Hercules

1

Hardware Guide

InColor Card
Model GB222

Owner's Manual

Edition 1.0, April 1987

Printed in USA

How to Use this Manual

This is Part 1 of your Hercules InColor Card Owner's Manual. It contains information on installing and using your card and decribes technical features.

For information about the "Programs and Fonts" diskette and the "Drivers Diskette," included with your InColor Card, please refer to the Owner's Manual Part 2 Software in your InColor Card package.

This manual is divided into three sections.

Section 1 gets you started. It contains information on installing the Hercules InColor Card with RamFont in your system and on getting it working. It also discusses features of the Hercules InColor Card that can provide you with better performance from your system and software.

Section 2 offers in-depth coverage of the card's technical features. Designed primarily for the system developer and other technical users, it covers software and hardware considerations involved in developing applications for the InColor Card. If you are a non-technical user, you may still find useful information that can help you understand and explore the more advanced features of the InColor Card.

Section 3 contains the appendices. Topics covered include troubleshooting techniques, working with IBM-PC compatibles, and in-depth discussions of some of the technical features covered in Section 2.

Table of Contents

Section One—Installing and Using the Card

Section Two—Technical Reference

Section Three—Appendices

Section One

Installing and Using the Hercules InColor Card

Introduction
Setting Up
User Features

1

Chapter 1—Introduction

Product Features

The Hercules InColor Card with RamFont is a powerful new way to display color graphics.

The InColor Card supports applications in text mode, Hercules RamFont mode, and high-resolution, Hercules-standard graphics—all in up to sixteen colors.

- The 4k RamFont display lets you select your own on-screen typeface for standard text mode software—you can even design your own display characters if you choose.
- 48k RamFont works with applications designed to the new Hercules RamFont standard, to give you 12 typefaces in up to 16 colors, and as many as 48 typefaces in 4 colors.
- High resolution, 16-color graphics are possible, selected from a range of 64 available colors.
- With all these new features, the Hercules InColor Card with RamFont is still compatible with software written for both the Hercules Graphics Card and Graphics Card Plus.
- A parallel port for your printer is also included on the Hercules InColor Card.

Introduction

In the Package

In this package, you should find:

- Hercules InColor Card
- Owner's Manual Part 1—Hardware
- Owner's Manual Part 2—Software
- A Purchaser Registration Card
- Two diskettes. The "Programs and Fonts" disk contains useful utilities for controlling the operation of the InColor Card and a set of predefined fonts on disk for you to select from when you first start using the InColor Card. The "Software Drivers" disk contains links for using the InColor Card with several applications programs. These drivers allow the software to take full advantage of the InColor Card's unique features.
- The Hercules Software and Hardware Compatibility Guide.
- A conductive foam strip for storing the printer port chip.

Please take some time now to fill out and mail the Purchaser Registration Card. You will receive valuable information on updates and future products as they become available.

Introduction

What You Need

Before proceeding with installation, be sure you have the proper equipment and software to use the InColor Card. These include:

Computer

The InColor Card is designed for use with the IBM PC, XT or AT. It can work with **hardware** compatible PCs with a video adapter on a separate, user-removable card, or which can otherwise be disabled. If you're not sure whether your computer fits this category, see Appendix B for the process to determine whether or not it is.

Monitor

The InColor Card works with any enhanced RGB monitor operating at 22.0 kHz horizontal scan rate and 60 Hz vertical scan rate.

Printer

To use the InColor Card's printer port, you'll need an IBM-compatible parallel printer. You must also make certain that the software you plan to use is compatible with your printer. If your computer already has a parallel port installed as LPT1 you may have to remove the parallel port chip on the InColor card, or change the switches that control the parallel port in the computer. See Appendix B, "Compatibility with Other System Components," for information on compatibility and changes.

Software

The InColor Card works with any IBM text software. It runs existing software that supports the Hercules Graphics Card, providing a two-color display. And it works with software that conforms to the new Hercules RamFont standard, providing extra flexibility in text and graphics operations.

Introduction

Assumptions in this Manual

Some examples in this book assume that you are using either a system equipped with two floppy disks or one with a single floppy and a hard disk drive. In these cases, details are included to help you perform the described operations using both these configurations. If your system differs from those described, or you have questions regarding specific computer features or DOS commands mentioned, consult the DOS manual or your computer users' manual.

We're Here to Help

Telephone support is available if you should run into problems you can't solve on your own. However, before calling, please be sure to look through Appendix A, "Troubleshooting" for suggestions on how to solve problems that may commonly be encountered.

When you do call, please be prepared to describe the problem you are encountering, and any symptoms that go along with it. It is helpful for you to be at your system when you call; if possible, we can then step you through the procedures for correcting the problem.

Before Starting

It's important to make sure your computer and monitor are compatible with the InColor Card before you install it. Appendix B discusses several compatibility issues that may arise when you install the InColor Card in your computer. These include:

- Another parallel port.
- Another display adapter.

If your computer has one or both of these features, turn to Appendix B for instructions on how to deal with them.

Chapter 2—Setting Up

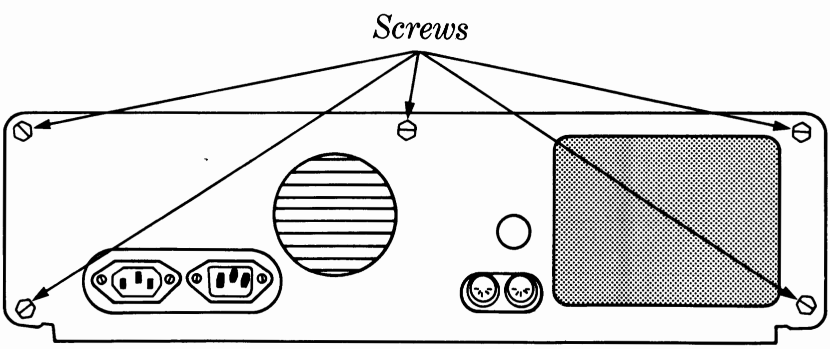
Installing the InColor Card

Installing the Hercules InColor Card is a very simple operation. In most cases, all that's required is to remove the top of your computer, insert the card, replace the top, plug in your monitor and switch the machine on.

