM BIOS of the Mindset computer provides detection for collisions and clipping of objects in the destination buffer.

The ROM BIOS includes one routine to set the bit pattern that defines a collision. A collision occurs when two objects overlap and the bit pattern of the pixels within the overlapping area matches the user-defined collision pattern.

Another ROM BIOS routine defines a clipping rectangle within the destination buffer. Clipping occurs when an object exceeds user-defined clipping boundaries within the destination frame buffer.

A single ROM BIOS routine enables the user to enable or disable clipping and/or collision detection. This routine specifies the address of the user-defined service routine that the system invokes when it detects a clip or a collision.

As an alternative to using the collision and clip interrupts, the user can periodically call the GCP WAIT command to poll the collision/clip status. The GCP WAIT command returns the collision/clip status of up to 16 objects placed in the destination buffer by the preceding graphics call.

Block Transfer

The Mindset computer provides extensive commands for transferring graphics data in blocks from place to place in memory. This capability facilitates image creation and animation. You can define components (blocks) of an image and build the whole image by transferring the components to the appropriate locations. You can transfer image blocks as desired and use double buffering to create animated sequences. You can also modify the data as it is transferred to change color or patterns (see the SET WRITE MASK command). See the BLT COPY command for more details on block transfers.

Interrupt Handling

The ROM BIOS of the Mindset computer system provides two user-defined interrupt-handling routines. One of these interrupts occurs each time the display processor displays a specific scan line. The other interrupt occurs when the system detects a collision, a clip, or the end of a GCP task.

The user must ensure that his interrupt service routines can be interrupted safely by another interrupt service routine. The user must disable (reset to zero) all user interrupt vectors before terminating a program. Otherwise, the next program could branch to a non-existent interrupt routine.

The user must save and restore any registers used by the interrupt service routine. When an interrupt occurs, the system interrupt handler invokes the user-defined interrupt routine with a FAR CALL instruction. Therefore, the user must terminate each interrupt service routine with a FAR RET instruction (not an IRET). When assembling programs, the user should create all interrupt service routines as FAR procedures.

The user may control the GCP as part of either a normal routine or an interrupt routine. The following procedures are recommended for interrupt service routines if the user includes GCP operations in both normal and interrupt routines:

1. Use the GCP WAIT command to let the GCP finish its current operation.

2. Use the SAVE GCP command to save the current status of the GCP on the stack.

3. Perform the user-defined interrupt service routine.

4. Use the RESTORE GCP command to return the GCP to its previous status.

5. Use a FAR RET instruction to return from the interrupt.

Orientation of Screen Images

In all references in this section, the pixel at the upper left corner of a rectangular region is location (0,0). Distances measured to points below and to the right of location (0,0) are positive values. For example, the pixel at location (5,3) is 5 pixels to the right and 3 pixels below location (0,0).

DESCRIPTION OF GRAPHICS COPROCESSOR BIOS COMMANDS

The user invokes the following graphics commands with software interrupt EFH. The description for each command explains the register usage and the functions of all parameters.

SET TRANSFER MODE -- Int. EFH, (AH) = 02H

Function: Specifies how the GCP modifies the destination data with the source data.

Description: The SET TRANSFER MODE command selects two modes for GCP operations. The visibility mode can be either opaque or transparent. The logical transfer mode can be one of eight possible logical operations for combining the source and destination data.

The opaque visibility mode transfers all data from the source to the destination.

When operating in the transparent mode, the GCP does not transfer any pixel whose bits are all 0. Other than ignoring any pixel with all 0 bits, the transparent mode operates the same as the opaque mode.

The logical transfer modes determine how the data from each source pixel modifies the data of each destination pixel. The modes that include NOT source data read the source data into a register, complement it, and then perform the AND, OR, exclusive-OR, or replace operation with the destination data. Transparency applies only to pixels that are 0 before the NOT operation.

The SET SCREEN MODE command automatically resets the visibility mode to opaque and the logical transfer mode to replace the destination data with the source data.

Input Parameters:

(BH) selects either the opaque or the transparent visibility mode.

(BH)	Visibility Mode
0	Transfers all bits from the source to the destination.
1	Does not transfer any source pixels with all zero bits.

(BL) selects the logical combination mode for GCP operations.

(BL)	Logical Combination Mode
0	Replaces destination data with source data.
1	Logically ANDs the destination and source data.
2	Logically ORs the destination and source data.
3	Logically exclusive-ORs the destination and source data.
4	Replaces destination data with NOT source data.
5	Logically ANDs the destination and NOT source data.
6	Logically ORs the destination and NOT source data.
7	Logically exclusive-ORs the destination and NOT source data.

Output Parameters:

(AX) is indeterminate.

All other registers remain unchanged.

GET TRANSFER MODE -- Int. EFH, (AH) = 03H

Function: Returns the current transfer mode parameters.

Description: The GET TRANSFER MODE command returns the current values for the visibility mode and the logical transfer mode.

Input Parameters: None.

Output Parameters: GET TRANSFER MODE returns the following information:

(AX) is undefined.

(BH) is 0 for the opaque visibility mode or 1 for the transparent visibility mode.

(BL) returns the logical transfer mode as follows:

(BL)	Transfer Mode
0	Replaces destination data with source data.
1	Logically ANDs the destination and source data.
2	Logically ORs the destination and source data.
3	Logically exclusive-ORs the destination and source data.
4	Replaces destination data with NOT source data.
5	Logically ANDs the destination and NOT source data.
6	Logically ORs the destination and NOT source data.
7	Logically exclusive-ORs the destination and NOT source data.

All other registers remain unchanged.

SET DESTINATION BUFFER -- Int. EFH, (AH) = 04H

Function: Specifies the address and size of the destination buffer for all subsequent GCP operations.

Description: The SET DESTINATION BUFFER command prepares the system for GCP operations by defining the destination buffer for all subsequent data transfers originating from any source buffer. This command automatically resets the clip rectangle to match the bounds of the new destination buffer.

The destination buffer may reside within any segment of the 1-megabyte address space of the Mindset computer as long as the buffer begins on a word boundary.

Input Parameters:

(CX) is the number of bytes per scan line within the destination buffer. (This value must be even.)

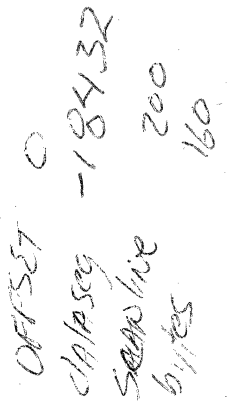
(DX) is the number of scan lines in the destination buffer. (A scan line is 1 pixel high.)

(ES:[BX]) is the address of the first word of the destination buffer.

Output Parameters:

(AX) is indeterminate.

All other registers remain unchanged.



GET DESTINATION BUFFER -- Int. EFH, (AH) = 05H

Function: Returns the current destination buffer parameters.

Description: The GET DESTINATION BUFFER command returns the address and size of the area of memory currently designated as the destination buffer.

Input Parameters: None.

Output Parameters:

(AX) is undefined.

(CX) is the number of bytes per scan line within the destination buffer.

(DX) is the number of scan lines in the destination buffer.

(ES:[BX]) is the address of the first word in the destination buffer.

All other registers remain unchanged.

SET WRITE MASK -- Int. EFH, (AH) = 06H

Function: Sets the write mask for all subsequent GCP operations.

Description: The SET WRITE MASK command defines a 16-bit write mask. The GCP examines the write mask before transferring data to the destination buffer. The bit positions of the write mask correspond to the bit positions of the words within the destination buffer.

A 0 bit in the write mask prevents the GCP from modifying the corresponding bit in a word within the destination buffer. The GCP can modify only those bits that have a 1 in the same bit-position in the write mask.

As a result, a mask of OFFFFH specifies that all bits within each word of the destination buffer may be modified. Conversely, a mask of 0000H prevents a GCP operation from having any effect because this mask protects all destination bits from modification.

The SET SCREEN MODE command automatically resets the write mask to OFFFFH.

Input Parameters: (BX) is the 16-bit write mask.

Output Parameters: (AX) is undefined.

All other registers remain unchanged.

GET WRITE MASK -- Int. EFH, (AH) = 07H

Function: Returns the value of the current write mask.

Description: The GET WRITE MASK command returns the write mask last set by the SET WRITE MASK command (unless modified by the SET SCREEN MODE routine). Each 1 bit within the write mask enables the GCP to modify the corresponding bit of the words within the destination buffer. A 0 bit within the write mask prevents the GCP from modifying that bit in a word.

The SET SCREEN MODE command automatically resets the write mask to OFFFFH.

Input Parameters: None.

Output Parameters:

(AX) is undefined.

(BX) is the current GCP write mask.

All other registers remain unchanged.

BLT COPY -- Int. EFH, (AH) = 08H

Function: Transfers data from the source buffer to the destination buffer (see Figure 4-3).

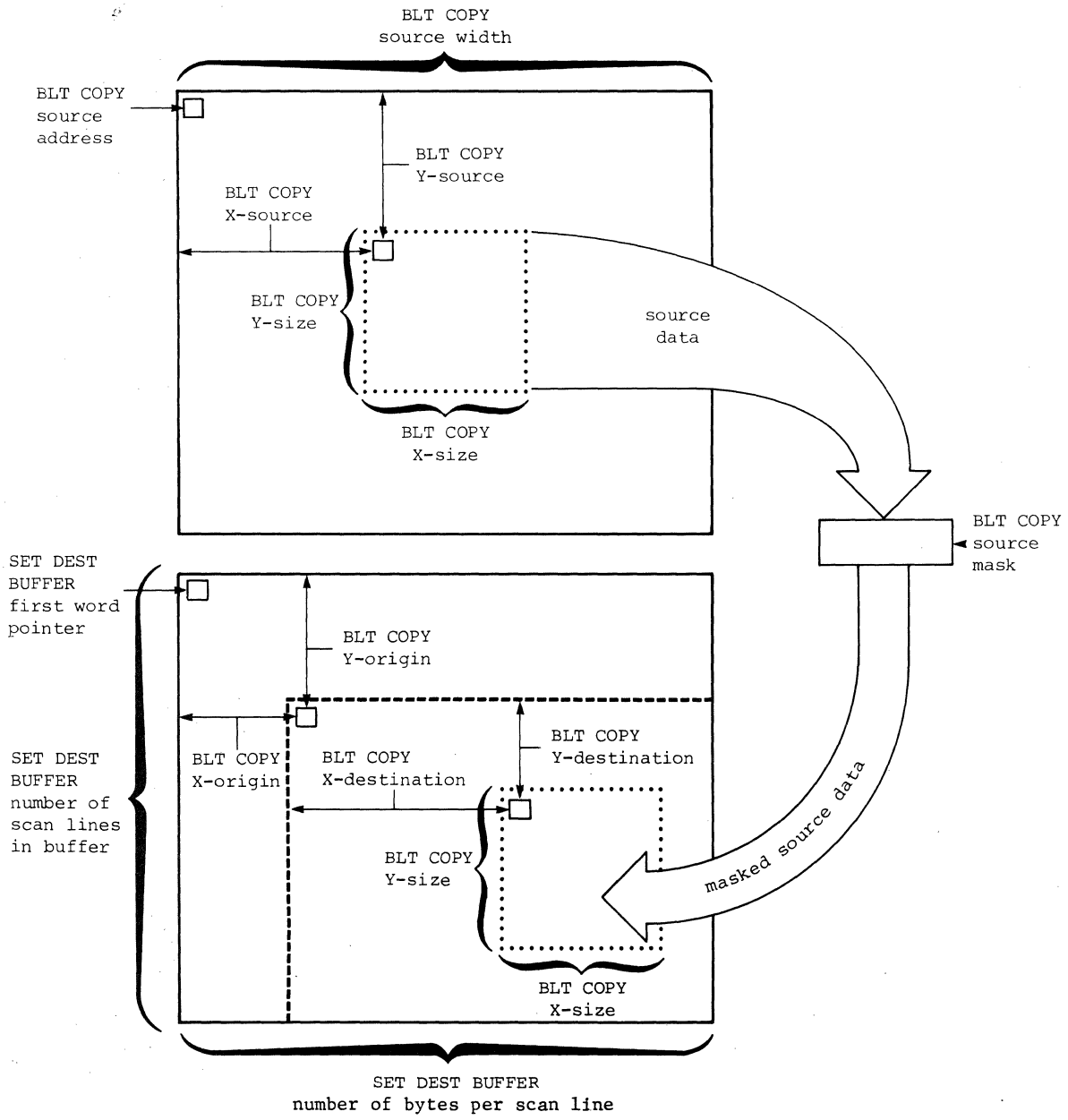


Figure 4-3: Operation of the BLT COPY command

Description: Each BLT COPY command specifies a series of block transfer operations from one or more source buffers to a common destination buffer. The object definitions are stored in memory as an array of parameter groups. The user must supply a BLT COPY parameter group for each separate object to be transferred.

The BLT COPY command includes a BLT ID number, a destination X-ORIGIN, a destination Y-ORIGIN, a BLT COPY parameter group count (N), and the PARAMETER POINTER, which is the location of the first byte of the first 20-byte BLT COPY parameter group.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine in the event that this BLT COPY operation causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are signed offsets (in pixels) that provide a common point of reference within the destination buffer for all block transfers specified by a single BLT COPY command. The pixel at location (0,0) occupies the upper left corner of the destination buffer. The X-ORIGIN and Y-ORIGIN values are positive to the right of and below this pixel, respectively.

The 16-bit BLT COPY parameter group count N specifies the number of transfer operations that are part of a single BLT COPY command. This value should not exceed 16 if the user intends to use the GCP WAIT command to poll for collision and clipping information.

The PARAMETER POINTER is the address of the first byte of the first BLT COPY parameter group. Each BLT COPY parameter group specifies a separate block transfer operation.

The 20-byte BLT COPY parameter group for each block transfer includes the SOURCE ADDRESS offset, the SOURCE ADDRESS PARAGRAPH, the SOURCE WIDTH, an X-SOURCE value, a Y-SOURCE value, an X-SIZE value, a Y-SIZE value, an X-DESTINATION value, a Y-DESTINATION value, and a SOURCE MASK.

The SOURCE ADDRESS offset and the SOURCE ADDRESS PARAGRAPH specify the word that represents the upper left corner of the rectangular source buffer. They are both 16-bit values.

The 16-bit SOURCE WIDTH value within the BLT COPY parameter group specifies the number of bytes per scan line of the source buffer. This value must be an even number.

Although the GCP requires that both the source and destination buffers be word-aligned and a multiple of 16 bits in width, the rectangular region that the hardware transfers may begin on any pixel boundary and use any (even or odd) number of pixels as the horizontal or vertical size.

The 16-bit X-SOURCE and Y-SOURCE values are signed offsets (in pixels) from the pixel at the upper left corner of the source buffer. These values define the upper left corner of the region of the source buffer that the GCP transfers.

The 16-bit X-SIZE and Y-SIZE values are dimensions (in pixels). These values define the width and length of the region of the source buffer that the GCP transfers. If either the X-SIZE or Y-SIZE value is zero, the GCP does not perform the transfer.

The GCP does not perform clipping on the source buffer, so the user must make sure that the specified rectangle lies within the source buffer.

The 16-bit X-DESTINATION and Y-DESTINATION values are signed offsets (in pixels) from the pixel specified by the X-ORIGIN and Y-ORIGIN values. The GCP adds the (X,Y) destination values to the (X,Y) origin values to specify the location of the first destination pixel in a BLT COPY operation. (The (X,Y) origin values are themselves offsets.)

The SOURCE MASK is a 16-bit value that the GCP ANDs with each word of the source data before it transfers the data to the destination buffer. By selectively turning off bits in each pixel, the SOURCE MASK can display an object in a color different from the color in which it was defined. (Note that this operation is different from BLT WRITE MASK, which determines which destination bits may be modified by a BLT COPY operation.)

The following example shows one way to use the BLT COPY command:

1. Use SET SCREEN MODE to specify the screen resolution, to select single- or double-buffered operation, and to select interlaced or non-interlaced output.

2. Use SET DESTINATION BUFFER to specify the location and size of the destination buffer that is to receive the information. To display the transferred information, the destination buffer must be a frame buffer.

3. Use SET TRANSFER MODE to specify the type of transfer.

4. Use SET WRITE MASK to determine which bits within the destination buffer may be modified.

5. Use BLT COPY to specify the following information, which may vary each time BLT COPY is called:

    a. The BLT ID number used for collision and clip detection (BLT ID).
    b. The (X,Y) origin within the destination buffer (X-ORIGIN and Y-ORIGIN). The data transferred by each BLT COPY parameter group is relative to this origin.

c. The number of BLT COPY parameter groups to be processed (N).
d. Certain modes of operation (see input parameter (DX) below).
e. The location of the first BLT COPY parameter group (PARAMETER POINTER).

6. Use each BLT COPY parameter group to specify the following information relating to a single block transfer:

a. The location of the source buffer (SOURCE ADDRESS offset and SOURCE ADDRESS PARAGRAPH).
b. The width (in an even number of bytes) of the source buffer (SOURCE WIDTH).
c. X and Y offsets (in pixels) into the source buffer (X-SOURCE and Y-SOURCE). These values determine the upper left corner of the rectangular region of the source buffer to be transferred.
d. The width and height (in pixels) of the portion of the source buffer that is to be transferred (X-SIZE and Y-SIZE).
e. X and Y offsets (in pixels) into the destination buffer (X-DESTINATION and Y-DESTINATION). These offset values are relative to X-ORIGIN and Y-ORIGIN.
f. The mask that the system ANDs with each source word before it transfers the word to the destination buffer (SOURCE MASK).
g. The offset to the next parameter group (optional; see the SET PARAM BLOCK MODE command).

7. Use SET COLLISION/CLIP/DONE DETECT to specify collision and clip operations.

8. Use SET CLIP RECTANGLE to specify the clipping bounds if necessary.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine if this BLT COPY operation results in a collision or a clip.

(CX) is N, the number of parameter groups in this BLT COPY operation.

(DX) specifies mode command flags for the N BLT COPY parameter groups as follows:

Bit(s)	Bit Value	Definition
0	0	Read source from top to bottom.
	1	Read source from bottom to top.