Before proceeding with installation, be sure you have reviewed the last section, "Before Starting."

Note: These instructions assume you are installing your Hercules InColor Card in an IBM PC, XT or AT. If you have a different computer, you may need to consult your owner's manual for details at some of the steps. These instructions cover computers current through January, 1987.

1. Remove the five cover screws from the back of your computer's case, as shown:



Location of cover screws on the back of an IBM PC/XT/AT.

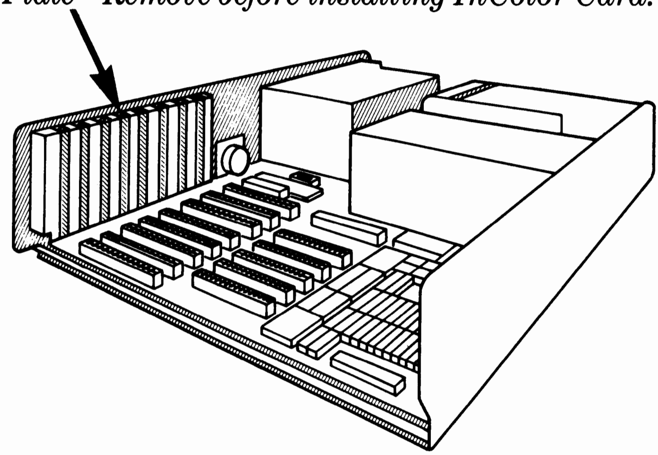
Setting Up

2. Slide the cover forward as far as it will go, then lift and remove it from the unit.

3. If you have a monochrome video card in your system, be sure to remove it. If your system contains a color graphics adapter, see Appendix B for instructions on what to do with it.

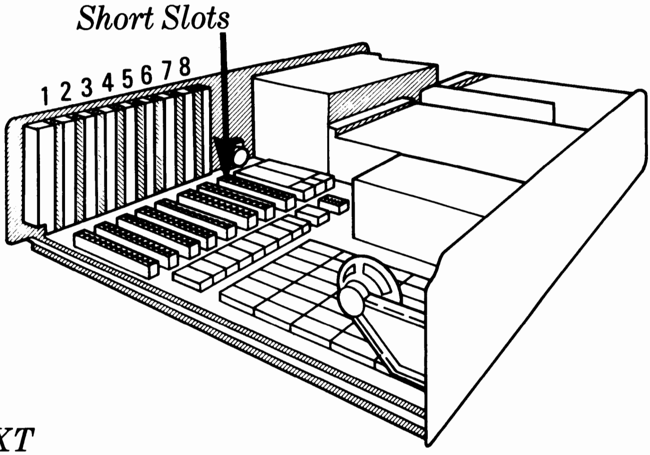
4a. Locate an empty expansion slot. In a PC or AT, any slot will do.

Cover Plate—Remove before installing InColor Card.



IBM AT
Put your Hercules InColor Card in any vacant slot.

4b. In an XT, it must be a slot that can accommodate a full length card, slots 1 through 5. If you've removed another display card to install the InColor Card, you may be able to use the vacant slot. Slots 7 and 8 are short slots and will not accommodate your InColor Card.



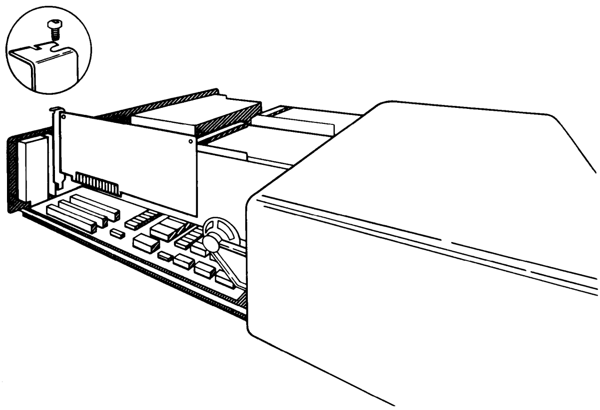
IBM XT

5. Remove the screw holding the metal cover plate on the back panel, behind the slot position you have chosen for the graphics card. Remove the cover plate.

The label identifying the InColor Card's video output port can remain attached to the card. Thread the label through the slot in the chassis when you insert the InColor Card. The label is affixed so that it is less likely for an improper monitor to be attached.

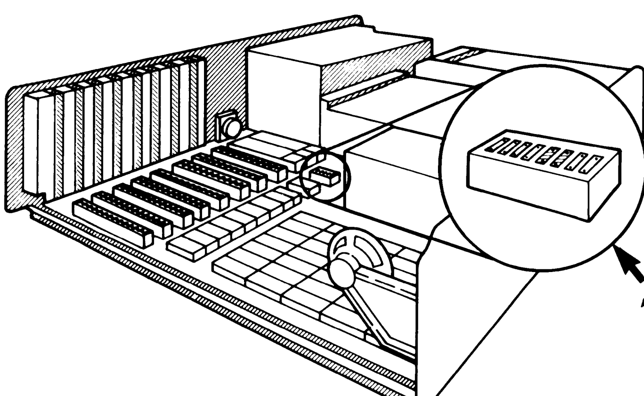
6. Firmly press the Hercules InColor Card down into the slot. It may be useful to rock it gently back and forth to make sure the card is snugly in place.

7. Replace the screw to secure the card's mounting bracket. (Make sure the bracket is seated properly while tightening the screw—otherwise it may bend and distort the monitor and printer port connections.)



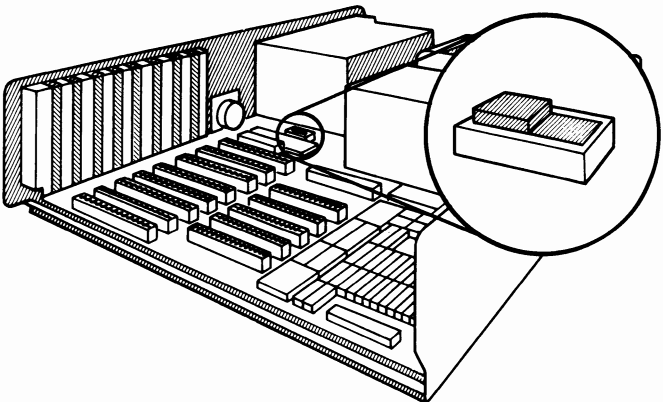
8a. If you are using an IBM PC or XT, or compatible, make sure that switches 5 and 6 of Switch Block 1 on the motherboard are set for the monochrome display as shown (5 and 6 both off):

Note: If you are unsure about how to set the switches to off, consult your computer's documentation.



IBM XT
Set Switches 5 and 6 off.

8b. If you are using an IBM PC/AT, set the motherboard display switch (SW1) to the monochrome position, as shown:



IBM AT Switch Block
Slide the switch toward the back of the machine for monochrome mode.

After setting the AT's switches, run the setup program on the AT Diagnostic disk, specifying a monochrome display.

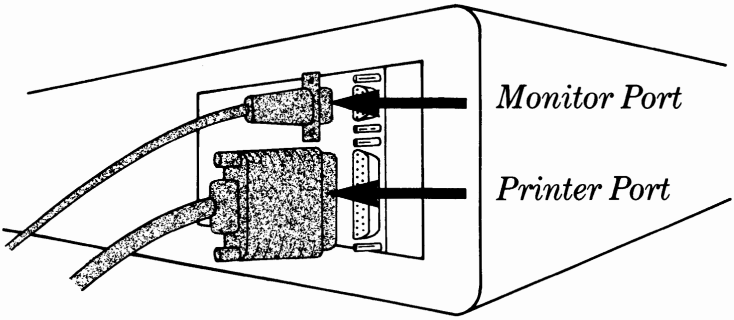
For more details on these switch settings, or for the settings for compatibles, refer to the owner's manual for your computer.

Note: It is important to stress that while the Hercules InColor Card displays color graphics on an enhanced RGB display, **your computer "sees" it as a monochrome adapter.** The display switch settings **must be set to monochrome** for correct operation of your Hercules InColor Card and enhanced RGB monitor.

Note: You do **not** need to reset the memory configuration switches for your computer to include the video memory on the InColor Card.

9. Replace your computer's cover and the screws that hold it in place.

10. Plug the enhanced RGB display and printer into the connections provided.



Note: Do not attempt to plug a monochrome display adapter into the Monitor Port on the Hercules InColor Card. Severe damage to the display unit can result. See the section "What You Need" earlier in this chapter for monitor specifications.

Setting Up

Testing the InColor Card

Testing Text Display and the Printer Port

You can easily test the text display and printer port, just by running some text-mode software that you use frequently— a word processing program will do the job most effectively.

Run the software to make sure that all menus and text displays appear correctly. Try printing a document to make sure that the parallel port is operating correctly.

Alternatively, you can run your computer's diagnostics program to check the new set up. To do so, see the **Guide to Operations** for the IBM PC, XT or AT (you'll find it covered under the chapter heading "Problem Determination" or "Testing your System"). Note that the IBM Diagnostics program identifies the Hercules InColor Card as an IBM Monochrome Display/Printer Adapter.

Testing Graphics and RamFont Modes

To check out the graphics and RamFont text displays of your InColor Card, make sure your copy of the "Programs and Fonts" disk is in the current drive, then type:

TEST [ENTER]

Follow the instructions on the screen. When you finish, you are ready to use your Hercules InColor Card.

Consult the InColor Card Software Manual for instructions.

In Case of Difficulty

If you encounter any difficulties when testing the InColor Card, go back through the installation procedure step by step to make sure you've setup your computer and installed the card correctly.

If you run into any display problems while using the InColor Card, be sure to run the tests described in this section again. They'll help determine whether the problem is with the InColor Card or in some other part of your system.

While most users don't encounter any problems in day-to-day use, Appendix A, "Troubleshooting," details potential problems and their solutions. If you're still having difficulty after testing the card and re-tracing the installation procedure, you may want to turn there. Finally, technical support is available by calling Hercules at 415/540-6000.

Chapter 3—User Features

Text, Graphics and RamFont Modes

Introduction

To understand RamFont, it's good to start with a clear idea of how it differs from the more traditional display modes.

Text Mode

In **text mode,** items that appear on screen are selected from a set of pre-programmed characters stored in the computer's **character generator.** There are 256 such characters: upper and lowercase letters, numerals, graphics elements (lines, corners, etc.), punctuation marks, and a limited number of foreign language and technical characters. The characters are displayed in 25 rows and 80 columns.

Graphics mode

In this mode, software can display images covering the whole screen by controlling each of the individual dots (pixels) that make up the screen. This is also referred to as "bit-mapped" mode, since each pixel is mapped to a specific bit in the display memory. Images are constructed by software as a map of pixels. Graphics mode operations are generally slower and more involved than in text mode.

Hercules RamFont Mode

RamFont mode shares some characteristics of both modes, combining the speed of text mode with the versatility of graphics mode.

Like text mode, RamFont utilizes pre-defined characters. However, the characters are not stored within the computer itself; instead, they are loaded from files on disk.

The "Programs and Fonts" disk includes a number of different character sets for use in RamFont mode. Details for using these are described in the separate "Programs and Fonts" pamphlet.

RamFont may be used in two modes.

4k RamFont mode is essentially a replacement for standard text mode; it lets you load any font and use it with standard text software.