Bit(s)	Bit Value	Definition
1	0	Write destination from top to bottom.
	1	Write destination from bottom to top.
2	0	Read and write from left to right.
	1	Read and write from right to left.
3		Reserved. Must be the same as bit 2.
4	0	Normal BLT mode.
	1	Fast BLT mode--provided to enhance master-slave double buffering. In fast BLT mode, the system moves data from the source to the destination as quickly as possible by: 1. Ignoring the current setting of the transfer mode (opaque or replace mode is used). 2. Ignoring the current setting of collision detect (collision testing is suspended). Clipping, if enabled, is performed as usual. The user must ensure that, after clipping: 1. (DX) bits 0 through 3 are zero. 2. The first pixel of both the source and destination rectangles are word aligned. 3. The width of the transferred rectangle is an integer number of words.
5-15		Reserved. Must be zero.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the PARAMETER POINTER, which is the address of the first byte of the first 20-byte BLT COPY parameter group. The BLT COPY parameter groups can be consecutive or in the form of a linked list (see the SET PARAM BLOCK MODE command description). The GCP services each BLT COPY parameter group in sequence until it performs the number of transfer operations specified in the BLT COPY parameter group count.

Each 20-byte parameter group contains the following parameters (listed by their decimal offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	SOURCE ADDRESS offset
+ 2	SOURCE ADDRESS PARAGRAPH
+ 4	SOURCE WIDTH in bytes (must be an even number)
+ 6	X-SOURCE offset from source address in pixels
+ 8	Y-SOURCE offset from source address in pixels
+10	X-DESTINATION offset from X-ORIGIN in pixels
+12	Y-DESTINATION offset from Y-ORIGIN in pixels
+14	X-SIZE width of source buffer region in pixels
+16	Y-SIZE length of source buffer region in pixels
+18	SOURCE MASK to AND with source data during a block transfer
+20	Optional: OFFSET to the next parameter group. This OFFSET must be included if SET PARAM BLOCK MODE is used to specify the linked list mode. This OFFSET must <u>not</u> be included when SET PARAM BLOCK MODE is used to specify the contiguous parameter block representation mode. An OFFSET value of OFFFFH indicates the end of the list. BLT COPY terminates on the first occurrence of N BLTs (N is the number in CX) <u>or</u> a OFFFFH link OFFSET value.

Output Parameters: None. All registers remain unchanged.

BLT COPY WORD -- Int. EFH, (AH) = 09H

Function: Fills rectangular regions of the destination buffer with a 16-bit pixel pattern.

Description: Each BLT COPY WORD command specifies a series of block fill operations within the destination buffer. The definitions of the regions to be filled and the fill patterns are stored as an array of parameter groups in memory. The user must supply a BLT COPY WORD parameter group for each filled block in the destination buffer.

The BLT COPY WORD command includes a BLT ID number, a destination X-ORIGIN, a destination Y-ORIGIN, a BLT COPY WORD parameter group count (N), and the PARAMETER POINTER, which is the location of the first byte of the first 10-byte BLT COPY WORD parameter group.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine if this BLT COPY WORD operation causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are signed offsets in pixels. They provide a common point of reference within the destination buffer for all rectangles filled by a single BLT COPY WORD command. The pixel at location (0,0) occupies the upper left corner of the destination buffer; the X-ORIGIN and Y-ORIGIN values are relative to this pixel.

The 16-bit BLT COPY WORD parameter group count N specifies the number of fill operations that are part of a single BLT COPY WORD command. This value should not exceed 16 if the user intends to use the GCP WAIT command to poll for collision and clipping information.

The PARAMETER POINTER is the address of the first byte of the first BLT COPY WORD parameter group. Each BLT COPY WORD parameter group specifies a separate fill operation.

The 10-byte BLT COPY WORD parameter group for each fill operation includes an X-DESTINATION value, a Y-DESTINATION value, an X-SIZE value, a Y-SIZE value, and a FILL PATTERN.

The 16-bit X-DESTINATION and Y-DESTINATION values are signed offsets (in pixels) from the pixel specified by the X-ORIGIN and Y-ORIGIN values. The GCP adds the (X,Y) destination values to the (X,Y) origin values to specify the location of the first destination pixel in a fill operation. (The (X,Y) origin values are themselves offsets.)

The 16-bit X-SIZE and Y-SIZE values define, in pixels, the width and height of the filled area. The 16-bit FILL PATTERN is a bit pattern that the GCP uses to fill the target rectangle within

the destination buffer. The GCP begins filling each line of the target rectangle with the left-most bits of the FILL PATTERN. If necessary, the GCP truncates the pattern at the right edge of the target region.

The SET DESTINATION BUFFER command specifies the area of memory in which the BLT COPY WORD command can create filled rectangles. The flags set with the SET TRANSFER MODE command determine the type of transfer.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine if this operation results in a collision or a clip.

(CX) is N, the number of filled rectangles to be created with this command.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the PARAMETER POINTER, which is the address of the first byte of the first 10-byte BLT COPY WORD parameter group. The BLT COPY WORD parameter groups can be consecutive or in the form of a linked list (see the SET PARAM BLOCK MODE command description). The GCP services each BLT COPY WORD parameter group in sequence until it performs the number of fill operations specified in the BLT parameter group count.

Each 10-byte parameter group contains the following parameters (listed by their decimal offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	FILL PATTERN word
+ 2	X-DESTINATION offset from X-ORIGIN in pixels
+ 4	Y-DESTINATION offset from Y-ORIGIN in pixels
+ 6	X-SIZE width of filled rectangle in pixels
+ 8	Y-SIZE height of filled rectangle in pixels
+10	Optional: OFFSET to the next parameter group. This OFFSET must be included if SET PARAM BLOCK MODE is used to specify the linked list mode. This OFFSET must <u>not</u> be included when SET PARAM BLOCK

Byte Offset	Parameter
	MODE is used to specify the contiguous parameter block representation mode. An OFFSET value of OFFFFH indicates the end of the list. BLT COPY WORD terminates on the first occurrence of N BLTs (N is the number in (CX)) or a OFFFFH link OFFSET value.

Output
Parameters: None. All registers remain unchanged.

BLT POLYPOINT -- Int. EFH, (AH) = 0CH

Function: Draws a collection of points of the same color into the destination buffer.

Description: The BLT POLYPOINT command draws a collection of points at locations that the user specifies as a list of coordinates.

The BLT POLYPOINT command includes a BLT ID number, a destination buffer X-ORIGIN and Y-ORIGIN, a color index value, the number of points to be drawn (N), and a pointer.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine if this BLT POLYPOINT command causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are signed offsets (in pixels) from the pixel occupying the upper left corner of the destination buffer. The X-ORIGIN and Y-ORIGIN values are positive relative to this pixel; all point coordinates are relative to X-ORIGIN and Y-ORIGIN.

The color index is an index into the color palette that selects the color for all points drawn in a single BLT POLYPOINT command. (See the SET PALETTE command for more color information.)

The 16-bit coordinate set count N specifies the number of points to be drawn. This value should not exceed 16 if the user intends to use the GCP WAIT command to poll for collision and clipping information. BLT POLYPOINT does nothing if N is 0.

The pointer in the BLT POLYPOINT command is the address of the first byte of the first 4-byte coordinate set. Each coordinate set specifies a signed 16-bit X-coordinate value and a signed 16-bit Y-coordinate value. The X and Y coordinates are described in pixels.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns if a collision or a clip occurs.

(CX) is N, the number of points to be drawn.

(DH) is the color index for this collection of points.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the address of the first 4-byte coordinate set. The coordinate sets must be consecutive. The GCP draws each point in sequence until it draws N points.

Each 4-byte coordinate set contains the following parameters (listed by their offset from the beginning of the coordinate set):

Byte Offset	Parameter
+ 0	X coordinate (in pixels)
+ 2	Y coordinate (in pixels)

Output Parameters: None. All registers remain unchanged.

BLT POLYLINE -- Int. EFH, (AH) = 0DH

Function: Draws a collection of straight lines of the same color into the destination buffer.

Description: The BLT POLYLINE command draws a collection of lines in paired or chained format. The chained format uses the end of one line segment as the beginning of the next line segment. The paired format uses a pair of point coordinates to define an isolated line segment. The list of user-defined point coordinates resides in memory.

BLT POLYLINE draws horizontal and vertical line segments much faster than diagonal line segments. Horizontal and vertical line segments require only one call to the GCP. All other line segments, however, require a calculation (using Bresenham's algorithm) and a call to the GCP for each pixel in the line segment.

The BLT POLYLINE command includes a BLT ID number, a destination buffer X-ORIGIN and Y-ORIGIN, a color index value, a parameter group count (N), a pointer to the first 4-byte parameter group, and a chained/paired mode specification.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine if this BLT POLYLINE command causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are signed offsets (in pixels) from the pixel occupying the upper left corner of the destination buffer. The X-ORIGIN and Y-ORIGIN values are positive relative to this pixel; all point coordinates are relative to X-ORIGIN and Y-ORIGIN.

The color index is an index into the color palette that selects the color for all points drawn in a single BLT POLYLINE command. (See the SET PALETTE command description for more color information.)

The 16-bit parameter group count N specifies the number of parameter groups in the list. BLT POLYLINE draws a point if N is 1 or if the endpoints of a segment are coincident.

The BLT POLYLINE command produces N/2 line segments in paired mode and N-1 line segments in chained mode. N should not specify more than 16 line segments if the user intends to use the GCP WAIT command to poll for collision and clipping information.

The pointer parameter for the BLT POLYLINE command is the address of the first byte of the first 4-byte parameter group. Each parameter group specifies a 16-bit X-coordinate value and a 16-bit Y-coordinate value. The X and Y coordinates are described in signed pixels.

The chained/paired mode specification determines the line drawing format. In chained mode the first two parameter groups specify the first line segment, the second and third parameter groups specify the second line segment, and so on. In paired mode the first two parameter groups specify the first line segment, the third and fourth parameter groups specify the second line segment, and so on.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine if a collision or a clip occurs.

(CX) is N, the number of parameter groups in the parameter group list.

(DH) is the color index for the collection of lines.

(DL) is 0 for chained mode or 1 for paired mode.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the address of the first byte of the first 4-byte parameter group. The parameter groups must be consecutive. The GCP draws each line in sequence until it draws either N-1 chained line segments or N/2 paired line segments.

Each 4-byte parameter group contains the following parameters (listed by their offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	X coordinate (in pixels)
+ 2	Y coordinate (in pixels)

Output Parameters: None. All registers remain unchanged.

SET COLLISION PATTERN -- Int. EFH, (AH) = 12H

Function: Defines the criteria for collision detection.

Description: The SET COLLISION PATTERN command specifies a bit pattern that indicates, when matched, that the preceding BLT operation caused a collision.

SET COLLISION PATTERN includes a pixel pattern and a don't-care bit mask.

The pattern parameter of this command defines the bit pattern that indicates a collision when it matches the bit pattern of a pixel in the destination buffer. The number of bits in the pattern is equal to the number of bits in a single pixel in the current display mode.

The don't-care bit mask parameter includes a 0 bit for each bit position in the pattern that the system should ignore when checking for collisions. A value of 0 for all bits in the mask causes the system to detect a collision after every BLT operation. The number of bits in the mask is equal to the number of bits in a single pixel in the current display mode.

Input Parameters:

(AL) is reserved. Must be zero.

(BL) is the bit pattern for collision detection.

(BH) is the don't-care bit mask.

Output Parameters:

(AX) is undefined.

All other registers remain unchanged.

GET COLLISION PATTERN -- Int. EFH, (AH) = 13H

Function: Returns the current criteria for collision detection.

Description: The GET COLLISION PATTERN command returns the pattern last set by the SET COLLISION PATTERN command. This bit pattern, when matched, indicates that the preceding BLT operation caused a collision.

GET COLLISION PATTERN includes a pixel pattern and a don't-care bit mask.

The pattern parameter of this command defines the bit pattern that indicates a collision. The collision bit pattern matches the bit pattern of a pixel in the destination buffer. The number of bits in the pattern is equal to the number of bits in a single pixel in the current display mode.

The don't-care bit mask parameter includes a 0 bit for each bit position in the pattern that the system should ignore when checking for collisions. A value of 0 for all bits in the mask causes the system to detect a collision after every BLT operation. The number of bits in the mask is equal to the number of bits in a single pixel in the current display mode.

Input Parameters: None.

Output Parameters:

(AH) is undefined.

(AL) is the last value of AL passed to SET COLLISION PATTERN. This value should be zero.

(BL) is the bit pattern for collision detection.

(BH) is the don't-care bit mask.

All other registers remain unchanged.

SET CLIP RECTANGLE -- Int. EFH, (AH) = 14H

Function: Specifies the clipping rectangle for all BLT operations.

Description: The SET CLIP RECTANGLE command defines the clipping rectangle. The clipping rectangle specifies the region of the destination buffer in which a BLT operation may display pixels. The system clips all points specified in a BLT operation that exceed the clipping rectangle.

The ROM BIOS software performs clipping before calling the GCP. The GCP then performs a BLT or draw to the destination buffer.

If the user does not specify a clipping rectangle, the default clipping rectangle is the same as the boundaries of the destination buffer.

The user can disable clipping. In this case, the user is responsible for guaranteeing that the GCP does not write outside the bounds of the destination buffer.

The SET CLIP RECTANGLE command includes X-LEFT, X-RIGHT, Y-TOP, and Y-BOTTOM parameters to define the boundaries of the clipping rectangle. These boundaries are specified in pixels. The boundaries are relative to location (0,0), the pixel at the upper left corner of the destination buffer.

For example, to define a clipping rectangle that includes the entire frame buffer in 320 x 200 mode, specify 0, 319, 0, 199 as X-LEFT, X-RIGHT, Y-TOP, Y-BOTTOM parameter values, respectively. If the user-defined boundaries of the clipping rectangle exceed the size of the destination buffer, the system truncates the clipping rectangle to match the boundaries of the destination buffer.

Input Parameters:

(SI) is the X-LEFT boundary (in pixels).

(DI) is the X-RIGHT boundary (in pixels).

(CX) is the Y-TOP boundary (in pixels).

(DX) is the Y-BOTTOM boundary (in pixels).

Output Parameters:

(AX) is undefined.

All other registers remain unchanged.

GET CLIP RECTANGLE -- Int. EFH, (AH) = 15H

Function: Returns the current clipping rectangle boundaries.

Description: The GET CLIP RECTANGLE command returns the current boundaries of the clipping rectangle. The clipping rectangle specifies the region of the destination buffer in which a BLT operation may display pixels. The system clips all points specified in a BLT operation that exceed the clipping rectangle.

The GET CLIP RECTANGLE command returns the X-LEFT, X-RIGHT, Y-TOP, and Y-BOTTOM parameters that define the boundaries of the clipping rectangle. These boundaries are specified in pixels. The boundaries are relative to location (0,0), the pixel at the upper left corner of the destination buffer.

Input Parameters: None.

Output Parameters:

(AX) is undefined.

(SI) is the X-LEFT boundary (in pixels).

(DI) is the X-RIGHT boundary (in pixels).

(CX) is the Y-TOP boundary (in pixels).

(DX) is the Y-BOTTOM boundary (in pixels).

All other registers remain unchanged.

SET COLLISION/CLIP/DONE DETECT -- Int. EFH, (AH) = 16H

Function: Enables and disables collision and clip detection and the GCP-task-done interrupt. Also specifies the address of the collision/clip/done interrupt service routine.

Description: The SET COLLISION/CLIP/DONE DETECT command enables and disables collision and clip detection and the GCP-task-done interrupt for future BLT operations. This command also specifies the address of the user-defined service routine for the interrupt that occurs when the system detects a collision, clip, or GCP-task-done condition.

Disabling collision/clip/done detection improves system performance. The user should disable collision detection whenever collisions are not possible or not important. The user should also disable the GCP-task-done interrupt when not needed. Disabling clipping, however, is highly discouraged. When clipping is disabled, it is possible for the GCP to write data outside the destination buffer, yielding unpredictable results (including potential system crashes).

The SET COLLISION/CLIP/DONE DETECT command includes two parameters: DETECT-ENABLE, to enable and disable detection modes, and ADDRESS, to point to a user-defined collision/clip/done interrupt service routine.

Three DETECT-ENABLE flags specify the enabled/disabled status of collision detect, clip detect, and GCP-task-done interrupt independently.

The ADDRESS pointer serves two functions. If the value of ADDRESS is zero, the system does not call a user-specified interrupt service routine when it detects a collision, clip, or GCP-task-done condition. If the value of ADDRESS is other than zero and detection is enabled, the system interrupt handler calls the service routine at ADDRESS when a collision, clip, or GCP-task-done condition occurs.

The system makes a collision interrupt call at the end of a BLT operation that caused a collision. The system makes a clip interrupt call when the operation of the current BLT command is started. The system calls the interrupt routine for each parameter group that causes a collision, clip, or GCP-task-done condition.

The system passes the following information to the user-defined interrupt service routine:

Parameter	Function
ID	Register AL returns the 8-bit BLT ID number that the user specified with the command that caused the collision or the clip.
N	Register CX returns the number of the parameter group that caused a collision, clip, or GCP-task-done condition to occur. (For BLT POLYLINE, N refers to the line segments. For BLT POLYGON, N is always 1.)
CLIP	Bit 3 of AH returns a 1 if a clip triggered the interrupt.
COLLISION	Bit 4 of AH returns a 1 if a collision triggered the interrupt.
DONE	Bit 6 of AH returns a 1 if a GCP-task-done triggered the interrupt.
	(The other AH bits are system flags.)

The system interrupt handler invokes the user-defined interrupt service routine with a FAR CALL instruction. Therefore, this routine must terminate with a FAR RET instruction instead of an IRET instruction.

The user's main program must disable the interrupt vector before terminating. Otherwise, a future collision or clip could cause the system to jump to a non-existent routine.

As an alternative to using interrupts, the user can employ the GCP WAIT command to poll for the collision/clip/done status after each BLT or draw command.

Input Parameters: (AL) specifies enabled/disabled statuses as follows:

Bit	Definition
0	is a 1 to enable clip detection or a 0 to disable clip detection.
1	is a 1 to enable collision detection or a 0 to disable collision detection.
2	is a 1 to enable the GCP-task-done interrupt or a 0 to disable the GCP-task-done interrupt.

(ES:[BX]) is the paragraph and offset address of the user-defined interrupt service routine that the system calls in the event of a collision, clip, or GCP-task-done condition. A value of (0:[0]) directs the system to not call a user-defined interrupt service routine when it detects a collision, clip, or GCP-task-done condition.

Output Parameters: (AX) is indeterminate.

All other registers remain unchanged.

GET COLLISION/CLIP/DONE DETECT -- Int. EFH, (AH) = 17H

Function: Returns the values last set with the SET COLLISION/CLIP/DONE DETECT command.