48k RamFont lets software developers create programs that display a variety of type styles (or even graphic objects) on screen. For applications such as desktop publishing, 48k RamFont is a faster, more flexible alternative to graphics mode.

RamFont characters are stored in a specially organized video display memory called the RamFont buffer, and these characters can be accessed at any time.

Many types of programs can benefit dramatically from the capabilities of RamFont mode. Multilingual and scientific text processing are obvious examples. Any application in which displayable items consist of a static set of as many as 12,288 elements, will benefit from the speed and ease of use of this unique mode.

User Features

Software and RamFont

Software that Uses RamFont

There are currently a wide variety of applications available or in development that take full advantage of the InColor card's RamFont capabilities. The **Software and Hardware Compatibility Guide** included with your Hercules InColor Card lists most of these products. If you have a question about any product not included on the list (perhaps because it is too new) please contact the software manufacturer directly for information on compatibility with the InColor Card.

RamFont and Memory-Resident Software

"Terminate and Stay Resident" software such as Sidekick™, Turbo Lightning™, Keyworks™, and Ready™ all work properly with the Hercules InColor Card operating in text mode or in standard character size (9 × 14) 4k RamFont mode. However, such programs may produce unpredictable results when the card is in 48k RamFont mode or displaying an unusual character size in 4k RamFont.

We recommend that you do not use any such programs while running any applications which use the 48k RamFont mode or non-standard 4k RamFont mode.

HPRINT and RamFont

HPRINT does not print an accurate representation of the screen in RamFont mode, nor do any of the other screen dump programs intended for the Hercules Graphics Card's graphics mode.

If the Hercules InColor Card is in 4k RamFont mode, with 9 × 14 characters, the **[Shift][PrtScr] space bar** option will print the text on the screen, though not necessarily in the font shown.

If the card is in any other RamFont configuration, the result of a text mode screen dump is unpredictable. Executing a graphics mode screen dump when the Hercules InColor Card is in RamFont mode has no useful effect.

Loading Fonts for Text Mode Programs

On your "Programs and Fonts" diskette, the directory \RAMFONT contains a number of font files (indicated by a .FNT suffix). To see the list of fonts, make sure your copy of the "Programs and Fonts" disk is in your currently selected drive, then type:

DIR \RAMFONT* .FNT [ENTER]

This command displays a directory of all available fonts.

The program RAMFONT.COM lets you load any font file on the disk.

To load a font, type:

CD \RAMFONT [ENTER]
RAMFONT (filename).FNT

The font loads, then the screen goes blank and the DOS prompt returns in the new typeface. The font is now loaded, so any text mode application that you run displays this selected font.

You can try all the different fonts by scanning the directory and using the RAMFONT program to load the various font files.

You can use the 4k RamFont mode for any (or all) of your text software (word processing programs, database applications, etc.).

User Features

Graphics Programming from High Level Languages

In order to obtain InColor Card compatible graphics from programs written in most implementations of high-level programming languages (such as BASIC, Pascal, and so on), you must use a library of assembly-language graphics subroutines, such as GRAPH-X® from Hercules.

For information on Hercules graphics compatible high level languages or graphics tool kits that support the language of your choice, see the Programming section of the Software and Hardware Compatibility Guide that came with the card.

Section Two

Technical Reference Section

Operations
Hardware Features

2

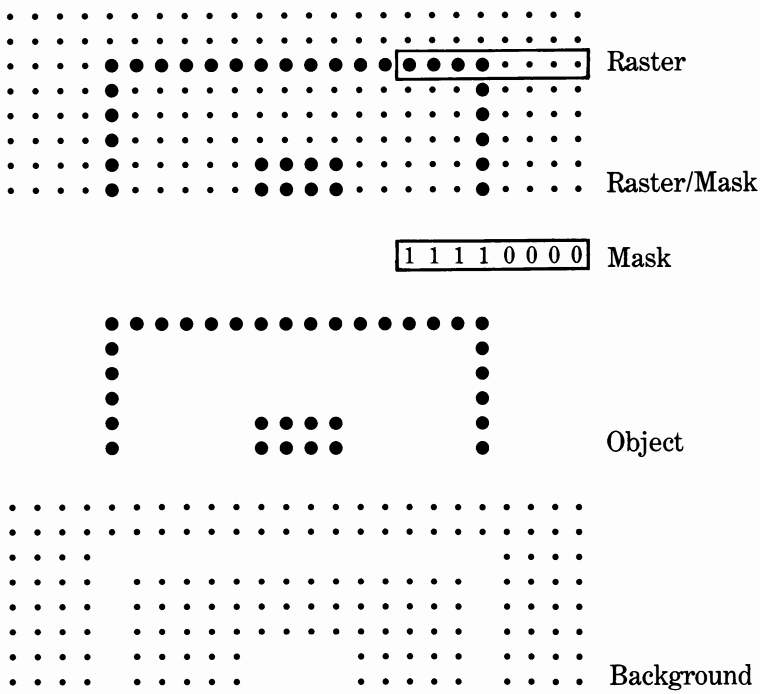
Chapter 4—Operations

Theory of Operations

Terminology

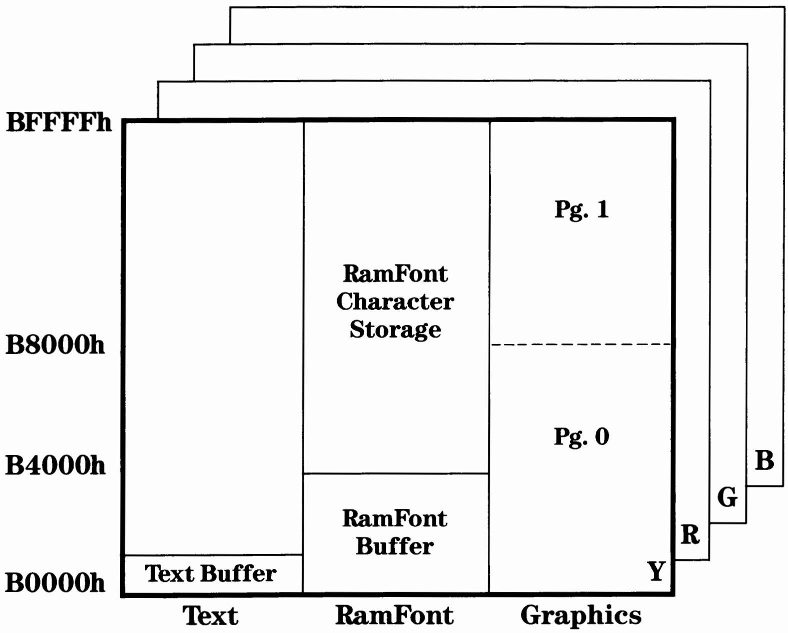
Color graphics operations can be thought of as a series of operations on a set of rasters. A **raster**, in this context, is defined as a grouping of horizontally adjacent screen locations 8 pixels wide. A **pixel** is defined by four bits, each bit of a pixel's value being held on a separate memory plane. An **object** is a map of an image to be created, made up from one or more adjacent rasters. Operations on a given object are broken up into repeated operations on the rasters included in that object.

Associated with each object is a **mask** which specifies the location of the object within the associated raster. The mask is useful when performing one operation on the object and another operation on the **background** (the area within the involved rasters but not included in the object itself).



Operations

The InColor Card translates CPU operations into raster operations. For the fastest possible raster operations, each bit of CPU memory access operates on a pixel independently of the pixel's color value. To accomplish this, a plane-organized pixel map is used, in which each raster is stored in four blocks of memory (each associated with a plane). A single raster resides at the same offset address in all four blocks.



Hercules Card Memory Map Multi-Formatted Video Memory

Operations

Raster Read:

The read operation is a tradeoff of speed over generality. For every CPU read operation, an 8-pixel raster (the target raster) is copied into a 32-bit latch called the **source latch.** Each pixel is then compared with the background value in the read/write color register (R1Ah).

The result of this comparison, called the **background bit mask** (BBM) is returned to the CPU at the end of its read cycle. Software can control the polarity of bits in the BBM—set if equal to background, or cleared if equal to background.

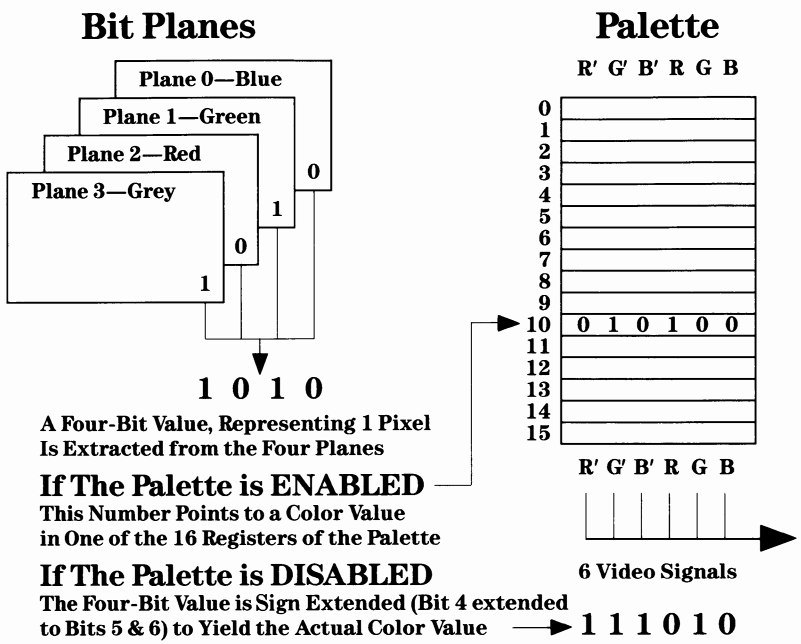
Raster Write:

The most straightforward way to write a raster is to specify a coordinate and a pixel value. A raster write is often performed on a known object against a random backgound. The object is sometimes made up of pixels of different colors.

The InColor Card provides four kinds of raster write operations. For all of them, the CPU write data is the **object bit mask.** The content of the read/write color register provides the color value, and the source latch is assumed to hold the current value of the raster to be modified (the target raster).

Palette:

Pixel values translate into colors by indexing into a table called the **palette.** The enhanced RGB monitor offers 64 color choices, but there are only 16 pixel values. Therefore, the palette is a 16 register file. The final color is coded as the on/off status for each of the six input signals to an enhanced RGB monitor. The palette is alterable by the CPU to change the range of displayable colors.



Bit Planes
Plane 0—Blue
Plane 1—Green
Plane 2—Red
Plane 3—Grey
0
1
0
1
1 0 1 0
A Four-Bit Value, Representing 1 Pixel Is Extracted from the Four Planes
If The Palette is ENABLED
This Number Points to a Color Value in One of the 16 Registers of the Palette
If The Palette is DISABLED
The Four-Bit Value is Sign Extended (Bit 4 extended to Bits 5 & 6) to Yield the Actual Color Value
Palette
R′ G′ B′ R G B
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0 1 0 1 0 0
R′ G′ B′ R G B
6 Video Signals
1 1 1 0 1 0

Palette

Exceptions:
The palette can be bypassed so the pixel value itself becomes the color specification. The 4 bit pixel value is sign extended to make up the 6 bit color code.

Monochrome Compatibility

Monochrome software is inherently faster than color software because it does not have to manipulate color information. An object mask is used to modify the target raster directly. In a plane organized color map, this must be done once for each plane. Such an operation is often done in an algorithmic loop causing the loop to be repeated four times once for each plane.