Description: The GET COLLISION/CLIP/DONE DETECT command returns the value of the DETECT-ENABLE flags and the ADDRESS pointer to the user-defined interrupt service routine.

The DETECT-ENABLE flags are three single-bit flags that regulate collision and clip detection and the GCP-task-done interrupt.

The ADDRESS pointer serves two functions. If the value of ADDRESS is zero, the system does not call a user-specified interrupt service routine when it detects a collision, clip, or GCP-task-done condition. If the value of ADDRESS is other than zero and detection is enabled, the system interrupt handler calls the service routine at ADDRESS when a collision, clip, or GCP-task-done condition occurs.

Input Parameters: None.

Output Parameters: (AH) is indeterminate.

(AL) specifies enabled/disabled status as follows:

Bit	Definition
0	is a 1 to enable clip detection or a 0 to disable clip detection.
1	is a 1 to enable collision detection or a 0 to disable collision detection.
2	is a 1 to enable the GCP-task-done interrupt or a 0 to disable the GCP-task-done interrupt.

(ES:[BX]) is the paragraph and offset address of the user-defined interrupt service routine that the system calls in the event of a collision, clip, or GCP-task-done condition. A value of (0:[0]) directs the system to not call a user-defined interrupt service routine when it detects a collision, clip, or GCP-task-done condition.

All other registers remain unchanged.

GCP WAIT -- Int. EFH, (AH) = 18H

Function: Returns the collision/clip status as soon as the GCP is not busy.

Description: The GCP WAIT command provides a method of obtaining the collision/clip status without the use of a user-defined collision/clip interrupt service routine. The system waits until the GCP is idle before returning the collision/clip status, enabling the user to synchronize program execution with the GCP, if desired.

The GCP WAIT command returns a BLT ID number, a COLLISION STATUS word, and a CLIP STATUS word.

The BLT ID number is the ID number that the user supplies for all BLT and draw operations. This number enables the user to determine which BLT operation caused the collision or clip.

The COLLISION STATUS word returns the collision status for the first 16 BLT operations within a single BLT command. Likewise, the CLIP STATUS word returns the clip status for the first 16 BLT operations within a single BLT command.

Because the status words contain only 16 bits, it is recommended (when using GCP WAIT) that BLT commands be restricted to 16 or fewer operations (parameter groups). The contents of the status words are unpredictable if the number of objects in a BLT operation exceeds 16.

For both of the status words, the first BLT operation sets bit 0, the second BLT operation sets bit 1, and so on. A bit value of 1 indicates that a collision or clip occurred during the BLT operation corresponding to that bit position. A bit value of 0 indicates that a collision or clip did not occur.

Input Parameters: None.

Output Parameters:

(AH) is undefined.

(AL) is the BLT ID of the BLT operation just completed.

(BX) is the COLLISION STATUS word.

(CX) is the CLIP STATUS word.

All other registers remain unchanged.

BLT POLYGON -- Int. EFH, (AH) = 19H

Function: Draws a filled polygon into the destination buffer.

Description: The BLT POLYGON command draws a filled polygon. A list of user-defined point coordinates in memory specifies the "corners" of the polygon. The BLT POLYGON command automatically completes the polygon by connecting the last point to the first point.

The BLT POLYGON filling algorithm fills the interior and the boundary of the polygon. This algorithm can fill any polygon whose boundary (ignoring horizontal lines) does not intersect any horizontal line at more than two points. For example, the algorithm can properly fill an upright hourglass-shaped polygon but not an hourglass-shaped polygon lying on its side.

The BLT POLYGON command includes a BLT ID number, a destination buffer X-ORIGIN and Y-ORIGIN, two color index values, a parameter group count (N), and a pointer to the first 4-byte parameter group.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine if this BLT POLYGON command causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are signed offsets (in pixels) from the pixel occupying the upper left corner of the destination buffer. The X-ORIGIN and Y-ORIGIN values are positive relative to this pixel; all point coordinates are relative to X-ORIGIN and Y-ORIGIN.

The BLT POLYGON command includes an even color index parameter and an odd color index parameter. The terms even and odd refer to the sum of the X- and Y-coordinate values for each pixel in the polygon. For example, a pixel at (5,5) uses the even color index because 5 + 5 = 10, an even value.

This technique is called dithering; it enables the user to combine two colors to create a third. Dithering is primarily for use with a monitor; it has unpredictable effects when used with a television. (See the SET PALETTE command description for more color information.)

The 16-bit parameter group count N specifies the number of parameter groups in the list. BLT POLYGON draws a point if N is 1 or if all point coordinates for the polygon are coincident. BLT POLYGON draws a line if N is 2 or if there are only two distinct point coordinates.

The pointer parameter for the BLT POLYGON command is the address of the first byte of the first 4-byte parameter group. Each parameter group specifies a 16-bit X-coordinate value and a 16-bit Y-coordinate value. The X and Y coordinates are described in signed pixels.

The algorithm for drawing polygons may draw some points more than once. This may cause undesirable effects when combined with exclusive-OR specifications.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine in the event of a collision or a clip.

(CX) is N, the number of parameter groups in the parameter group list.

(DH) is the even color index for the polygon.

(DL) is the odd color index for the polygon.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the address of the first byte of the first 4-byte parameter group. The parameter groups must be consecutive.

Each 4-byte parameter group contains the following parameters (listed by their offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	X coordinate (in pixels)
+ 2	Y coordinate (in pixels)

Output Parameters: None. All registers remain unchanged.

BLT FILLED ELLIPSES -- Int. EFH, (AH) = 1AH

Function: Draws one or more filled ellipses or ellipse sectors.

Description: The BLT FILLED ELLIPSES command draws a series of filled ellipses or elliptical sectors. The ellipses (or the ellipses on which the elliptical sectors are based) can be oriented either horizontally or vertically within the destination buffer (see Figure 4-4).

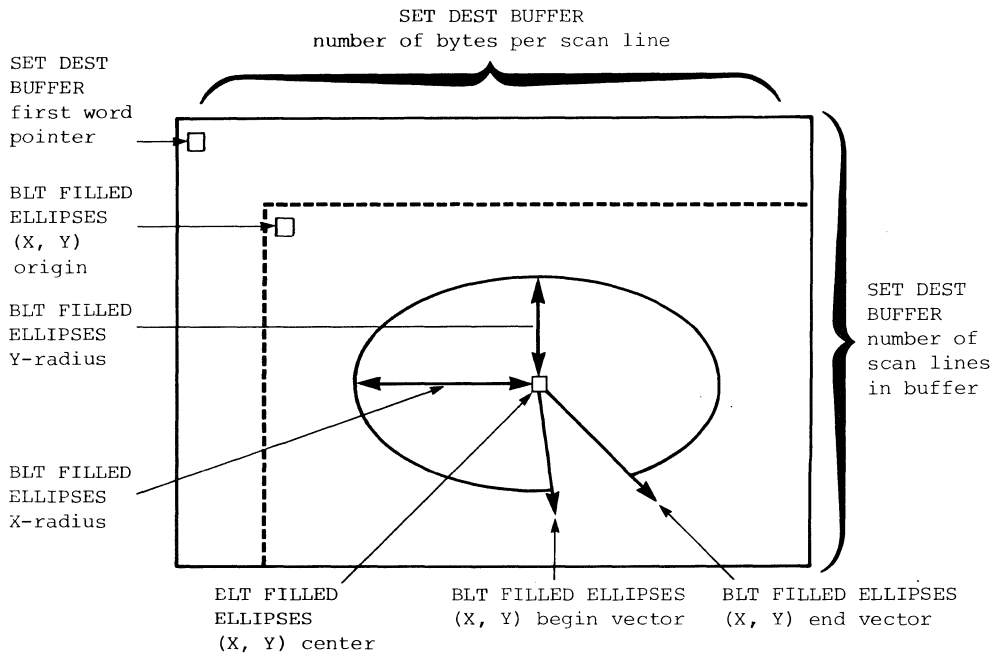


Figure 4-4: Operation of BLT FILLED ELLIPSES command

The BLT FILLED ELLIPSES command includes a BLT ID number, a destination buffer X-ORIGIN and Y-ORIGIN, two color index values, a parameter group count (N), a filled/hollow/segment mode specification, and a pointer to the first 16-byte parameter group.

The BLT ID number is a user-defined 8-bit value. The GCP returns this ID number to the user's interrupt routine if this BLT FILLED ELLIPSES command causes a clip or a collision.

The 16-bit X-ORIGIN and Y-ORIGIN values are offsets (in pixels) from the pixel occupying the upper left corner of the destination buffer. The X-ORIGIN and Y-ORIGIN values are positive relative to this pixel, and all elliptical coordinates are relative to X-ORIGIN and Y-ORIGIN.

As with the BLT POLYGON command, the BLT FILLED ELLIPSES command includes an even color index parameter and an odd color index parameter to permit dithering. The terms even and odd refer to the sum of the X- and Y-coordinate values for each pixel in the ellipse. Dithering is primarily for use with a monitor; it has unpredictable effects when used with a television. (See the SET PALETTE command description for more color information.)

The 16-bit parameter group count N specifies the number of parameter groups in the list. The value of N should not exceed 16 ellipses if the user intends to use the GCP WAIT command to poll for collision and clipping information.

The pointer parameter for the BLT FILLED ELLIPSES command is the address of the first byte of the first 16-byte parameter group. Each parameter group includes an X-CENTER, a Y-CENTER, an X-RADIUS, a Y-RADIUS, an X-BEGIN, a Y-BEGIN, an X-END, and a Y-END. All X and Y values are in signed pixels. X-CENTER and Y-CENTER are relative to X-ORIGIN and Y-ORIGIN.

The X-CENTER and Y-CENTER parameters specify the center of the ellipse. The X-RADIUS and Y-RADIUS parameters specify the magnitude of the horizontal and vertical dimensions of the ellipse, respectively. The GCP converts negative radius values to positive values. BLT FILLED ELLIPSES draws a point if both the X-RADIUS and Y-RADIUS values are 0. BLT FILLED ELLIPSES draws a line if either the X-RADIUS or Y-RADIUS value is 0.

The X-BEGIN, Y-BEGIN, X-END, and Y-END parameters specify two points through which the system draws radial vectors. The system draws a "begin-arc" vector from (X-CENTER, Y-CENTER) through (X-BEGIN, Y-BEGIN) and an "end-arc" vector from (X-CENTER, Y-CENTER) through (X-CENTER + X-BEGIN, Y-CENTER + Y-BEGIN). The magnitudes of these vectors are not important.

The begin-arc and end-arc vectors determine whether the BLT FILLED ELLIPSES routine uses an intermediate circle to draw an ellipse or not. If the circle is used, the routine adjusts the points of the circle so that they fall on the ellipse. Thus, the begin-arc and end-arc vectors determine the bounds of the ellipse indirectly through the definition of an intermediate circle.

The BLT FILLED ELLIPSES command defines an arc sector as the "pie-shaped" segment created by sweeping visually clockwise from the begin-arc vector to the end-arc vector. (Mathematically, the sweep is counterclockwise in a left-handed coordinate system.)

BLT FILLED ELLIPSES draws the entire ellipse if the begin-arc vector and the end-arc vector are the same. BLT FILLED ELLIPSES also draws the entire ellipse if X-BEGIN and Y-BEGIN are 0 or if X-END and Y-END are 0. The results are undefined if the begin and end-arc vectors point in the same direction but have unequal lengths.

Due to the "nonsquare" aspect ratio of the screen, equating the X-RADIUS and the Y-RADIUS of the ellipse does not produce a circle. For example, to create a circle in 320 x 200 mode, use a Y-RADIUS to X-RADIUS ratio of 5 to 6 (4/3 x Y/X = 4/3 x 200/320 = 5/6).

The algorithm for drawing ellipses may draw some points more than once. This may cause undesirable effects when combined with exclusive-OR specifications.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine if a collision or a clip occurs.

(CX) is N, the number of parameter groups in the parameter group list.

(DH) is the even color index for this collection of filled ellipses.

(DL) is the odd color index for this collection of filled ellipses.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the address of the first byte of the first 16-byte parameter group. The parameter groups must be consecutive. The GCP draws each ellipse in sequence until it draws N ellipses.

Each 16-byte parameter group contains the following parameters (listed by their decimal offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	X-CENTER (in pixels)
+ 2	Y-CENTER (in pixels)
+ 4	X-RADIUS (in pixels)
+ 6	Y-RADIUS (in pixels)
+ 8	X-BEGIN (in pixels)

Byte Offset	Parameter
+10	Y-BEGIN (in pixels)
+12	X-END (in pixels)
+14	Y-END (in pixels)
+16	Optional: OFFSET to the next parameter group. This OFFSET must be included if SET PARAM BLOCK MODE is used to specify the linked list mode. This OFFSET must not be included when SET PARAM BLOCK MODE is used to specify the contiguous parameter block representation mode. An OFFSET value of OFFFFH indicates the end of the list. BLT FILLED ELLIPSES terminates on the first occurrence of N BLTs (N is the number in (CX)) or a OFFFFH link OFFSET value.

Output Parameters: None. All registers remain unchanged.

BLT HOLLOW ELLIPSES -- Int. EFH, (AH) = 1BH

Function: Draws one or more hollow ellipses or elliptical arcs.

Description: The BLT HOLLOW ELLIPSES command draws a series of hollow ellipses or elliptical arcs. The ellipses (or the ellipses on which the arcs are based) can be oriented either horizontally or vertically within the destination buffer (see Figure 4-4 in the BLT FILLED ELLIPSES command description).

The BLT HOLLOW ELLIPSES command includes the same parameters as the BLT FILLED ELLIPSES command with the exception that all pixels are drawn with the color index in (DH)--(DL) is ignored.

Input Parameters:

(AL) is the user-specified 8-bit BLT ID number that the GCP returns to the user's interrupt routine if a collision or a clip occurs.

(CX) is N, the number of parameter groups in the parameter group list.

(DH) is the color index for this collection of hollow ellipses.

(SI) is the X-ORIGIN within the destination buffer.

(DI) is the Y-ORIGIN within the destination buffer.

(ES:[BX]) is the address of the first byte of the first 16-byte parameter group. The parameter groups must be consecutive. The GCP draws each ellipse in sequence until it draws N ellipses.

Each 16-byte parameter group contains the following parameters (listed by their decimal offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	X-CENTER (in pixels)
+ 2	Y-CENTER (in pixels)
+ 4	X-RADIUS (in pixels)
+ 6	Y-RADIUS (in pixels)
+ 8	X-BEGIN (in pixels)
+10	Y-BEGIN (in pixels)
+12	X-END (in pixels)
+14	Y-END (in pixels)

Byte Offset	Parameter
+16	Optional: OFFSET to the next parameter group. This OFFSET must be included if SET PARAM BLOCK MODE is used to specify the linked list mode. This OFFSET must not be included when SET PARAM BLOCK MODE is used to specify the contiguous parameter block representation mode. An OFFSET value of OFFFFH indicates the end of the list. BLT HOLLOW ELLIPSES terminates on the first occurrence of N BLTs (N is the number in (CX)) or a OFFFFH link OFFSET value.

Output Parameters: None. All registers remain unchanged.

SAVE GCP -- Int. EFH, (AH) = 1CH

Function: Saves the current state of the GCP in a user-specified area of memory.

Description: The SAVE GCP command enables the user to save the state of the GCP before the GCP is used by an interrupt service routine.

The user should employ the SAVE GCP command when using the GCP in normal and interrupt routines simultaneously. The user-specified data area should be on the stack to ensure that all routines can return normally.

The SAVE GCP command includes one output parameter and two input parameters. The output parameter returns the current size of the BLT data.

The first input parameter is a flag bit. If this flag bit is 1, the system saves the data and returns the size of the data. If this flag bit is 0, the system does not save the data, but still returns the size of the data.

The second input parameter is a user-specified pointer to the area in which the system saves the GCP data. This area is normally on the stack.

Input Parameters:

(AL) is 0 to specify that the BLT data is not to be saved. Alternatively, (AL) is 1 to specify that the data is to be saved.

(ES:[BX]) is the address of the first byte of the area that receives the BLT data. The system ignores this address if (AL) is 0.

Output Parameters:

(AX) returns the size (in bytes) of the BLT data regardless of the value of (AL).

All other registers remain unchanged.

RESTORE GCP -- Int. EFH, (AH) = 1DH

Function: Restores the data previously saved by the SAVE GCP command.

Description: The RESTORE GCP command enables the user to restore the GCP to the state it was in before the GCP operations were interrupted.

The only parameter for the RESTORE GCP command is a pointer to the user-specified data area used in a previous SAVE GCP command.

Input Parameters:

(ES:[BX]) is the address of the first byte of the data area used in a previous SAVE GCP command.

Output Parameters:

(AX) is undefined.

All other registers remain unchanged.

FILL DEST BUFFER -- Int. EFH, (AH) = 1EH

Function: Fills the entire destination buffer with specified data.

Description: The FILL DEST BUFFER command fills the destination buffer word-by-word with the data in BX. This command provides a simple method of quickly clearing the screen.

The parameters set by the SET TRANSFER MODE and SET CLIP RECTANGLE commands do not affect the operation of the FILL DEST BUFFER command. This command always fills the entire destination buffer with the exact data in BX.

Input Parameters:

(BX)	is the data used to fill the destination buffer, word-by-word.

Output Parameters:

(AX)	is indeterminate.

All other registers remain unchanged.

SET FONT POINTER -- Int. EFH, (AH) = 1FH

Function: Sets the font information pointer used by the BLT STRING command.

Description: The SET FONT POINTER command specifies a pointer to a data area that determines font characteristics for the BLT STRING command. The data area includes the following font data:

Parameter	Description
Font Type	One byte wide. A value of 0 indicates a fixed font, a value of 1 indicates a proportional font.
Excess White	One byte wide. Specifies excess inter-character white space in pixels, from -128 to +127.
Nominal Wid	One word wide. Specifies the nominal character width in pixels. This parameter is unused if proportional font is specified.
Nominal Ht	One word wide. Specifies the nominal character height in pixels, which is also the raster height in pixels.
Bitmap Wid	One word wide. Specifies the bitmap raster width in bytes; the raster width must be an even number.
Bitmap Addr	One doubleword wide. Specifies the bitmap address, which must be word aligned.
First ASCII	One byte wide. Specifies the ASCII value (numerical equivalent) of the first character in the bitmap.
Last ASCII	One byte wide. Specifies the ASCII value (numerical equivalent) of the last character in the bitmap.