With the InColor Card, all memory planes are operated on in parallel so that algorithmic loops need be entered only once. The settings of the Read/Write Control, Read/Write Color and Exception registers modify the monochrome object mask from the CPU for each plane. Hence the overhead is only in setting up the registers once.

Running Hercules Graphics Card Plus Software on the InColor Card

To run monochrome software, all the memory planes must be synchronized. This synchronization is achieved on power up, and can be restored by programming the InColor Card specific registers to their power-on states.

Running InColor Card Software on the Hercules Graphics Card Plus

Software written for the InColor Card will have a predictable appearance when run on the Hercules Graphics Card Plus. This appearance can be simulated on the color screen by setting all the InColor Card specific registers to their power-on default states. Software that is intended to run on both monochrome and color cards can be tested in this way.

Operations

CPU Context

It is important to note that many CPU operations imply a read followed by a write. Typically, these are the logical, arithmetic, and shift operations (AND, OR, NOT, XOR, ADD, SUB, INC, DEC, SHR, SHL...). These instructions will automatically put the target raster into the source latch and return the background bit mask during their read phase. The Read/Write Control and/or Read/Write Color registers are manipulated to yield the backgound bit mask that can most efficiently be combined with the object mask for the ensuing write.

16-bit CPU Operations

A 16-bit CPU operation cannot be used in raster operations on the InColor Card. When the CPU does a 16 bit read-modify-write, it does 2 successive reads followed by 2 successive writes. This means that at the beginning of the write the source latch will be holding the raster from the second read. For example, MOVSB, not MOVSW is to be used to copy rasters within the buffer.

Operational Details

Display Modes

The display buffer is located at the offset:

B0000:BFFFFh (64k-bytes)

For each of the display modes (Text, RamFont and Graphics), this buffer is organized differently.

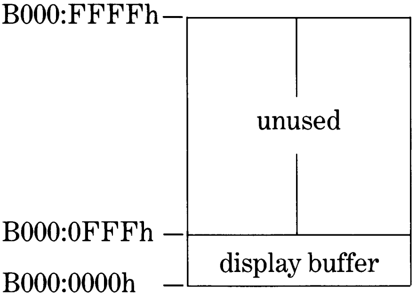
Text Operation

In text mode, the display buffer stores the character and attribute codes for the displayed characters. The offset of text mode storage is:

B0000:B0FFFh (4k-bytes)

Text display is 80 characters wide and 25 lines high. All characters are stored contiguously, left to right, top to bottom. Each line uses 160 bytes of memory, or two bytes per character. One byte is used for the character code; the other is used for the attribute.

Text Mode Memory Mapping



In text mode the display buffer always occupies the first 4k at the start of the Hercules video memory. Character data is held in the character generator ROM. Attributes are specified as 8 bits, and characters are coded as the standard ASCII 8 bits.

Text Mode Offsets

The following formulas are used to determine the address offset of character codes and attributes for particular character positions. (In these formulas, ROW is a row number between 1 and 25, COLUMN is a column number between 1 and 80.)

Character code offset = 160 * (ROW – 1) + 2 * (COLUMN – 1)
Attribute code offset = 160 * (ROW – 1) + 2 * (COLUMN – 1) +1

Operations

Text Mode Display Buffer Organization

Segment at B0000	Offsets	+0	+1	...	+158	+159
Offset	*Row #*					
00	1	char 1	attrib 1	...	char 80	attrib 80
0A0h	2	char 1	attrib 1	...	char 80	attrib 80
.	.	.	.		.	.
.	.	.	.		.	.
.	.	.	.		.	.
0f00h	25	char1	attrib 1	...	char 80	attrib 80

The Hercules Graphics Card Plus has a hardware character generator which provides 256 different characters specified by the character codes. It conforms to the standard IBM PC character font. There is an attribute decoder which can underline, reverse video, blank, intensify, or blink any character.

Text Mode Attributes

The Hercules InColor Card has a hardware character generator which provides 256 different characters specified by the ASCII codes (conforming to the standard IBM PC character font). There is an attribute decoder which supports all special character display functions: underline, reverse video, blanking, intensity, and blinking.

The attribute decoder follows these rules:

7	6	5	4	3	2	1	0	Attribute Codes
B	0	0	0	I	0	0	0	Blank
B	0	0	0	I	0	0	1	Underline
B	0	0	0	I	1	1	1	Normal Display
B	1	1	1	I	0	0	0	Reverse Video

Notice that some of the attributes listed in the table are mutually exclusive. However, bits 3 and 7, which control character intensity, background intensity and blinking, can be set independently of the attributes listed.

See Appendix E for a complete table of display modes and attributes.

Operations

Character Intensity
Bit 3 controls the intensity for any character, regardless of the other attribute code settings. As shown in the table:

I = 0 for normal intensity
I = 1 for high intensity

Background Intensity
If the blinker is off (Display Mode Control Port bit 5 set to 0) then bit 7 has these effects:

B = 0 for normal background
B = 1 for high intensity background

Blinking
If the blinker is on (Display Mode Control Port bit 5 set to 1) then the contents of bit 7 have these effects:

B = 0 for no blink
B = 1 for blinking

For information on how character colors are selected in text mode, see the section "Color Attributes in Text and RamFont Modes," page 38.

RamFont Operations
RamFont operations are themselves divided up into two distinct modes of operation: 4k RamFont and 48k RamFont. Activating RamFont and selecting the mode is performed by manipulating the InColor Card's xMode Register.