The remaining parameters are required only if the Font Type selected is proportional font. These parameters, which constitute a 4-byte block, are repeated for each of the characters defined in the bitmap. The number of characters in the bitmap is:

Last ASCII - First ASCII + 1

Parameter	Description
Bitmap Off	One doubleword wide. Specifies the offset (in pixels) of the character into the bitmap. Use a value of -1 if the character is not in the bitmap. (For example, use -1 for a blank character.)
Char Wid	One byte wide. Specifies the character width in pixels, from -128 to +127.
Char Ht	One byte wide. Specifies the character height in pixels, from -128 to +127. Char Wid and Char Ht are used to move to the next character position if Bitmap Off is -1.

Input Parameters: (ES:[BX]) is the address of the first byte of the data area.

Output Parameters: (AX) is indeterminate.

All other registers remain unchanged.

GET FONT POINTER -- Int. EFH, (AH) = 20H

Function: Returns a pointer to the current font information.

Description: The GET FONT POINTER command returns the pointer that was set using the SET FONT POINTER command. See SET FONT POINTER for a description of the font pointer and font data information.

Input Parameters: None.

Output Parameters: (AX) is indeterminate.

(ES:[BX]) is the pointer to the current font information.

All other registers remain unchanged.

BLT STRING -- Int. EFH, (AH) = 21H

Function: Transfers a character string to the destination buffer.

Description: The BLT STRING command transfers a character string into the destination buffer as directed by the input parameters.

For each string, the upper left (DH=0,2), upper right (DH=1), and lower left (DH=3) corner of the first character drawn is at location (X-DESTINATION + X-ORIGIN, Y-DESTINATION + Y-ORIGIN). Succeeding characters will be drawn at a location depending on the value of DH.

The SET FONT POINTER command must be called before using BLT STRING.

Input Parameters:

(AL) is the BLT identification number.

(CH) is the number (N) of character strings that (ES:[BX]) points to.

(CL) is the number of characters that are to be ignored at the beginning of each string.

(DH) is the direction to draw each string on the screen:

(DH)	Direction
0	Left to right
1	Right to left
2	Top to bottom
3	Bottom to top

(DL) is the color specification. (The program expands this specification to a full word source AND mask.)

(SI) is the X-ORIGIN (in pixels).

(DI) is the Y-ORIGIN (in pixels).

(ES:[BX]) is the pointer to a list of string descriptor blocks, each 10 or 12 bytes long. Each descriptor block contains the following parameters (listed by their decimal offset from the beginning of the parameter group):

Byte Offset	Parameter
+ 0	X-DESTINATION in pixels (based on X-ORIGIN)
+ 2	Y-DESTINATION in pixels (based on Y-ORIGIN)

Byte Offset	Parameter
+ 4	Number of characters in the string
+ 6	ADDRESS OFFSET of the first character
+ 8	ADDRESS SEGMENT of the first character
+10	Optional: ADDRESS OFFSET of the next parameter field. This OFFSET must be included if SET PARAM BLOCK MODE is used to specify the linked list mode. This OFFSET must not be included when SET PARAM BLOCK MODE is used to specify the contiguous parameter block representation mode. An OFFSET value of OFFFFH indicates the end of the list. BLT STRING terminates on the first occurrence of N BLTs (N is the number in (CH)) or a OFFFFH link OFFSET value.

Output Parameters: None. All registers remain unchanged.

SET PARAM BLOCK MODE -- Int. EFH, (AH) = 22H

Function: Specifies contiguous or linked list representation for parameter blocks used by BLT COPY, BLT COPY WORD, BLT STRING, and BLT ELLIPSES.

Description: The SET PARAM BLOCK MODE command enables the user to specify contiguous or linked list representation for the parameter blocks required by BLT COPY, BLT COPY WORD, BLT STRING, and BLT ELLIPSES. The parameter block mode is reset to contiguous representation whenever SET SCREEN MODE is called.

Input Parameters:

(AL)	is 0 to specify contiguous parameter blocks or 1 to specify linked list parameter blocks.

Output Parameters:

(AX)	is indeterminate.

All other registers remain unchanged.

GET PARAM BLOCK MODE -- Int. EFH, (AH) = 23H

Function: Returns the current parameter block representation mode.

Description: The GET PARAM BLOCK MODE command returns the current parameter block representation mode--contiguous or linked list. The SET PARAM BLOCK MODE specifies which mode is in effect, except that the parameter block mode is reset to contiguous whenever SET SCREEN MODE is called.

Input Parameters: None.

Output Parameters:

(AH) is indeterminate.

(AL) is 0 for contiguous parameter blocks or 1 for linked list parameter blocks.

All other registers remain unchanged.

GET GCP STATUS -- Int. EFH, (AH) = 24H

Function: Returns the current GCP status word.

Description: The GET GCP STATUS command returns status bits that indicate whether the GCP is idle, whether vertical flyback (late VBLANK) is occurring, and whether a collision was detected during the last GCP task.

Input Parameters: None.

Output Parameters: (BX) is the GCP status word. The GCP status word bit flags are as follows:

Bit(s)	Definition
0	is a 1 if the GCP is idle (task done) or a 0 otherwise.
1	is a system flag.
2	is a 1 if a collision was detected during the last GCP task or a 0 otherwise.
3-15	are unused.

All other registers remain unchanged.

GET CHAR BITMAP ADDR -- Int. EFH, (AH) = 25H

Function: Returns the addresses of the two system character bitmaps.

Description: The GET CHAR BITMAP ADDR command returns the addresses of the two system character bitmaps. One address points to the bitmap for ASCII values 0 through 127 and the other address to ASCII values 128 through 255.

Input Parameters: None.

Output Parameters:

(ES) is the address segment for both bitmaps.

(SI) is the starting address offset for the bitmap covering ASCII values 0 through 127.

(DI) is the starting address offset for the bitmap covering ASCII values 128 through 255.

All other registers remain unchanged.

GET BLT MEMORY BOUNDS -- Int. EFH, (AH) = 26H

Function: Returns the bounds of memory available for block transfers (BLTs).

Description: The GET MEMORY BOUNDS command returns the bounds of the memory available for block transfers (BLTs). The bounds returned consist of the beginning and ending addresses of memory available for block transfers. The addresses returned may not reflect the actual memory configuration of your system; GET MEMORY BOUNDS may return an address for memory that is not actually installed in your system.

Input Parameters: None.

Output Parameters:

(AL) is 0 if all memory can be used for block transfers or 1 if only part of memory can be used for block transfers.

(SI) is the segment address of the first 64K byte segment that can be used for block transfers. (SI) is 0000H if all memory can be used for block transfers.

(DI) is the segment address of the last 64K byte segment that can be used for block transfers. (DI) is 0FFFFH if all memory can be used for block transfers.

All other registers remain unchanged.

Section 5

CUSTOM SOUND-PROCESSOR OPERATION

OVERVIEW OF SOUND-PROCESSOR OPERATION

The custom sound-processor (CSP) of the Mindset Personal Computer provides sound for both music and game play. The CPU directs CSP operations by placing values in the registers of the CSP.

The optional Stereo Sound Module contains a second CSP that is identical to the on-board CSP. This second CSP operates concurrently with the on-board CSP.

The Mindset ROM BIOS provides four commands to enable the user to control the on-board CSP and the optional Stereo Sound Module.

SOUND-PROCESSOR OPERATING MODES

Table 5-1 lists the four operating modes of the CSP.

Table 5-1: Custom Sound-Processor Operating Modes

Mode Number	Mode Name	Characteristics
1	Music	Four musical voices with limited special effects
2	Sound Effects/ Music	Three voices with special music and noise effects
3	Max Voice	Six voices with limited controls
4	See-Through	Provides direct access to the digital-to-analog (D/A) converter

Music Mode Operation

The music mode (mode 1) produces high-quality musical notes in the form of a sine wave. Each of the four voices in music mode has its own registers to control volume, attack/decay, and amplitude modulation (AM). In addition, voices 1 and 2 each have their own frequency-modulation (FM) and frequency-ramping registers.

Sound Effects/Music Mode Operation

The sound effects/music mode (mode 2) provides three voices with enhanced sound effects capabilities. All three voices provide the functions of mode 1 but can also use modified sine waves to produce different tones. Voices 1 and 2 provide distortion and noise effects for game play.

Max Voice Mode Operation

The max voice mode (mode 3) provides six voices with a single volume level for all voices. This mode uses a ramped waveform rather than a sine wave so it does not provide the pure tones of modes 1 and 2. The max voice mode does not provide control over sound modulation.

See-Through Mode Operation

The see-through mode (mode 4) enables the 80186 CPU to control the digital-to-analog (D/A) converter of the CSP directly. The see-through mode is the most flexible of the four modes but it also requires the greatest amount of program overhead for operation.

After selecting the see-through mode, the user invokes the SOUND DATA command to transfer data to the D/A converter. This command requires approximately 22 microseconds to transfer data from the CSP register to the D/A converter. To compensate for this delay, the user should use the SOUND DATA command at intervals of 22 microseconds or greater.

The creation of some sound signals (such as speech) requires a sampling period shorter than that provided by the SOUND DATA command. In general, if an application requires a sampling period shorter than 50 microseconds, the application program should write sound data directly to the CSP input register (in see-through mode). To determine the address for this register, add 2 to the address of the CSP returned by the GET MODULE ID TABLE command.

The see-through mode sets the volume to one-half the maximum volume level of the CSP. The user can control the volume by multiplying or dividing the data before sending it to the CSP.

SINE TABLE ADDRESS MASK OPERATION

Mode 2 includes a sine table address mask register to provide variable waveforms for voices 1 and 2. The table is 256 bytes long with peaks at 40H and C0H. The default waveform is a sine wave.

To generate a sine wave, the CSP uses values from a lookup table called the sine table. The CSP continuously steps through the sine table and transfers the sine table values to the D/A converter (see Appendix G, "Sound Generator Sine-Value Table"). The D/A converter, in turn, produces the sine wave that is the basis for musical notes.

Before the CSP addresses the sine table to get the next value for the sine wave, it performs a logical AND between the table address and the sine table address mask. This address masking has no effect if the sine table address mask is FFH (the default value).

If the sine table address mask is a value other than FFH, the CSP skips certain values in the sine table. The missing values cause the CSP to produce a permutation of the basic sine wave. For example, a mask of 3FH produces an approximate ramp wave with a frequency four times that of the basic sine wave.

NOISE MASK OPERATION

Mode 2 includes a random noise register and noise mask registers for voices 1 and 2. The random noise register contains an 8-bit value produced by a random number generator. The random number generator changes its value at a frequency of approximately 10 KHz. Voices 1 and 2 each have an 8-bit mask register. The CSP performs the following steps to produce distorted sound:

1. Logically ANDs the contents of the noise register and the noise mask register for the voice being played.

2. Logically exclusive-ORs the masked noise with the high-order byte of the frequency control register for the voice being played.

3. Plays the distorted note.

Different masks produce various forms of noise and band-limited noise sounds.

CUSTOM SOUND-PROCESSOR REGISTERS

Each CSP contains 26 8-bit registers. The functions of these registers vary depending on the selected mode and enabled features. Some functions cannot be used simultaneously with the same voice because they use the same registers.

The SET REGISTERS command controls the setup of the CSP registers. Modulation control registers #1 and #2 enable and disable the different modulation features of the CSP.

Volume Control

Modes 1 and 2 include a separate volume control register for each voice. Mode 3 uses one volume control register to set the volume level for all six voices. Each volume control register contains 8 bits to provide 256 distinct volume levels. Sound control register value 00H specifies the lowest volume level and FFH specifies the highest volume level. The volume control is linear.

Frequency Control

Modes 1, 2, and 3 use frequency control registers to specify the sound pitch. The CSP includes a pair of 8-bit frequency control registers for each voice.

The combination of the two 8-bit frequency control registers provides a 16-bit frequency count. The pitch is proportional to the 16-bit frequency count for values in the range 0 through 7FFFH. Exceeding 7FFFH causes the pitch to decrease rather than increase. The lowest pitch is either 0 or FFFFH. Table 5-2 gives the conversion factors that determine the pitch from the frequency count:

Table 5-2: Frequency Control Conversion Factors

Mode	Pitch (Hz)
1	Frequency Count/6.5536
2	Frequency Count/8.0282
3	Frequency Count/6.5536

Modulation Control Registers

Modes 1 and 2 use two modulation control registers to enable and disable amplitude modulation (AM), frequency modulation (FM), frequency ramping, and attack/decay operation. Table 5-3 describes the function of each bit in the two modulation control registers.

Table 5-3: Modulation Control Registers

Modulation Control Register #1

Bit	Function	Applicable Modes
0	Voice 1 attack/decay enable	1, 2
1	Must be 0	
2	Voice 2 attack/decay enable	1, 2
3	Must be 0	
4	Voice 3 attack/decay enable	1, 2
5	Must be 0	
6	Voice 4 attack/decay enable	1
7	Must be 0	

Modulation Control Register #2

Bit	Function	Applicable Modes
0	Voice 1 AM enable	1, 2
1	Voice 2 AM enable	1, 2
2	Voice 3 AM enable	1, 2
3	Voice 4 AM enable	1
4	Voice 1 FM enable	1, 2
5	Voice 2 FM enable	1, 2
6	Voice 1 FM frequency ramp enable	1, 2
7	Voice 2 FM frequency ramp enable	1, 2

Attack/Decay Control

Modes 1 and 2 use attack control registers and decay control registers for each voice to control the attack and decay rates for each note. The attack rate is the rate at which the volume increases after the note begins and the decay rate is the rate at which the volume decreases to end the note. Modulation control register #1 selectively enables and disables attack/decay operation for each voice.

When a note begins, the CSP adds the value in the attack control register to the volume every 3.2 milliseconds until the next addition would result in an overflow. At this point, the system disables the attack and the decay begins. The CSP then subtracts the value in the decay control register from the volume every 3.2 milliseconds until the next subtraction would result in a negative value. At this point the CSP disables the decay and the volume remains at the current level.

The CSP disables attack/decay each time it plays a note, so the user must enable the attack/decay bit in modulation control register #1 before each note if attack/decay operation is desired for that note.

Attack/decay operation and AM use the same register and cannot be used together for the same voice.

AM Control

Modes 1 and 2 include an AM frequency register for each voice. The CSP performs AM by varying the volume of a voice in proportion to a second modulating sine wave. The modulating sine wave is usually of lower frequency than the voice it is modulating.

The AM frequency registers control the frequency of the modulating sine wave. The frequency is proportional to the 8-bit modulating frequency for values in the range 0 through 7FH. Values greater that 7FH produce decreasing modulation frequencies.

The following formula determines the modulating frequency from the value of the AM frequency register:

Modulating Frequency (Hz) = AM Frequency Value/0.8192

When the modulating sine wave is at its positive peak, it attenuates the volume to one-half of the maximum volume. When this sine wave is at its negative peak, it attenuates the volume to zero. Enabling the AM function causes the volume control registers to have no effect.

AM uses the same register as attack/decay operation and cannot be used together for the same voice.

FM Control

Modes 1 and 2 each include FM frequency registers for voices 1 and 2. The CSP performs FM by varying the frequency of a voice in proportion to a second modulating sine wave.

The modulating sine wave affects only the lower byte of the two-byte frequency count. The system varies this value from 0H to FFH according to the value of the modulating sine wave. The modulating frequency operates the same as with AM control. The following formula determines the modulating frequency from the value of the FM frequency register:

Modulating Frequency (Hz) = FM Frequency Value/0.8192

FM and FM frequency ramping use the same register and cannot be used together for the same voice.

FM Frequency Ramping

Modes 1 and 2 each include frequency ramp count registers for voices 1 and 2. Each voice has two 8-bit frequency ramp count registers that combine to provide a signed 16-bit ramp count. The CSP performs FM frequency ramping by adding this 16-bit ramp count to the frequency count for that voice every 3.2 milliseconds.

The ramping function increases frequency until the combined value of the ramp count and the frequency count reaches 7FFFH. When the total exceeds 7FFFH, the frequency decreases until the total reaches 0 (as described previously under "Frequency Control"). Leaving the FM frequency ramping enabled produces a continual upsweep and then downsweep through the frequency range.

FM frequency ramping and FM use the same register and cannot be used together for the same voice.

DESCRIPTIONS OF SOUND-PROCESSOR COMMANDS

The Mindset ROM BIOS includes four commands to control the operation of the CSP. The user invokes these commands with the software interrupt instruction using interrupt EEH. Before executing this interrupt, the user sets the AH register to select a command and sets other registers to supply the input parameters for the selected command. A list of the four CSP commands follows.

Command	Value of AH Decimal	Hex	Function
Set Sound Mode	36	24	Sets the operation mode of one or two CSPs.
Set Sound Registers	37	25	Sets the registers that control CSP operation.

Command	Value of Decimal	AH Hex	Function
Sound Data	38	26	Transfers sound data directly from the CPU to the D/A converter.
Stereo Module Check	39	27	Checks for the presence of a stereo module.

The operation of the optional stereo CSP is the same as that of the on-board CSP. Table 5-4 describes how register AL specifies one or both of the CSPs in the following commands.

Table 5-4: CSP Selection

Value of AL	Processor
1	On-board CSP
2	Optional stereo module CSP
3	Both CSPs

SET SOUND MODE -- Int. EEH, (AH) = 24H

Function: Sets the operation mode of one or two CSPs.

Description: The SET SOUND MODE command enables the user to select one of the four CSP modes.

Input Parameters:

(AL) selects one or both processors for a register update, as described in Table 5-4.

(BL) is the mode number as follows:

CSP Mode	(BL)
1	1
2	2
3	4
4	8

Output Parameters: None. All registers remain unchanged.

SET SOUND REGISTERS -- Int. EEH, (AH) = 25H

Function: Sets the registers that control CSP operation.

Description: The SET SOUND REGISTERS command sets up and plays from one to six simultaneous notes from the on-board CSP. This command can set up and play from one to 12 simultaneous notes on systems with the optional stereo module.

This command operates by setting bits in the BX (and/or CX) register to specify which CSP registers to update. Table 5-5 below describes which bits to set. The SET SOUND REGISTERS command includes a pointer to a table of values for the CSP registers.

The table for a single CSP is 26 bytes long. This table is 52 bytes long when the program is updating both the on-board and stereo CSPs with the same SET SOUND REGISTERS command.

Table 5-5 describes the contents and order of the table that supplies the register values. The information stored in these bytes varies depending on the way the CSP is being used.

Table 5-5: CSP Register Update Table

Table Byte	BX/CX Update Bit	CSP Register
0	0	Modulation control register #1
1	1	Modulation control register #2
2	2	Voice 1 frequency low byte
3	2	Voice 1 frequency high byte
4	3	Voice 2 frequency low byte
5	3	Voice 2 frequency high byte
6	4	Voice 3 frequency low byte
7	4	Voice 3 frequency high byte
8	5	Voice 4 frequency low word or voice 1 noise mask
9	5	Voice 4 frequency high word or voice 2 noise mask
10	6	Voice 1 volume

Table Byte	BX/CX Update Bit	CSP Register
11	7	Voice 2 volume
12	8	Voice 3 volume
13	9	Voice 4 volume
14	10	Voice 1 attack rate or voice 1 AM frequency or voice 5 frequency low byte
15	10	Voice 1 decay rate or voice 5 frequency high byte
16	11	Voice 2 attack rate or voice 2 AM frequency or voice 6 frequency low byte
17	11	Voice 2 decay rate or voice 6 frequency high byte
18	12	Voice 3 attack rate or voice 2 AM frequency
19	12	Voice 3 decay rate
20	13	Voice 4 attack rate or voice 4 AM frequency or voice 1 sine address mask
21	14	Voice 4 decay rate or voice 2 sine address mask
22	14	Voice 1 FM ramping rate low byte or voice 1 FM frequency
23	14	Voice 1 FM ramping rate high byte
24	15	Voice 2 FM ramping rate low byte or voice 2 FM frequency
25	15	Voice 2 FM ramping rate high byte

When programming the on-board and stereo module CSPs using this command, place the data for the stereo module directly after the data for the on-board CSP.

Input Parameters:

(AL)	selects one or both of the processors for a register update, as described in Table 5-4.
(BX)	updates the on-board CSP registers (see Table 5-5).
(CX)	updates the stereo module CSP registers. The bit definitions for register CX are the same as those of register BX for this command.
(ES: [SI])	is the address of the table of register data. This address points to the start of the data for the on-board CSP even if (AL) selects the stereo module CSP for updating.

Output Parameters: None. All registers remain unchanged.

SOUND DATA -- Int. EEH, (AH) = 26H

Function: Transfers sound data directly from the 80186 CPU to the D/A converter. This command operates only in the see-through mode (mode 4).

Description: The SOUND DATA command provides sound data for the see-through mode. The see-through mode bypasses the CSP and enables the user to operate the 8-bit D/A converter directly.

Input Parameters:

(AL) selects one or both processors to receive the sound data parameters. Table 5-4 describes the values that select each processor.

(BL) is the sound data for the on-board D/A converter.

(BH) is the sound data for the stereo D/A converter.

Output Parameters: None. All registers remain unchanged.

STEREO MODULE CHECK -- Int. EEH, (AH) = 27H

Function: Checks for the presence of a stereo module.

Description: The STEREO MODULE CHECK command enables the program to detect whether or not the system has the optional stereo module.

Input Parameters: None.

Output Parameters:

(AL)	is 2 if the stereo module is present or 0 if the stereo module is not present.

Section 6

KEYBOARD OPERATION

INTRODUCTION

The keyboard of the Mindset computer is laid out in the familiar typewriter keyboard format (see Figure 6-1). The keys to the right of the character keyboard are the cursor keys, and the keys above the character keyboard are the function keys. One of the unique Mindset function keys (SYS CONFIG) enables the operator to reconfigure the system.

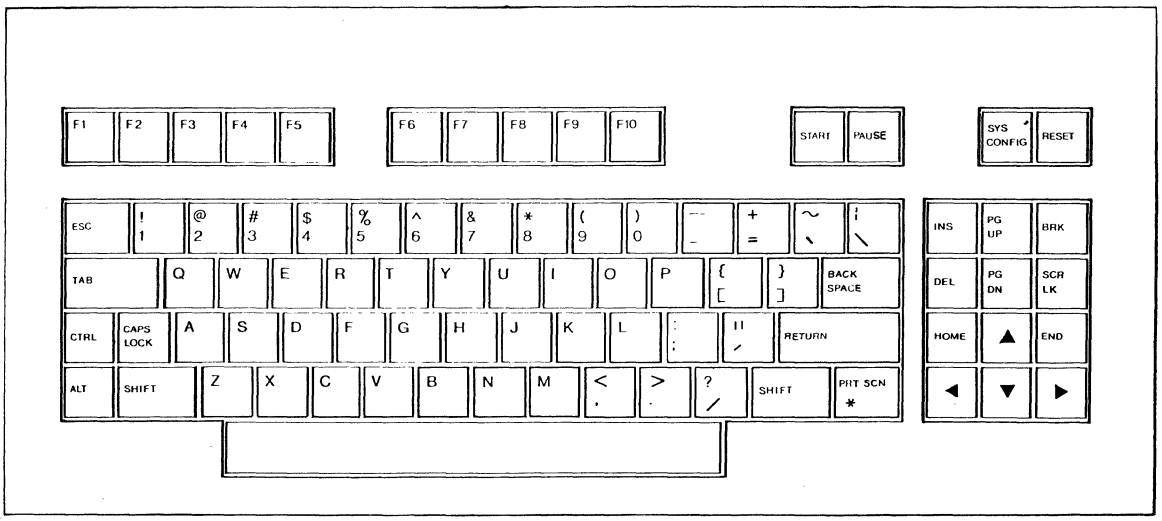


Figure 6-1: Mindset Keyboard

KEYBOARD COMPATIBILITY

The keyboard of the Mindset computer produces scan codes and extended scan codes that are compatible with those of the industry-standard personal computer keyboard. The differences between the two keyboards are as follows:

- The START and PAUSE keys of the Mindset computer provide additional extended scan codes 133 and 134, respectively. Operation of the START and PAUSE keys is application dependent.
- The ALT-SYS CONFIG key sequence displays the Mindset system configuration screen.
- The ALT-RESET key sequence of the Mindset keyboard provides the ALT-CTRL-DEL reset function of the industry-standard keyboard.

- There is no NUM LOCK key on the Mindset keyboard; the cursor keys are cursor keys only and do not produce numbers.
- The CTRL-PAUSE key sequence on the Mindset keyboard replaces the CTRL-NUMLOCK key sequence on an industry-standard keyboard.
- The SHIFT key only shifts up, unlike the industry-standard SHIFT key that shifts down if CAPS LOCK is set.

THE SYS CONFIG KEY

The ALT-SYS CONFIG key sequence does not provide a scan code. Instead, it automatically presents the Mindset system configuration screen (see Figure 6-2).

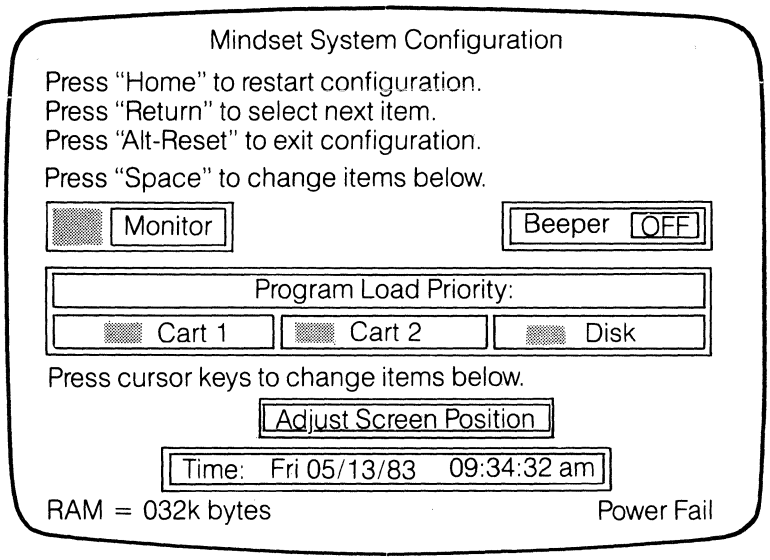


Figure 6-2: System configuration screen

The operator can perform the following functions using the configuration screen:

- Select either television or RGB color monitor operation.
- Enable or disable the system beeper.

- Establish the order in which the computer checks the cartridge slots and the disk drive for loadable programs.
- Adjust the screen position using the cursor keys so that all four sides of the frame buffer are visible.
- Set the time and date for the Mindset real-time clock.
- Check the RAM size.
- Check to see if power has been removed from the computer since the last time the ALT-SYS CONFIG key sequence was used.

The parameters set on the system configuration screen become the default parameters that the system uses when it is turned on. The system retains the system configuration parameters as long as the System Unit is plugged in. The ENABLE/DISABLE BEEPER, SET DISPLAY DEVICE, and SET SCREEN POSITION commands do not affect the default parameters, only the current values.

The following steps describe the operation of the configuration screen:

1. The first time the unit is powered up after it is plugged in, a red power-fail message appears in the lower right corner of the menu. (The initial default configuration is shown in Figure 6-2.)

2. The operator selects a configuration parameter for modification by pressing the RETURN key until the green cursor arrow points to the desired field. The HOME key returns the cursor arrow to the top of the configuration screen.

3. After selecting a parameter field from the top of the screen (TV/Monitor, Beeper, or Program Load Priority), the operator uses the space bar to step through the different values available for that parameter. After selecting a parameter field from the bottom half of the screen (Adjust Screen Position or Time), the operator uses the cursor keys to step through the different values available.

4. The individual parameter fields operate as follows:

    a. When the cursor arrow points to the TV/Monitor parameter field, the operator toggles between television and monitor operation using the space bar. The menu appears green, red, and white on a black background with a blue border when the operator selects the correct display device.

    b. When the cursor arrow points to the Beeper field, the operator uses the space bar to toggle between beeper enabled (ON) and beeper disabled (OFF). The choice of ON or OFF appears in reverse video green characters.

c. When the cursor arrow points to the number in one of the Program Load Priority fields, the operator steps through the available priorities using the space bar. After selecting the load priority for Cart 1, the operator can select the load priority for Cart 2 from the two remaining priorities. The system assigns the last remaining load priority to the Disk.

d. When the cursor arrow points to the Adjust Screen Position field, the operator can use the cursor keys to adjust the screen position to compensate for the overscan characteristics of the television. The up, down, right, and left cursor keys move the screen in those respective directions.

e. When the cursor arrow points to any part of the date and time parameter field, the system freezes the date and time. The operator uses the up and down cursor keys to increase or decrease the values of the date and time fields, respectively. The system begins automatically updating the date and time fields when the operator uses the HOME or the RETURN key to move the cursor arrow to another parameter field.

   Note: The system does not check the validity of the month and date combination. Selecting an invalid month and date combination causes the real-time clock to continue incrementing the date beyond a value of 31.

5. After the configuration parameters are complete, the operator uses the ALT-RESET key sequence to make the new selections the default configuration. Pressing this key sequence exits the configuration screen and resets the system.

Section 7

MINDSET REAL-TIME CLOCK OPERATION

OVERVIEW OF REAL-TIME CLOCK OPERATION

The real-time clock of the Mindset Personal Computer is not the same as the industry-standard time-of-day clock, although the time-of-day clock is also supported. The real-time clock maintains the current time (in 24-hour mode) and date as long as the Mindset computer is plugged in.

The real-time clock controls two timed events--the system wake-up and the real-time clock alarm interrupt. The system wake-up turns on the system, and the real-time clock alarm interrupt invokes a user-defined interrupt service routine.

The wake-up feature can be used in conjunction with a cartridge-based program or an auto-loading disk program to perform functions automatically--without operator initiation or intervention. Thus, the Mindset computer can be programmed to wake up, perform specified functions, and go back to sleep (using the POWER OFF command) at any time or even periodically.

DESCRIPTION OF REAL-TIME CLOCK COMMANDS

A listing of the 11 real-time clock commands follows. Access to these commands is through interrupt EEH. Each command is described in more detail in the remainder of this section.

Command	Value of AH Decimal	Hex	Function
RT Clock Enable Wake-up/Alarm	08	08	Enables/disables the real-time clock wake-up/alarm options.
RT Clock Get Wake-up/Alarm Status	09	09	Returns the enabled/disabled status of the real-time clock wake-up/alarm options.
RT Clock Set Status	10	0A	Sets or clears the valid/invalid status of the real-time clock.
RT Clock Get Status	11	0B	Returns the valid/invalid status of the real-time clock.
RT Clock Read Time	12	0C	Returns the time from the real-time clock.
RT Clock Set Time	13	0D	Sets the time on the real-time clock.

Command	Value of AL Decimal	AH Hex	Function
RT Clock Read Date	14	0E	Returns the date from the real-time clock.
RT Clock Set Date	15	0F	Sets the date on the real-time clock.
RT Clock Read Alarm	16	10	Returns the time of the alarm-interrupt setting from the real-time clock.
RT Clock Set Alarm	17	11	Sets the time for the alarm interrupt.
RT Clock Set Int	18	12	Sets the interrupt pointer for the user-specified alarm-interrupt service routine.
Reserved	19	13	N/A

Individual command descriptions follow.

RT CLOCK ENABLE WAKE-UP/ALARM -- Int. EEH, (AH) = 08H

Function: Sets the enabled/disabled status of the real-time clock wake-up option and alarm-interrupt.

Description: The RT CLOCK ENABLE WAKE-UP/ALARM command sets the wake-up enable and alarm-interrupt enable.

The wake-up event causes the system to power up and begin operation of a cart-based or auto-loading disk program. The alarm-interrupt causes the system to execute a user-defined interrupt service routine at a user-specified time. See the RT CLOCK SET INT command later in this section for information on setting the alarm-interrupt vector. The alarm-interrupt also causes the system to power up if both the wake-up and alarm-interrupts are enabled.

Input Parameters: (AL) sets the real-time clock wake-up/alarm enabled/disabled status according to the following bit definitions:

Bit	Function
0	is unused.
1	is a 1 to enable the wake-up interrupt or a 0 to disable the wake-up interrupt.
2	is a 1 to enable the alarm-interrupt or a 0 to disable the alarm-interrupt.

Output Parameters: None. All registers remain unchanged.

RT CLOCK GET WAKE-UP/ALARM STATUS -- Int. EEH, (AH) = 09H

Function: Returns the current wake-up/alarm status of the real-time clock.

Description: The RT CLOCK GET WAKE-UP/ALARM STATUS command returns the information set by the RT CLOCK ENABLE WAKE-UP/ALARM command.

Input Parameters: None.

Output Parameters: (AL) is the real-time clock wake-up/alarm status as specified by the following bit definitions:

Bit	Function
0	is unused.
1	is a 1 if the wake-up interrupt is enabled or a 0 if the wake-up interrupt is disabled.
2	is a 1 if the alarm interrupt is enabled or a 0 if the alarm interrupt is disabled.

RT CLOCK SET STATUS -- Int. EEH, (AH) = 0AH

Function: Sets or clears the valid/invalid time status.

Description: The RT CLOCK SET STATUS command enables the user to clear the bit which indicates that the time and date settings are invalid. This bit is set when the system is plugged in. The user program should clear this bit after the time and date are set.

Input Parameters: (AL) is 0 to indicate that the time is valid or 1 to indicate that the time is invalid.

Output Parameters: None. All registers remain unchanged.

RT CLOCK GET STATUS -- Int. EEH, (AH) = 0BH

Function: Returns the valid/invalid status of the real-time clock.

Description: The RT CLOCK GET STATUS command enables the user to determine if the system has been unplugged since the real-time clock was last set.

Input Parameters: None.

Output Parameters:

(AH)	is 0 if the time is valid or 1 if the time is invalid.

All other registers remain unchanged.

RT CLOCK READ TIME -- Int. EEH, (AH) = 0CH

Function: Returns the time from the real-time clock.

Description: The RT CLOCK READ TIME command returns the current time from the real-time clock. The user should check the status of the real-time clock with the RT CLOCK GET STATUS command to be sure the time is valid.

Input Parameters: None.

Output Parameters:

(BL) is the number of seconds (0 to 59).

(BH) is the number of minutes (0 to 59).

(CL) is the number of hours (0 to 23).

All other registers remain unchanged.

RT CLOCK SET TIME -- Int. EEH, (AH) = 0DH

Function: Sets the time on the real-time clock.

Description: The RT CLOCK SET TIME command sets the time on the real-time clock. Unplugging the system causes the invalid time bit to be set when the system is plugged in again. The RT CLOCK SET STATUS command is used to clear this bit after the time and date are reset (see the RT CLOCK SET STATUS command earlier in this section).

Input Parameters:

	(BL)	is the number of seconds (0 to 59).
	(BH)	is the number of minutes (0 to 59).
	(CL)	is the number of hours (0 to 23).

Output Parameters: None. All registers remain unchanged.

RT CLOCK READ DATE -- Int. EEH, (AH) = 0EH

Function: Returns the date from the real-time clock.

Description: RT CLOCK READ DATE returns the current date and day of the week from the real-time clock. The year returned by this command is a two-digit number from 0 to 99. The user should check the status of the real-time clock with the RT CLOCK GET STATUS command to be sure the date is valid.

Input Parameters: None.

Output Parameters:

(BL) is the day of the week (1 to 7).

(BH) is the day of the month (1 to 31).

(CL) is the month (1 to 12).

(CH) is the year (0 to 99).

All other registers remain unchanged.

RT CLOCK SET DATE -- Int. EEH, (AH) = 0FH

Function: Sets the date on the real-time clock.

Description: The RT CLOCK SET DATE command sets the date and day of the week on the real-time clock.

Input Parameters:

(BL)	is the day of the week (1 to 7).
(BH)	is the day of the month (1 to 31).
(CL)	is the month (1 to 12).
(CH)	is the year (0 to 99).

Output Parameters: None. All registers remain unchanged.

RT CLOCK READ ALARM -- Int. EEH, (AH) = 10H

Function: Returns the time of the alarm-interrupt setting from the real-time clock.

Description: The RT CLOCK READ ALARM command returns the current time setting for the real-time clock alarm. The real-time clock alarm invokes the user-defined interrupt routine specified by the RT CLOCK SET INT command described later in this section.

Input Parameters: None.

Output Parameters:

(BL) is the number of seconds (0 to 59).

(BH) is the number of minutes (0 to 59).

(CL) is the number of hours (0 to 23).

All other registers remain unchanged.

RT CLOCK SET ALARM -- Int. EEH, (AH) = 11H

Function: Sets the time for the alarm interrupt.

Description: The RT CLOCK SET ALARM command sets the time for the real-time clock alarm-interrupt. The real-time clock alarm invokes the user-defined interrupt routine specified by the RT CLOCK SET INT command described next.

Input Parameters:

(BL) is the number of seconds (0 to 59).

(BH) is the number of minutes (0 to 59).

(CL) is the number of hours (0 to 23).