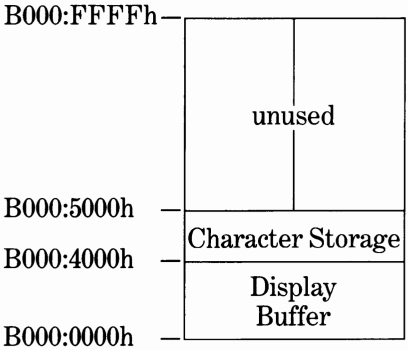
In discussions of RamFont, it is useful to think of fonts being organized into 12 typefaces, each consisting of 256 characters each. In 4k RamFont, only the first of these 12 typefaces can be used (T1). In 48k RamFont, the groupings may be thought of as a set of twelve alternate typefaces (T1-T12); however, they may in fact consist of purely graphical characters with no alpha-numeric meaning. Furthermore, 48k RamFont mode can be exploited in a way that allows four separate fonts to be stored in the four separate memory planes. This could accommodate 12,288 (4 x 12 x 256) characters.

Operations

4k RamFont Mode

4k RamFont mode is easy to understand, since its operation is directly comparable to standard text mode. Characters are represented by 8-bit codes, just like the standard ASCII codes. Of course, the assignment of a pixel outline to display for each code is in the hands of the programmer (from software) or the user (through the FONTMAN™ and RAMFONT programs on the "Programs and Fonts" disk). Character and attributes are specified just as in text mode: the character code followed by the attribute code. Text mode applications will run normally in 4k RamFont mode as long as a standard 9 x 14 character matrix is used.

4k RamFont Mode Memory Mapping



4k RamFont mode requires that the character data be held in the first font "type" at B4000h. The display buffer area is 16k. The mapping of characters and attributes is identical to the text mode—8-bit character and 8-bit attribute bytes. When character data is loaded into RamFont storage, this memory must be treated as a bit-map, with the read/write control and color registers set accordingly.

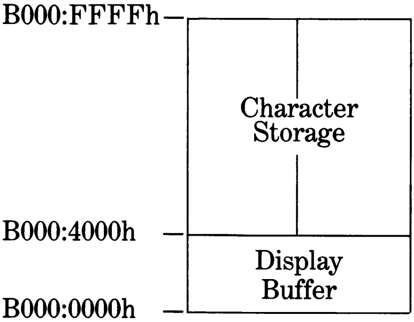
4k RamFont Attributes

In 4k RamFont mode, the attribute decoding is compatible with the text mode of the Hercules InColor Card (see "Text Mode Attributes," page 31).

48k RamFont Mode

In 48k RamFont mode, the character code possibilities are expanded: instead of an 8-bit code, a 12-bit character code is used. Values for the character codes may be in the range from 0 through 0BFFh. This allows a total of 3072 characters. In 48k RamFont, characters may be up to 16 dots tall, and either 8 or 9 dots wide.

48k RamFont Mode Memory Mapping



48k RamFont mode uses 48k of memory between B4000h and BFFFFh for character storage. This 48k can be accessed separately on the four planes for a maximum of 192k of character storage. This data is accessed using a 12-bit character/4-bit attribute code. When character data is loaded into RamFont storage, this memory must be treated as a bit-map, with the read/write control and color registers set accordingly.

The lowest 16k-bytes (8k words) are reserved for use as the display buffer. Character/attribute codes stored in this area determine which characters are displayed in the corresponding screen locations. The portion of the display buffer used depends on the size of the characters being displayed.

To determine the amount of the display buffer that will be required for a given character size, use the following formula. H (height) is a number between 4 and 16, W (width) is either 8 or 9.

Memory Used (in bytes) = (350 / H) * (720 / W) * 2

The RamFont character storage area begins at offset 4000h. Sixteen bytes of storage space are reserved for each character, regardless of the actual size of the characters to be displayed. The lowest address in the sixteen-byte character matrix holds the top row of the character. The character storage area occupies 48k bytes, so the last character (BFF) begins at offset 0FFF0h and ends at 0FFFFh. To find the offset of a character in RamFont storage, use the following formula. T equals the Typeface number (1 to 12) and C equals the character number (0 to 255).

Character Offset = 4000h + [(T – 1) * 256] +C

In 4k RamFont mode, you can access only the first 256 characters in a RamFont file (Typeface 1). These characters are stored in the lowest 4k bytes of the storage area. In 48k RamFont mode, you can utilize up to 3072 characters, filling the full 48 k-bytes of the character storage area.

RamFont Mode Display Buffer Organization

```
GB222       SEGMENT AT 0B000H
BUFFER      DW      8*1024 DUP    (?)
RAMFONT:
            ORG     4000H
CODE000:
            DB      ?             ; Top Row
            DB      ?             ; 2nd Row
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?
            DB      ?             ; Max row
CODE001:
            DB      16DUP         (?)
;
            ORG     0FFF0H
CODEBFF:
            DB      16 DUP        (?)
GB222       ENDS
```

48k RamFont Attribute/Character Words

In 48k RamFont mode, the highest 4 bits of a word hold the attribute specification. It may help to think of the buffer as a collection of attribute/character word values. The most significant 4 bits of each word specifies the attribute, the remaining 12 bits specify the character to display at the selected position. It is helpful to again divide the 12-bit character value into a 4-bit **type pointer** and an 8-bit ASCII code.

48k RamFont attributes are decoded as follows:

Attribute	Value	Remarks
Normal	0000h	
High-Intensity	8000h	Blink-enable bit set
Boldface	8000h	NEW ATTRIBUTE— Blink-enable bit clear
Blink	4000h	Blink-enable bit set
Reverse	4000h	Blink-enable bit clear
Overstrike	2000h	NEW ATTRIBUTE
Underline	1000h	

Notice that there is no blank attribute in 48k RamFont mode. However, there are two new attributes available:

Boldface duplicates each pixel of the character one position to the right, making the character appear thicker. This attribute is most useful with characters built of single-dot elements; wider characters may look smeared. The Boldface attribute will not cause smearing between adjacent characters.

Overstrike draws a row of illuminated pixels across the character matrix, at the position specified in the Strike Register (R16h). This attribute is useful for indicating text deletions that should not be completely excised.