Output Parameters: None. All registers remain unchanged.

RT CLOCK SET INT -- Int. EEH, (AH) = 12H

Function: Sets the address of the user-specified alarm-interrupt service routine.

Description: The RT CLOCK SET INT command sets the interrupt pointer for the real-time clock alarm. The default value of this pointer is zero. If the value is zero, the system does not call a user-defined alarm-interrupt service routine.

The system interrupt handler invokes all user-specified interrupt routines with a FAR CALL instruction. Therefore, the user should terminate these routines with a RETURN instruction and specify the routine as a FAR procedure to the assembler.

Input Parameters: (ES:[BX]) is the address of the user-specified interrupt service routine.

Output Parameters: None. All registers remain unchanged.

Section 8

RAM CARTRIDGE OPERATION

OVERVIEW OF RAM CARTRIDGE OPERATION

A RAM cartridge, or cart, can serve as additional random-access memory, as an executable program, or as a file-structured device. A ROM/RAM cartridge can perform the same functions as a RAM cartridge. A ROM cartridge, however, can contain only executable programs or read-only files.

The ROM BIOS provides commands for formatting RAM cartridges, reading the directory, writing the directory, creating files, reading files, writing files, and deleting files. The ROM BIOS also provides a command to return the status and type of ROM and RAM cartridges.

The ROM BIOS does not provide much error checking for RAM cartridge commands. The results are unpredictable if a RAM cartridge is replaced between commands. The application program should always check the cartridge directory to be sure that the subsequent command performs as anticipated.

A RAM cartridge may contain one of two different types of memory, each type having its own response time. The Mindset Personal Computer uses wait states to accommodate the cartridge memory read/write response times. The ROM/RAM CART STATUS command, which returns the cartridge status, also sets the computer for the number of wait states (0 or 1) required for the type of memory in the cartridge. For this reason, the ROM/RAM CART STATUS command must be executed before using any cartridge.

On power-up, the ROM BIOS performs a ROM/RAM CART STATUS command before booting from a cartridge.

The operation of RAM cartridges is based on memory blocks within the cartridge. The size of a block is defined as 128, 256, or 512 bytes per block. All blocks within a single RAM cartridge are the same size.

Table 8-1 describes the format of a directory entry for a file-structured RAM cartridge. This manual recommends that you use the MS-DOS file directory format for file attributes, date and time of last access, file name, and file extension.

Table 8-1: File-Structured RAM Cartridge Directory Entry Format

Directory Entry Byte Number	Function
0 - 7	File name (8 characters)
8 - 10	File name extension (3 characters)
11	File attribute (application-defined)
12,13	Date of last access
14,15	Time of last access
16,17	Number of bytes in the last block in the file
18	Number of blocks in the file
19	Start block number for the file

The first 18 bytes of directory information (bytes 0 through 17) provide file information. These bytes can change without changing the size of the file; they can be updated directly by the RAM CART PUT ENTRY command.

The ROM BIOS uses bytes 18 and 19 for locating file blocks. These bytes cannot be changed directly through a ROM BIOS command. Instead, the number of blocks (byte 18) is changed indirectly by the RAM CART WRITE and RAM CART DELETE BLOCK commands. The start block number (byte 19) is changed indirectly by the first RAM CART WRITE command used for that file or by a RAM CART DELETE BLOCK command, the latter of which deletes the first block in the file.

Table 8-2 describes the format of a file-structured RAM cartridge.

Table 8-2: File-Structured RAM Cartridge Format

Cartridge Byte Number	Function
0 - 7	Format specification
8	Number of blocks - 1 (= n)
9	Block size as follows:
	Byte Value: 7 → Block Size: 128; 8 → 256; 9 → 512
10 to (n + 9)	File allocation table (one byte per block)
(n + 10)	Number of files (= m)
(n + 12) to (n + 11 + 20 * m)	File entries

The following procedures describe how to create a RAM cartridge file:

1. Call the ROM/RAM CART STATUS command to set the number of wait states for this type of RAM cartridge. (This step is required only when a new cartridge is inserted.)

2. Call the RAM CART GET ENTRY command until an entry is found that is not a valid file.

3. Call the RAM CART PUT ENTRY command to name the file and add time and date information.

4. Call the RAM CART WRITE command to allocate blocks within the file and to write data in the file.

The following procedures describe how to read a RAM cartridge file:

1. Call the ROM/RAM CART STATUS command to set the number of wait states for this type of RAM cartridge. (This step is required only when a new cartridge is inserted.)

2. Call the RAM CART GET ENTRY command to find a file to read.

3. Call the RAM CART READ command to read data from blocks within the file.

The following procedures describe how to delete a RAM cartridge file:

1. Call the ROM/RAM CART STATUS command to set the number of wait states for this type of RAM cartridge. (This step is required only when a new cartridge is inserted.)

2. Call the RAM CART GET ENTRY command to find a file to delete.

3. Call the RAM CART DELETE BLOCK command and specify a start block of 0 and a block count of 0.

4. Call the RAM CART PUT ENTRY command to replace the name of the deleted file with an illegal name and/or put an unused attribute in the file directory to mark the file as unused. The ROM OS does not define any names as illegal. The user's file OS must define illegal names.

DESCRIPTION OF RAM CARTRIDGE OPERATION COMMANDS

A list of RAM cartridge operation commands follows. Access to these commands is through interrupt EEH. Each command is described in more detail in the remainder of this section.

Command	Value of AH Decimal	Hex	Function
ROM/RAM Cart Status	20	14	Returns the operational data for the cartridge and reads the cartridge wait state specification into the system.
RAM Cart Format	21	15	Formats a RAM cartridge to accept information and support a directory.

Command	Value of AH Decimal	Hex	Function
RAM Cart Get Entry	22	16	Returns RAM cartridge file entry information from the cartridge directory.
RAM Cart Put Entry	23	17	Writes RAM cartridge file entry information to the cartridge directory.
RAM Cart Read	24	18	Reads blocks of data from a RAM cartridge file.
RAM Cart Write	25	19	Writes blocks of data to a RAM cartridge file.
RAM Cart Delete Block	26	1A	Deletes blocks of data from a RAM cartridge file.
Reserved	27	1B	N/A
Reserved	28	1C	N/A
Reserved	29	1D	N/A
Reserved	30	1E	N/A

Individual command descriptions follow.

ROM/RAM CART STATUS -- Int. EEH, (AH) = 14H

Function: Returns operational data for the cartridge and reads the cartridge wait state specification into the system.

Description: The ROM/RAM CART STATUS command returns the following information about a cartridge:

- Type of cartridge (unformatted, executable, or file-structured)
- ROM/RAM size in 4K-byte units
- Start address of cartridge RAM
- Maximum number of available file slots
- Total number of blocks minus 1 in the cartridge
- Size of each block in the cartridge
- Control flags for the cartridge type and number of required wait states

The user must invoke this command whenever a new cartridge is inserted. If the cartridge is formatted, the ROM/RAM CART STATUS command selects the correct wait state hardware. If the cartridge is not formatted, call RAM CART FORMAT to specify the necessary format information.

Input Parameters:

(AL) is cartridge slot 0 or 1 for a status check.

Output Parameters:

(AL) is the cartridge type. (AL) is 0 to indicate an unformatted cartridge or 1 to indicate a formatted cartridge.

(SI) is ROM/RAM size in 4K-byte units. For a ROM/RAM cartridge, the lower byte is the ROM size and the upper byte is the RAM size. For a ROM cartridge or a RAM cartridge, the lower byte is the size of the first half and the upper byte is the size of the second half.

(CX) consists of the following control flags:

Bit(s)	Function
0,1	indicate the cart type as follows:

Bit Values	Cart Type
0,1	RAM cartridge
1,0	ROM cartridge
1,1	ROM/RAM cartridge

Bit(s)	Function
2	is the execute bit, where: 0 = non-executable cartridge. 1 = executable cartridge.
3	is a 1 for the second of a set of two cartridges.
4	is a 1 for the master of a set of two cartridges.
5	is a 1 if a wait state is required or a 0 if no wait state is required.
6,7	are reserved.
8	is a 1 for a file-structured cartridge, or a 0 for a non-file-structured cartridge.
9 - 15	are reserved.

The following registers are for file-formatted RAM cartridges:

(BH) is the maximum number of files for the cartridge.

(BL) is the total number of blocks minus 1 in the cartridge.

(DX) is the size of each block in bytes.

(ES) is the segment of the first half of the cartridge.

The following registers are for ROM or ROM/RAM cartridges, or RAM cartridges having no file format:

(BX) is the segment of the second half of the cartridge.

(ES) is the segment of the first half of the cartridge.

All other registers remain unchanged.

RAM CART FORMAT -- Int. EEH, (AH) = 15H

Function: Formats a RAM cartridge to accept information and support a directory of information blocks.

Description: The RAM CART FORMAT command sets up the directory and the file allocation table for a RAM cartridge.

The ROM CART FORMAT command writes the following information to the first four words in the RAM cartridge:

Cartridge Word Number	Word Contents
1	Cartridge ID number
2	Cartridge control flags
3	ROM/RAM size in 4K-byte units
4	Offset to first RAM address

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the maximum number of files permitted for this cartridge (provided CX bit 8 is 1).

(BL) is the number of blocks minus 1 for this cartridge (provided CX bit 8 is 1).

(CX) consists of the following control flags:

Bit(s)	Function
0,1	indicate the cart type bits as follows:

Bit Values	Cart Type
0,1	RAM cartridge
1,0	ROM cartridge
1,1	ROM/RAM cartridge

2	is the execute bit, where:

0 = non-executable cartridge.
1 = executable cartridge.

3	is a 1 to specify the second of a set of two cartridges.
4	is a 1 to specify the master of a set of two cartridges.

Bit(s)	Function
5	is a 1 if a wait state is required or a 0 if no wait state is required.
6,7	are reserved.
8	is a 1 to specify a file-structured cartridge, or a 0 to specify a non-file-structured cartridge.
9 - 15	are reserved.

(DX) is the size of each block in bytes (provided CX bit 8 is 1).

Output Parameters: None. All registers remain unchanged.

RAM CART GET ENTRY -- Int. EEH, (AH) = 16H

Function: Returns RAM cartridge file entry information from the cartridge directory.

Description: The RAM CART GET ENTRY command returns the 20 bytes of file entry information for a file in a RAM cartridge. See Table 8-1 for a list of the file entry information parameters. The user must supply a pointer to the area of memory that is to receive the 20 bytes of directory data.

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the file number.

(ES:[SI]) is a pointer to the area of memory that is to receive the 20 bytes of directory data.

Output Parameters:

(AX) is the success or failure status for this command as follows:

Value of AX	Status
0	No errors.
1	No such file. The system returned from a read or write call with a bad file number.
2	End of file. The system returned from a read call requesting data beyond the last block.
3	No more empty blocks. The system returned from a write call when the cartridge was already full.

All other registers remain unchanged.

RAM CART PUT ENTRY -- Int. EEH, (AH) = 17H

Function: Writes RAM cartridge file entry information to the cartridge directory.

Description: The RAM CART PUT ENTRY command writes the first 18 bytes of the 20-byte file entry information table. See Table 8-1 for a list of the file entry information parameters. The RAM CART WRITE and RAM CART DELETE BLOCK commands indirectly determine the contents of the last two bytes of the 20-byte file entry information table.

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the file number.

(ES:[SI]) is a pointer to the area of memory containing the 18 bytes of directory data.

Output Parameters:

(AX) is the success or failure status for this command as follows:

Value of AX	Status
0	No errors.
1	No such file. The system returned from a read or write call with a bad file number.
2	End of file. The system returned from a read call requesting data beyond the last block.
3	No more empty blocks. The system returned from a write call when the cartridge was already full.

All other registers remain unchanged.

RAM CART READ -- Int. EEH, (AH) = 18H

Function: Reads blocks of data from a RAM cartridge file.

Description: The RAM CART READ command copies the specified blocks of data from a cartridge file to the area of memory specified by the user.

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the file number.

(BL) is the first block to be read from the file (block 0 is the first block in a file).

(CL) is the number of blocks to read.

(ES:[SI]) is a pointer to the area of memory that is to receive the file information.

Output Parameters:

(AX) is the success or failure status for this command as follows:

Value of AX	Status
0	No errors.
1	No such file. The system returned from the read call with a bad file number.
2	End of file. The system returned from a read call requesting data beyond the last block.

(CX) is the number of blocks read.

All other registers remain unchanged.

RAM CART WRITE -- Int. EEH, (AH) = 19H

Function: Writes blocks of data to a RAM cartridge file.

Description: The RAM CART WRITE command copies information from memory into a RAM cartridge file.

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the file number.

(BL) is the first block to be written to the file in the cartridge memory.

(CL) is the number of blocks to write.

(ES:[BX]) contains a pointer to the area of memory containing the data to copy to the RAM cartridge.

Output Parameters:

(AX) is the success or failure status for this command as follows:

Value of AX	Status
0	No errors.
1	No such file. The system returned from the write call with a bad file number.
2	(unused).
3	No more empty blocks. The system returned from a write call when the cartridge was already full.

(CX) is the number of blocks written.

All other registers remain unchanged.

RAM CART DELETE BLOCK -- Int. EEH, (AH) = 1AH

Function: Deletes blocks of data from a RAM cartridge file.

Description: The RAM CART DELETE BLOCK command frees memory blocks within a RAM cartridge and removes the blocks from the linked list of blocks within the file.

Input Parameters:

(AL) is cartridge slot 0 or 1.

(BH) is the file number.

(BL) is the first block in the file to delete.

(CL) is the number of blocks to delete.

Output Parameters:

(AX) is the success or failure status for this command as follows:

Value of AX	Status
0	No errors.
1	No such file. The system returned with a bad file number.
2	End of file.

(CX) is the number of blocks deleted.

All other registers remain unchanged.

Section 9

MINDSET-NATIVE COMMUNICATIONS

OVERVIEW OF MINDSET-UNIQUE RS-232-C COMMUNICATION CAPABILITIES

The Mindset Personal Computer provides an expanded set of RS-232-C communication commands with capabilities beyond the industry-standard commands. These Mindset-unique commands support interrupt-driven RS-232-C character transfers that occur as a background process.

To use interrupt-driven character transfers, the user program must first call the industry-standard SET RS-232-C MODE command to set baud rates and data transmission parameters. Then the user program must specify buffer queues for transmitted and received data. The input buffer holds information received from an RS-232-C port until the user program reads it. The output buffer holds user program output information until the RS-232-C link transmits it.

The user can selectively enable various transmit and receive operations with the commands described in this section. These transmit and receive options must be specified before the user can invoke the other Mindset-unique RS-232-C commands to send and receive characters.

DESCRIPTION OF RS-232-C AND MODEM COMMANDS

A list of the Mindset RS-232-C and modem commands follows. Access to these commands is through interrupt EEH. Each command is described in more detail in the remainder of this section.

Command	Value of AL Decimal	AH Hex	Function
RS-232-C Send Character	40	28	Writes a character to the RS-232-C output buffer.
RS-232-C Get Character	41	29	Returns a character from the RS-232-C input buffer.
RS-232-C Send Character String	42	2A	Writes a character string to the RS-232-C output buffer.
RS-232-C Get Buffer Status	43	2B	Returns the available space in the RS-232-C input and output buffers.
RS-232-C Set Input Buffer	44	2C	Specifies the RS-232-C input buffer.
RS-232-C Set Output Buffer	45	2D	Specifies the RS-232-C output buffer.

Command	Value of AH Decimal	Hex	Function
RS-232-C Set Communications Control	46	2E	Specifies RS-232-C communications controls and interrupt enables.
RS-232-C Get Communications Control	47	2F	Returns RS-232-C communications controls and interrupt enables.
RS-232-C Get Modem Status	48	30	Returns information on modem installations.
RS-232-C Set Communications Break	49	31	Sends a break signal over the RS-232-C communications line.

Individual command descriptions follow.

RS-232-C SEND CHARACTER -- Int. EEH, (AH) = 28H

Function: Writes one character to the RS-232-C output buffer.

Description: The RS-232-C SEND CHARACTER command writes one character to the RS-232-C output buffer. If the buffer is full, the routine waits until space becomes available and then writes the character to the buffer.

Input Parameters:

(AL) is the character to be written into buffer.

(DX) is the RS-232-C module selected (0, 1, or 2).

Output Parameters: None. All registers remain unchanged.

RS-232-C GET CHARACTER -- Int. EEH, (AH) = 29H

Function: Returns a character from the RS-232-C input buffer.

Description: The RS-232-C GET CHARACTER command reads one character and the corresponding status byte from the RS-232-C input buffer. If the buffer is empty, the routine waits until a character is received.

Input Parameters:

(DX) is the RS-232-C module selected (0, 1, or 2).

Output Parameters:

(AL) is the character read from the buffer.

(AH) is the communications status for the character read in industry-standard format. (See the RS-232-C interrupt description in Section 3, "Industry-Compatible BIOS Functions", for format details.)

All other registers remain unchanged.

RS-232-C SEND CHARACTER STRING -- Int. EEH, (AH) = 2AH

Function: Writes a character string to the RS-232-C output buffer.

Description: The RS-232-C SEND CHARACTER STRING command writes a string of characters to the RS-232-C character output buffer. If the output buffer does not have sufficient space for all the characters, the routine waits for transmission over the line to empty the buffer. As each byte opens up in the output buffer, the routine writes another character to it until all the characters are written.

Input Parameters:

(CX)	is the number of characters in the string.
(DX)	is the RS-232-C module selected (0, 1, or 2).
(ES:[BX])	is the pointer to string.

Output Parameters: None. All registers remain unchanged.

RS-232-C GET BUFFER STATUS -- Int. EEH, (AH) = 2BH

Function: Returns the status of the RS-232-C input and output buffers.

Description: The RS-232-C GET BUFFER STATUS routine returns the number of empty bytes currently in the RS-232-C input and output buffers. It also returns buffer-empty and buffer-full status information.

Input Parameters:

(DX) is the RS-232-C module selected (0, 1, or 2).

Output Parameters:

(AH) is the buffer-empty/buffer-full status as follows:

Bit	Function
0	is 1 if the input buffer is empty or 0 otherwise.
1	is 1 if the output buffer is full or 0 otherwise.

(AL) is indeterminate.

(BX) is the number of empty words in the input buffer.

(CX) is the number of empty bytes in the output buffer.

All other registers remain unchanged.

RS-232-C SET INPUT BUFFER -- Int. EEH, (AH) = 2CH

Function: Specifies the size and location of the RS-232-C input buffer.

Description: The RS-232-C SET INPUT BUFFER command specifies the RS-232-C input buffer. The RS-232-C input buffer stores one byte of data and one byte of status bits for each character received. Thus, one word (two bytes) is required for each character that is stored in the input buffer. The input buffer stores incoming characters until they are read by a user program.

Input Parameters:

(CX)	is the buffer size (in words). (The buffer must lie in a contiguous 64K segment.)
(DX)	is the RS-232-C module selected (0, 1, or 2).
(ES:[BX])	is the buffer start address.

Output Parameters: None. All registers remain unchanged.

RS-232-C SET OUTPUT BUFFER -- Int. EEH, (AH) = 2DH

Function: Sets up the size and location of the RS-232-C output buffer.

Description: The RS-232-C SET OUTPUT BUFFER command specifies the RS-232-C output buffer. Unlike the input buffer, the output buffer is one byte wide. This buffer is used to store outgoing characters until they can be transmitted over an RS-232-C link.

Input Parameters:

(CX)	is the buffer size (in bytes). (The buffer must lie in a contiguous 64K segment.)
(DX)	is the RS-232-C module selected (0, 1, or 2).
(ES:[BX])	is the buffer start address.

Output Parameters: None. All registers remain unchanged.

RS-232-C SET COMMUNICATIONS CONTROL -- Int. EEH, (AH) = 2EH

Function: Sets up the communications controls and interrupt enables for RS-232-C communications.

Description: The RS-232-C SET COMMUNICATIONS CONTROL routine specifies the RS-232-C Request to Send (RTS) and Data Terminal Ready (DTR) control lines. It also specifies the transmit and receive interrupt enables.

Input Parameters:

(AL) specifies the RS-232-C control lines as follows:

Bit	Function
0	is the Data Terminal Ready (DTR) status (0 or 1).
1	is the Request to Send (RTS) status (0 or 1).

(BL) specifies the interrupt enables as follows:

Bit	Function
0	is a 1 to enable the receive interrupt or a 0 to disable the receive interrupt.
1	is a 1 to enable the transmit interrupt or a 0 to disable the transmit interrupt.

(DX) is the RS-232-C module selected (0, 1, or 2).

Output Parameters: None. All registers remain unchanged.

8080
8000
8010
8140
8180
81C0

RS-232-C GET COMMUNICATIONS CONTROL -- Int. EEH, (AH) = 2FH

Function: Returns the current specifications of the communications controls and interrupt enables for RS-232-C communications.

Description: The RS-232-C GET COMMUNICATIONS CONTROL routine returns the information set in the RS-232-C SET COMMUNICATIONS CONTROL command: the RS-232-C Request to Send (RTS) and Data Terminal Ready (DTR) control lines, and the transmit and receive enables.

Input Parameters:

(DX) is the RS-232-C module selected (0, 1, or 2).

Output Parameters:

(AL) indicates the status of the control lines as follows:

Bit	Function
0	is the Data Terminal Ready (DTR) status (0 or 1).
1	is the Request to Send (RTS) status (0 or 1).

(BL) indicates the interrupt enables as follows:

Bit	Function
0	is a 1 if the receive interrupt is enabled or a 0 if the receive interrupt is disabled.
1	is a 1 if the transmit interrupt is enabled or a 0 if the transmit interrupt is disabled.

(AH), (BH) are indeterminate.

All other registers remain unchanged.

RS-232-C GET MODEM STATUS -- Int. EEH, (AH) = 30H

Function: Returns information on which modem modules are installed.

Description: The RS-232-C GET MODEM STATUS command indicates whether a 300-baud modem module, 1200-baud modem module, or both modem modules are installed in the Mindset computer. This command also returns the RS-232-C index associated with each modem module. The index is the value placed in DX for all RS-232-C commands.

Input Parameters: None.

Output Parameters: (AL) is the modem module status as follows:

Bit(s)	Function
0,1	are unused.
2	is a 1 if a 300-baud modem module is installed or - 0 if no 300-baud modem module is installed.
3,4	indicate the 300-baud modem module index (valid only if bit 2 is a 1).
5	is a 1 if a 1200-baud modem module is installed or a 0 if no 1200-baud modem module is installed.
6,7	indicate the 1200-baud modem module index (valid only if bit 5 is 1).

(AH) is indeterminate.

All other registers remain unchanged.

RS-232-C SET COMMUNICATIONS BREAK -- Int. EEH, (AH) = 31H

Function: Sets or clears the communications break transmission enable.

Description: The RS-232-C SET COMMUNICATIONS BREAK command controls communications break transmissions. This command sets a communications break enable on a specified modem module to allow break transmissions to occur and clears the enable to prevent break transmissions.

Input Parameters:

	(AL)	is a 1 to set the break transmission enable or a 0 to disable break transmissions.
	(DX)	is the RS-232-C module selected (0, 1, or 2).

Output Parameters: None. All registers remain unchanged.

Section 10

MISCELLANEOUS BIOS COMMANDS

INTRODUCTION TO MISCELLANEOUS COMMANDS

The miscellaneous BIOS commands enable the user to write character strings to the screen display, control programs through a joystick, return the ID table of installed modules, turn off the system power, set the system LEDs, and copy the screen display to the printer.

DESCRIPTIONS OF MISCELLANEOUS COMMANDS

A list of the miscellaneous commands follows. Access to these commands is through interrupt EEH. Each command is described in further detail in the remainder of this section.

Command	Value of AH Decimal	Hex	Function
Write TTY String	00	00	Writes a character string to a screen buffer.
Write TTY String with Attributes	01	01	Writes a character string with character attributes to a screen buffer.
Set Cursor Shape	05	05	Sets the shape of the cursor for character modes.
Joystick/Mouse I/O	31	1F	Reads the status of two joystick/button, mouse, or other encoder-type controls.
Get Module ID Table	32	20	Returns a pointer to the table of IDs for the installed modules.
Set Power Off	33	21	Turns off the power to the system.
Set LEDs	34	22	Turns the system LEDs on or off.
Print String	35	23	Prints a string of characters at the printer.
Set Auxiliary Output Port	50	32	Sets the system auxiliary output port logic state.
Set Module Int	51	33	Specifies the interrupt pointer for the user-specified module interrupt service routine.

Command	Value of AH Decimal	Hex	Function
Set System Timer Rate	52	34	Changes the rate at which the system timer interrupts.
Get System Timer Rate	53	35	Returns the rate at which the system timer interrupts.
Enable/Disable Beeper	54	36	Enables or disables the system beeper.
Check Beeper Enable	55	37	Indicates whether the beeper is enabled.
Set Beeper	56	38	Turns the beeper on or off, provided it is enabled.

WRITE TTY STRING -- Int. EEH, (AH) = 00H

Function: Writes a character string to a screen buffer.

Description: The WRITE TTY STRING command writes a string of characters to the current cursor position within the page specified by (BH). The string can include the following TTY characters: bell, backspace, return, and line feed.

The parameters for the WRITE TTY STRING command include a count of the number of characters in the string, a foreground color for graphics modes, a destination page for character modes and double-buffered graphics modes, and a pointer to the first character of the string.

In graphics modes, setting the high bit of (BL) to 1 causes the character string to be exclusive-ORed into the background.

Input Parameters:

(BL)	is the foreground color for graphics modes. If the most significant bit is set, the system exclusive-ORs the character string into the background.
(BH)	is the destination page for character modes and double-buffered graphics modes.
(CX)	is the number of characters in the string.
(ES:[SI])	is the address of the first character in the string.

Output Parameters: None. All registers remain unchanged.

WRITE TTY STRING WITH ATTRIBUTES -- Int. EEH, (AH) = 01H

Function: Writes a character string with character attributes to a screen buffer.

Description: The WRITE TTY STRING WITH ATTRIBUTES command writes a string of characters with one character attribute byte specifying the attributes for every character in the string. This command writes the string to the current cursor position within the page specified by (BH). The string can include the following TTY characters: bell, backspace, return, and line feed.

For graphics modes, the operation of the WRITE TTY STRING WITH ATTRIBUTES command is the same as WRITE TTY STRING. BL, the register that specifies the attributes for strings in character modes, specifies the foreground color for objects in graphics modes. Setting the high bit of (BL) to 1 causes the string to be exclusive-ORed into the background.

The parameters for the WRITE TTY STRING command include a count of the number of characters in the string, an attribute for the string for character modes (or the foreground color for graphics modes), a destination page for character modes and double-buffered graphics modes, and a pointer to the first character of the string.

Input Parameters:

(BL)	specifies the character attribute for the entire string for character modes or sets the foreground color for graphics modes.
(BH)	is the destination page for character modes and double-buffered graphics modes.
(CX)	is the number of characters in the string.
(ES:[SI])	is the address of the first character in the string.

Output Parameters: None. All registers remain unchanged.

SET CURSOR SHAPE -- Int. EEH, (AH) = 05H

Function: Sets the cursor shape for character modes.

Description: The SET CURSOR SHAPE command enables the user to specify the shape of the cursor. The system superimposes the cursor on top of a character with an exclusive-OR operation. The system blinks the cursor on and off approximately four times per second.

The cursor is 8 pixels high and 8 pixels wide in both 40- and 80-column character modes (see Figure 10-1). Each pixel is represented by a single bit.

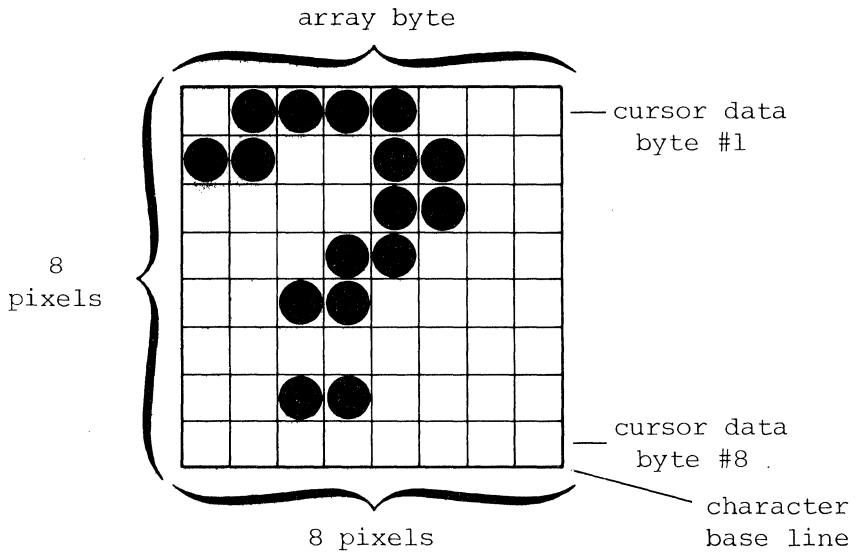


Figure 10-1: Cursor shape definition

The parameter for SET CURSOR SHAPE is a pointer to the 8 bytes of cursor shape data.

The cursor shape definition is left-justified and top-justified so that the first byte in the array defines the upper left pixel in the cursor. This pixel corresponds to the upper left pixel of the character cell that the cursor occupies.

After this command copies the cursor shape data from the user array to the system's reserved memory, the user may destroy the array without affecting the customized cursor.

Input Parameters: (ES:[SI]) is the address of the 8-byte cursor shape definition array in memory.

Output Parameters: None. All registers remain unchanged.

JOYSTICK/MOUSE I/O -- Int. EEH, (AH) = 1FH

Function: Returns the status of two joystick/button, mouse, or other encoder-type controls.

Description: The JOYSTICK/MOUSE I/O command enables the user to create programs that use up to two joysticks, mouse units, or other encoder-type devices as input. In addition to the usual five joystick inputs (up, down, right, left, and button), the JOYSTICK/MOUSE I/O command also reads X-axis and Y-axis values from joysticks, mouse units, or other devices with encoder outputs. These X-axis and Y-axis values reflect the change in position since the last time the JOYSTICK/MOUSE I/O routine was called.

Input Parameters: None.

Output Parameters:

(AL) represents the switch positions of JOYSTICK-A as follows (a 0 indicates a closed switch and a 1 indicates an open switch):

Bit	Switch Position
0	Up switch
1	Down switch
2	Left switch
3	Right switch
4	Button switch 1
5	Button switch 2

(AH) represents the switch positions of JOYSTICK-B (same as for JOYSTICK-A).

(BX) is the X-axis position change of JOYSTICK/MOUSE-A.

(CX) is the Y-axis position change of JOYSTICK/MOUSE-A.

(DX) is the X-axis position change of JOYSTICK/MOUSE-B.

(SI) is the Y-axis position change of JOYSTICK/MOUSE-B.

All other registers remain unchanged.

GET MODULE ID TABLE -- Int. EEH, (AH) = 20H

Function: Returns a pointer to the table of module IDs for modules installed in the system.

Description: The GET MODULE ID TABLE command returns a pointer to a byte table of the module IDs read at each module location in the system.

If so designed, a module can power up the system using the wake-up interrupt. To accomplish this action, the wake-up feature must be enabled and the module must place a "low" voltage on pin 7 of its connection to the system.

A value of 0 for bit 7 of the module ID byte indicates that the wake-up interrupt is active for that module. This wake-up bit is valid only if bits 0 through 6 indicate a valid module ID from the following table:

Module ID	Module
0FFH	None
070H	Printer
073H	RS-232-C
013H	Stereo sound
03FH	1200-baud modem
05FH	300-baud modem

Input Parameters: None.

Output Parameters:

(AX) is the I/O address of the custom sound-processor (CSP).

(CX) is the number of entries in the module ID table.

(DX) returns the I/O address of module zero. You can calculate the address of any module N from the formula:

Address of module N =
Address of module zero + (N times 40H)

(ES:[BX]) is the pointer to the module ID table.

All other registers remain unchanged.

SET POWER OFF -- Int. EEH, (AH) = 21H

Function: Turns off the power to the system.

Description: The SET POWER OFF command enables the user to turn off the system power from within a program. This command does not turn off the trickle charge that provides power for the real-time clock.

Input Parameters: None.

Output Parameters: None.

All registers are made inactive by the system power-down.

SET LEDS -- Int. EEH, (AH) = 22H

Function: Turns the system LEDs on or off according to a user-specified pattern.

Description: The SET LEDS command enables the user to selectively turn the system LEDs on or off.

Input Parameters: (AL) is the bit definition of the LED display as follows:

Bit	LED Definitions
0	Green LED: 1 = on 0 = off
1	Yellow LED: 1 = on 0 = off

Output Parameters: None. All registers remain unchanged.

PRINT STRING -- Int. EEH, (AH) = 23H

Function: Prints a string of characters at the printer.

Description: The PRINT STRING command enables the user to specify a string of characters in memory for output to the printer. This command returns the printer status to the user.

Input Parameters:

(CX)	is the number of characters to print.
(DX)	is the printer module (0, 1, or 2).
(ES:[SI])	is a pointer to the string in memory.

Output Parameters: (AH) is the printer status as follows:

Bit	Printer Status
0	Printer timeout
1	(unused)
2	(unused)
3	Printer I/O error
4	Printer selected
5	Out of paper
6	Printer acknowledge
7	Printer ready

All other registers remain unchanged.

SET AUXILIARY OUTPUT PORT -- Int. EEH, (AH) = 32H

Function: Sets the logic state of the auxiliary output port.

Description: The SET AUXILIARY OUTPUT PORT command sets the logic state of the auxiliary output port to 1 (on) or 0 (off). The auxiliary output port allows the Mindset computer to control an external device.

Input Parameters:

(AL)	is 1 to set (turn on) the auxiliary output port or 0 to reset (turn off) the auxiliary output port.

Output Parameters: None. All registers remain unchanged.

SET MODULE INT -- Int. EEH, (AH) = 33H

Function: Sets the address of a user-specified module interrupt service routine.

Description: The SET MODULE INT command specifies an interrupt pointer to be used by any of the installed I/O modules. The interrupt pointer directs the system to a user-specified routine when a module interrupt occurs. The user-supplied interrupt routine then invokes instructions appropriate to the module interrupt received.

The system interrupt handler invokes all user-specified interrupt routines with a FAR CALL instruction. Therefore, the user should terminate these routines with a RETURN instruction and specify the routine as a FAR procedure to the assembler.

Input Parameters: (ES:[BX]) is the address of the user-specified interrupt service routine. (0:0) specifies that no routine is to be called.

Output Parameters: None. All registers remain unchanged.

SET SYSTEM TIMER RATE -- Int. EEH, (AH) = 34H

Function: Determines the frequency of system timer interrupts.

Description: The SET SYSTEM TIMER RATE command changes the rate of interrupts generated by the system timer. In addition to calling a user-specified timing routine via interrupt 1CH, the system timer provides signals for industry-compatible Time-of-Day counting.

The default interrupt rate is 2.5 milliseconds. Mindset Time-of-Day operation is not affected by changes in the system timer rate.

Input Parameters: (CX) is the number of 2.5 millisecond ticks per timer interrupt.

Output Parameters: None. All registers remain unchanged.

GET SYSTEM TIMER RATE -- Int. EEH, (AH) = 35H

Function: Returns the frequency of system timer interrupts.

Description: The GET SYSTEM TIMER RATE command returns the number of 2.5 millisecond ticks per timer interrupt.

Input Parameters: None.

Output Parameters:

(CX)	is the number of 2.5 millisecond ticks per timer interrupt.

All other registers remain unchanged.

ENABLE/DISABLE BEEPER -- Int. EEH, (AH) = 36H

Function: Enables or disables the system beeper.

Description: The ENABLE/DISABLE BEEPER command controls beeper operation. If enabled, the system beeper beeps in response to a CTRL-G keyboard entry, the bell character generated with a WRITE TELETYPE command, a keyboard buffer full error, or a SET BEEPER ON command. The system beeper is useful if the system has no other sound output, but it is not necessary if a television speaker or other external speaker is connected.

Input Parameters: (AL) is a 1 to enable the beeper or a 0 to disable the beeper.

Output Parameters: None. All registers remain unchanged.

CHECK BEEPER ENABLE -- Int. EEH, (AH) = 37H

Function: Returns beeper enabled/disabled status.

Description: The CHECK BEEPER ENABLE command returns beeper enabled/disabled status. If enabled, the beeper sounds in response to CTRL-G, the bell character generated with a WRITE TELETYPE command, a keyboard buffer full error, or a SET BEEPER ON command.

Input Parameters: None.

Output Parameters:

(AL) is a 1 if the beeper is enabled or a 0 if the beeper is disabled.

All other registers remain unchanged.

SET BEEPER -- Int. EEH, (AH) = 38H

Function: Turns the beeper on or off, if the beeper is enabled.

Description: The SET BEEPER command turns the system beeper on or off, providing the beeper is enabled. The system beeper sounds from the time SET BEEPER turns it on until the time SET BEEPER turns it off. The SET BEEPER call has no effect if the beeper is disabled.

Input Parameters: (AL) is 1 to turn the beeper on or 0 to turn the beeper off.

Output Parameters: None. All registers remain unchanged.

Appendix A

POWER-UP SEQUENCE ERROR MESSAGES

An error message indicates that your system has encountered a problem while performing a test or function. If one of the following error messages appears on your video screen your system may still be operable. Press any key to continue. If your system does not operate, contact your dealer for assistance.

System RAM Test Failed

System ROM Test Failed

Bad Cartridge in Slot "x"

Display RAM Test Failed

Expansion RAM Test Failed

System Timer Test Failed

DMA Controller Test Failed

Printer Module Test Failed

RS-232-C Module Test Failed

Real Time Clock Test Failed

Modem Module Test Failed

Graphics Coprocessor Test Failed

Display Processor Test Failed

Appendix B

TABLE OF ASCII CODES

The ASCII codes (listed below) are the decimal values of 7-bit codes representing alphanumeric characters or non-printing control characters. The table below consists of a control characters part and an alphanumeric characters part.

ASCII Code (Decimal Character Value)	Control Character	Description	Control Sequence
0	NUL	Null	CTRL @
1	SOH	Start of Heading	CTRL A
2	STX	Start of Text	CTRL B
3	ETX	End of Text	CTRL C
4	EOT	End of Transmission	CTRL D
5	ENQ	Enquiry	CTRL E
6	ACK	Acknowledge	CTRL F
7	BEL	Bell	CTRL G
8	BS	Backspace	CTRL H
9	HT	Horizontal Tabulation	CTRL I
10	LF	Line Feed	CTRL J
11	VT	Vertical Tabulation	CTRL K
12	FF	Form Feed	CTRL L
13	CR	Carriage Return	CTRL M
14	SO	Shift Out	CTRL N
15	SI	Shift In	CTRL O
16	DLE	Data Link Escape	CTRL P
17	DC1	Device Control 1	CTRL Q
18	DC2	Device Control 2	CTRL R
19	DC3	Device Control 3	CTRL S
20	DC4	Device Control 4	CTRL T
21	NAK	Negative Acknowledge	CTRL U
22	SYN	Synchronous Idle	CTRL V
23	ETB	End of Transmission Block	CTRL W
24	CAN	Cancel	CTRL X
25	EM	End of Medium	CTRL Y
26	SUB	Substitute	CTRL Z
27	ESC	Escape	CTRL [
28	FS	File Separator	CTRL \
29	GS	Group Separator	CTRL]
30	RS	Record Separator	CTRL ^
31	US	Unit Separator	CTRL _
127	DEL	Delete	DEL

ASCII Code (Decimal Character Value)	Character	ASCII Code (Decimal Character Value)	Character
32	Space	79	O
33	!	80	P
34	"	81	Q
35	#	82	R
36	$	83	S
37	%	84	T
38	&	85	U
39	'	86	V
40	(	87	W
41	)	88	X
42	*	89	Y
43	+	90	Z
44	'	91	[
45	-	92	\
46	.	93	]
47	/	94	^
48	0	95	<
49	1	96	.
50	2	97	a
51	3	98	b
52	4	99	c
53	5	100	d
54	6	101	e
55	7	102	f
56	8	103	g
57	9	104	h
58	:	105	i
59	;	106	j
60	<	107	k
61	=	108	l
62	>	109	m
63	?	110	n
64	@	111	o
65	A	112	p
66	B	113	q
67	C	114	r
68	D	115	s
69	E	116	t
70	F	117	u
71	G	118	v
72	H	119	w
73	I	120	x
74	J	121	y
75	K	122	z
76	L	123	{
77	M	124	|
78	N	125	}
	126	~	

Appendix C

TABLE OF KEYBOARD SCAN CODES

The keyboard of the Mindset computer consists of an 8 x 11 key matrix and two independent keys (CAPS LOCK and SCROLL LOCK). Keyboard software in the system scans this matrix every 13.2 milliseconds. This software sends "make" and "break" codes to the System Unit for any keys depressed and released during the preceding 13.2 milliseconds. The scan codes generated by this software are given below.

Key Designation	Scan Code
ESC	01
1	02
2	03
3	04
4	05
5	06
6	07
7	08
8	09
9	10
0	11
-	12
=	13
BACKSPACE	14
TAB	15
Q	16
W	17
E	18
R	19
T	20
Y	21
U	22
I	23
O	24
P	25
[	26
]	27
RETURN	28
CTRL	29
A	30
S	31
D	32
F	33
G	34
H	35
J	36
K	37
L	38
;	39

Key Designation	Scan Code
'	40
`	41
LEFT SHIFT	42
\	43
Z	44
X	45
C	46
V	47
B	48
N	49
M	50
,	51
.	52
/	53
RIGHT SHIFT	54
*	55
ALT	56
SPACE	57
CAPS LOCK	58
F1	59
F2	60
F3	61
F4	62
F5	63
F6	64
F7	65
F8	66
F9	67
F10	68
SCR LK	70
HOME	71
CURSOR UP	72
PAGE UP	73
CURSOR LEFT	75
CURSOR RIGHT	77
END	79
CURSOR DOWN	80
PAGE DOWN	81
INSERT	82
DEL	83
START	84
PAUSE	85
SYS CONFIG	86
RESET	87
BREAK	88

Appendix D

SYSTEM MEMORY MAP

Decimal Address	Hexadecimal Address	Mindset Memory
0	00000	32K on-board RAM
16K	04000	
32K	08000	
48K	0C000	Up to 224K bytes expansion RAM
64K	10000	
80K	14000	
96K	18000	
112K	1C000	
128K	20000	
144K	24000	
160K	28000	
176K	2C000	
192K	30000	
208K	34000	
224K	38000	
240K	3C000	
256K	40000	
272K	44000	Reserved
288K	48000	
304K	4C000	
320K	50000	
336K	54000	
352K	58000	
368K	5C000	
384K	60000	
400K	64000	
416K	68000	
432K	6C000	
448K	70000	
464K	74000	
480K	78000	
496K	7C000	
512K	80000	128K bytes for cartridge slot #1
528K	84000	
544K	88000	
560K	8C000	
576K	90000	
592K	94000	
608K	98000	
624K	9C000	
640K	A0000	Reserved
656K	A4000	
672K	A8000	

Decimal Address	Hexadecimal Address	Mindset Memory
688K	AC000	
704K	B0000	Reserved
720K	B4000	
736K	B8000	32,000 bytes for color
752K	BC000	graphics frame buffer
768K	C0000	
784K	C4000	
800K	C8000	128K bytes
816K	CC000	for
832K	D0000	cartridge
848K	D4000	slot #2
864K	D8000	
880K	DC000	
896K	E0000	
912K	E4000	
928K	E8000	
944K	EC000	Reserved
960K	F0000	
976K	F4000	
992K	F8000	32K bytes for the ROM-based
1008K	FC000	operating system

768 bytes reserved

Appendix E

SYSTEM STACK USAGE

The following tables describe the stack usage of the Mindset computer.

The software interrupt stack usage values include the five words for the software call. None of the values allow for the occurrence of a hardware interrupt during the routine.

When the user selects collision detection, the system requires 18 words of stack to handle the interrupt and call the user's procedure. The system ignores the stack usage of the user's procedure when performing collision or clipping detection.

SOFTWARE INTERRUPTS

Interrupt Number (Hex)	Command Name	Value of AH Decimal	AH Hex	Stack Usage (Words)
05	Print Screen	N/A	N/A	32
08	Timer Interrupt	N/A	N/A	6
0A	Refresh Interrupt	N/A	N/A	5
0C	Disk Interrupt	N/A	N/A	6
0F	Keyboard Interrupt	N/A	N/A	24
10	Set Mode	00	00	9
10	Set Cursor Type	01	01	12
10	Set Cursor Position	02	02	9
10	Read Cursor Position	03	03	7
10	Reserved	04	04	N/A
10	Set Active Page	05	05	10
10	Scroll Up	06	06	26
10	Scroll Down	07	07	24
10	Read Character	08	08	12 in character modes 22 in graphics modes
10	Write Character with Attributes	09	09	14 in character modes 22 in graphics modes
10	Write Character Only	10	0A	13 in character modes 22 in graphics modes
10	Set Colors	11	0B	12
10	Write Pixel	12	0C	18
10	Read Pixel	13	0D	15

Interrupt Number (Hex)	Command Name	Value of Decimal	AH Hex	Stack Usage (Words)
10	Write TTY Character	14	OE	16 for character in character modes 25 for character in graphics modes 32 if character causes scroll up 32 for bell
10	Get Mode	15	OF	6
11	Equipment Check	N/A	N/A	4
12	Memory Size	N/A	N/A	4
13	Disk I/O	N/A	N/A	16
14	RS-232-C I/O	N/A	N/A	9
15	Cassette I/O	N/A	N/A	11
16	Keyboard I/O	N/A	N/A	5
17	Printer I/O	N/A	N/A	9
19	Bootstrap	N/A	N/A	3
EE	RT Clock Enable Wake-up/Alarm	08	08	13
EE	RT Clock Get Wake-up/Alarm Status	09	09	11
EE	RT Clock Set Status	10	OA	13
EE	RT Clock Get Status	11	OB	11
EE	RT Clock Read Time	12	OC	15
EE	RT Clock Set Time	13	OD	14
EE	RT Clock Read Date	14	OE	15
EE	RT Clock Set Date	15	OF	14
EE	RT Clock Read Alarm	16	10	15
EE	RT Clock Set Alarm	17	11	14
EE	RT Clock Set Int	18	12	7
EE	Write TTY String	00	00	40 maximum
EE	Write TTY String with Attributes	01	01	40 maximum
EE	Set Display Device	02	02	6
EE	Set Screen Position	03	03	11
EE	Get Screen Position	04	04	8
EE	Set Cursor Shape	05	05	11
EE	Set Display Sync Features	06	06	8
EE	Set Display Interrupt	07	07	10
EE	ROM/RAM Cart Status	20	14	15
EE	RAM Cart Format	21	15	16
EE	RAM Cart Get Entry	22	16	12
EE	RAM Cart Put Entry	23	17	13
EE	RAM Cart Read	24	18	15
EE	RAM Cart Write	25	19	18

Interrupt Number (Hex)	Command Name	Value of Decimal	AH Hex	Stack Usage (Words)
EE	RAM Cart Delete Block	26	1A	15
EE	Joystick I/O	31	1F	
EE	Get Module ID Table	32	20	
EE	Set Power Off	33	21	
EE	Set LEDS	34	22	
EE	Print String	35	23	
EE	Set Sound Mode	36	24	12
EE	Set Sound Registers	37	25	9
EE	Sound Data	38	26	7
EE	Stereo Module Check	39	27	21
EF	Set Screen Mode	00	00	15
EF	Get Screen Mode	01	01	6
EF	Set Transfer Mode	02	02	7
EF	Get Transfer Mode	03	03	6
EF	Set Destination Buffer	04	04	12
EF	Get Destination Buffer	05	05	6
EF	Set Write Mask	06	06	6
EF	Get Write Mask	07	07	6
EF	BLT Copy	08	08	24
EF	BLT Copy Word	09	09	23
EF	Set Palette	10	0A	9
EF	Get Palette	11	0B	9
EF	BLT Polypoint	12	0C	21
EF	BLT Polyline	13	0D	36
EF	Get Buffer Info	14	0E	5
EF	Set Display Int Address	15	0F	6
EF	Get Display Int Address	16	10	6
EF	Switch Active Buffer	17	11	7
EF	Set Collision Pattern	18	12	8
EF	Get Collision Pattern	19	13	6
EF	Set Clip Rectangle	20	14	8
EF	Get Clip Rectangle	21	15	6
EF	Set Collision/Clip/Done Detect	22	16	7
EF	Get Collision/Clip/Done Detect	23	17	6
EF	GCP Wait	24	18	6
EF	BLT Polygon	25	19	47 to 447 (*)
EF	BLT Filled Ellipses	26	1A	48
EF	BLT Hollow Ellipses	27	1B	48
EF	Save GCP	28	1C	9
EF	Restore GCP	29	1D	14
EF	Fill Dest Buffer	30	1E	7
EF	Set Font Pointer	31	1F	6
EF	Get Font Pointer	32	20	6
EF	BLT String	33	21	20
EF	Set Param Block Mode	34	22	7
EF	Get Param Block Mode	35	23	6
EF	Get GCP Status	36	24	6

Interrupt Number (Hex)	Command Name	Value of Decimal	AH Hex	Stack Usage (Words)
EF	Get Char Bitmap Address	37	25	5
EF	Get BLT Memory Bounds	38	26	5
EE	RS-232-C Send Character	40	28	
EE	RS-232-C Get Character	41	29	
EE	RS-232-C Send Character String	42	2A	
EE	RS-232-C Get Buffer Status	43	2B	
EE	RS-232-C Set Input Buffer	44	2C	
EE	RS-232-C Set Output Buffer	45	2D	
EE	RS-232-C Set Communications Control	46	2E	
EE	RS-232-C Get Communications Control	47	2F	
EE	RS-232-C Get Modem Status	48	30	
EE	RS-232-C Set Communications Break	49	31	
EE	Set Auxiliary Output Port	50	32	
EE	Set Module Int	51	33	
EE	Set System Timer Rate	52	34	
EE	Get System Timer Rate	53	35	
EE	Enable/Disable Beeper	54	36	
EE	Check Beeper Enable	55	37	
EE	Set Beeper	56	38	

(*) BLT POLYGON requires (Y-BOTTOM - Y-TOP + 1) words of stack in addition to its fixed requirement of 47 words. (Y-BOTTOM AND Y-TOP are the vertical clipping bounds and can vary from 0 to 400.)

HARDWARE INTERRUPTS

Interrupt	Stack Usage (Words)
Display	9 in character modes 9+ user-defined display interrupt service routine in graphics modes
Late VBLANK (vertical flyback)	13

Appendix F

SOUND FREQUENCY TABLE

The following table shows frequencies (in Hertz--cycles per second) for selected musical notes over 8 octaves.

Note	Octave 0	Octave 1	Octave 2	Octave 3	Octave 4	Octave 5	Octave 6	Octave 7
C	16.35	32.70	65.40	130.80	261.60	523.20	1046.40	2092.80
C#	17.32	34.64	69.28	138.56	277.12	554.24	1108.48	2216.96
D	18.35	36.70	73.40	146.80	293.60	587.20	1174.40	2348.80
D#	19.45	38.90	77.80	155.60	311.20	622.40	1244.80	2489.60
E	20.60	41.20	82.40	164.80	329.60	659.20	1318.40	2636.80
F	21.83	43.66	87.32	174.64	349.28	698.56	1397.12	2794.24
F#	23.12	46.24	92.48	184.96	369.92	739.84	1479.68	2959.36
G	24.50	49.00	98.00	196.00	392.00	784.00	1568.00	3136.00
G#	25.96	51.92	103.84	207.68	415.36	830.72	1661.44	3322.88
A	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00
A#	29.14	58.28	116.56	233.12	466.24	932.48	1864.96	3729.92
B	30.87	61.74	123.48	246.96	493.92	987.84	1975.68	3951.36

Appendix G

SOUND GENERATOR SINE-VALUE TABLE

The custom sound-processor (CSP) uses the sine-value table below to generate sine-wave-based tones. The CSP samples the table at a regular rate and writes AMT values to the D/A converter. The frequency register of the CSP determines which AMT values the CSP writes to the D/A converter. For example, if the frequency register value is 1, the CSP samples AMT values in TIME increments of 1 (1,2,3,...). If the frequency register value is 10, the CSP samples the AMT values in TIME increments of 10 (1,11,21,...).

TIME	AMT	TIME	AMT	TIME	AMT	TIME	AMT	TIME	AMT	TIME	AMT	TIME	AMT
1	128	37	227	73	253	109	186	145	78	181	5	217	22
.	131	.	229	.	252	.	183	.	75	.	4	.	24
.	134	.	231	.	251	.	180	.	72	.	3	.	26
4	137	40	233	76	250	112	177	148	69	184	2	220	28
.	141	.	234	.	250	.	174	.	67	.	1	.	30
.	144	.	236	.	249	.	171	.	64	.	1	.	32
.	147	.	238	.	247	.	168	.	61	.	0	.	34
8	150	44	239	80	246	116	165	152	59	188	0	224	36
.	153	.	241	.	245	.	162	.	56	.	0	.	39
.	156	.	242	.	244	.	159	.	53	.	0	.	41
.	159	.	244	.	242	.	156	.	51	.	0	.	43
12	162	48	245	84	241	120	153	156	48	192	0	228	46
.	165	.	246	.	239	.	150	.	46	.	0	.	48
.	168	.	247	.	238	.	147	.	43	.	0	.	51
.	171	.	249	.	236	.	144	.	41	.	0	.	53
16	174	52	250	88	234	124	141	160	39	196	0	232	56
.	177	.	250	.	233	.	137	.	36	.	0	.	59
.	180	.	251	.	231	.	134	.	34	.	0	.	61
.	183	.	252	.	229	.	131	.	32	.	1	.	64
20	186	56	253	92	227	128	128	164	30	200	1	236	67
.	188	.	254	.	225	.	127	.	28	.	2	.	69
.	191	.	254	.	223	.	124	.	26	.	3	.	72
.	194	.	255	.	221	.	121	.	24	.	4	.	75
24	196	60	255	96	219	132	118	168	22	204	5	240	78
.	199	.	255	.	216	.	114	.	21	.	5	.	81
.	202	.	255	.	214	.	111	.	19	.	6	.	84
.	204	.	255	.	212	.	108	.	17	.	8	.	87
28	207	64	255	100	209	136	105	172	16	208	9	244	90
.	209	.	255	.	207	.	102	.	14	.	10	.	93
.	212	.	255	.	204	.	99	.	13	.	11	.	96
.	214	.	255	.	202	.	96	.	11	.	13	.	99
32	216	68	255	104	199	140	93	176	10	212	14	248	102
.	219	.	255	.	196	.	90	.	9	.	16	.	105
.	221	.	255	.	194	.	87	.	8	.	17	.	108
.	223	.	254	.	191	.	84	.	6	.	19	.	111
36	225	72	254	108	188	144	81	180	5	216	21	252	114
												.	118
												.	121
												.	124
												256	127

INDEX

248 pages