Note that in 48k RamFont mode, the high intensity and boldface attributes are mutually exclusive, as are blink and reverse video. These particular attribute combinations were made mutually exclusive because the reverse video and boldface attributes can both be simulated by appropriate font design. For example, it is possible to have blink and reverse video simultaneously, simply by designating a reverse-video font and using it with the blink attribute.

The blink-enable bit in the Display Mode Control Port (bit 5, port 03B8) determines which attribute set will be active.

Here are two simple examples of the effects of attribute/character words:

Attribute Code	+	Type Pointer	+	ASCII Code	=	Resultant 48K RamFont Code
0000		1000		01000001		0841h
0010		1010		11110001		2AF1h

In the first example, the attribute code (the four most significant bits of the high byte) represent the "normal" attribute, the type pointer (the four least significant bits) select typeface 9 (the pointer value equals the typeface number minus one), and the low byte represents the ASCII code for 'A' (41h).

The second example shows a character with an ASCII code of F1h, located in typeface 11, with the overstrike attribute selected.

See Appendix E for a complete table of display modes and attributes.

Character Width in RamFont

In either RamFont mode, the character matrix can be either 8 bits or 9 bits wide. This means, in effect, that you may have either an 80-column (9-bit) or 90-column (8-bit) screen.

When the 9-bit mode is selected, the hardware automatically generates a blank 9th bit to the right of the character, except in the range of ASCII codes 0C0h-0DFh. For these 48 characters (the ASCII graphics characters), the hardware automatically duplicates the contents of the 8th bit into the 9th. This lets those ASCII line elements intersect. This automatic 9th-bit generation is the same as in standard text mode.

The 8-bit wide RamFont mode disables this automatic 9th bit handling. If you need to create a set of elements that will combine to form a larger image, you need to use 8-bit wide RamFont mode so that a 9th bit will not automatically be inserted.

The Configuration Switch in RamFont Mode

You need to set the InColor Card's configuration switch in order to use RamFont mode. For 4k RamFont mode, the switch may be set to either HALF (OUT 03BF,1) or FULL (OUT 03BF,3). For 48k RamFont mode, the switch may be set to HALF, provided that no more than 1024 characters (equivalent to four standard ASCII fonts) are required. Otherwise, it must be set to FULL, since memory above B8000h will be used.

Color Attributes in Text and RamFont Modes

Character mode color attributes are controlled by the Attribute Selector Bit in the Exception Register (bit 5, register 17h).

The most significant 4 bits of the attribute byte indicate a background color value, and the least significant 4 bits of the attribute byte define a foreground color value.

For text mode and 4k RamFont mode this bit selects between the "normal" attribute set (normal, blink, blank, reverse, high-intensity and underline) and the alternate attribute set which yields a foreground and background color for a character.

The full range of sixteen background values is only available when the blink-enable (bit 5 of the Display Mode Control Port—03B8) bit is off. When the blink enable bit is set, the range of displayable background colors consists of the values 0-7 only. For any value in the range 8-15, the most significant bit is ignored.

For example, when:

Attribute Selector Bit = 1 (normal attribute set)

Then:

- Attribute byte of 07h indicates normal character display
- Attribute byte of 70h indicates reverse video
- Attribute byte of 01h indicates underline

And when:

Attribute Selector Bit =0 (alternate attribute set)

Then:

- Attribute byte of 07h indicates black background, gray foreground
- Attribute byte of 70h indicates gray background, black foreground
- Attribute byte of 01h indicates black background, blue foreground

If the palette is enabled, the color value represents an index into the palette.

For 48k RamFont mode, the Attribute Selector Bit selects between two types of attributes. The normal attribute set(s) consists of either normal, high-intensity, blink, overstrike, and underline **or** normal, boldface, reverse, overstrike and underline, depending on the setting of the blink enable bit. The alternate attribute set interprets the four bit attribute code as a mask of the memory planes. This means that up to 48 "types" are available to store character data for use in 48k RamFont mode. Access to the desired font is accomplished by using the high four bits of the attribute byte as a mask for the memory planes.

For example, when:

Attribute Selector Bit = 1 (normal attribute set)

Then:

- Attribute byte of 01h indicates a normal character stored in the second font type.
- Attribute byte of 25h indicates an overstrike character stored in the sixth font type.

And when:

Attribute Selector Bit = 0 (alternate attribute set)

Then:

- Attribute byte of 01h indicates the character stored in the second font type with none of the planes masked off.
- Attribute byte of 25h indicates the character stored in the sixth type with the "green" plane masked off.

If the same character data is loaded into each of the four memory planes, then the high four bits of the attribute byte indicate a complement color value.

Note that to achieve the high-intensity attribute in 48k RamFont mode, character data should be loaded into planes 0, 1 and 2 only. Initialize the foreground value in the Read/Write Color Register to 7, rather than the default value of 15, before loading the font data. This will prevent character data from loading to plane 3. Map Palette index 8 to your background color to suppress high-intensity background.

See Appendix E for a complete table of display modes and attributes.

Operations

Read/Write Exception:

A read/write to the area B000:0000 to B000:3FFF (character buffer) in either text or RamFont modes will ignore the color control settings and access the planes as if they were synchronized.

The read operation in text mode needs to have the source latch unprotected. Furthermore, the target raster that might have been in the source latch is overwritten.

The same exception holds true in both 4k and 48k RamFont modes, although for access to the area B000:4000h to B000:FFFFh (RamFont character storage) all the color and control settings are followed.

Cursor:

The cursor color is programmable independently of the character attribute (color).

It is not possible to specify the value 0000b for the cursor color. If the value 0000b is programmed in, 0111b will be used instead. In text and 4k RamFont modes, using the normal attribute set, the cursor color value is affected by bit three (character intensity) of the attribute byte. The most significant bit of the cursor color value always maps to the value of bit 3 in the attribute byte. When the alternate attribute set is selected, the programmed value of the cursor color will be used directly.

This same scheme applies to the behavior of the underline attribute in text and 4k RamFont modes.

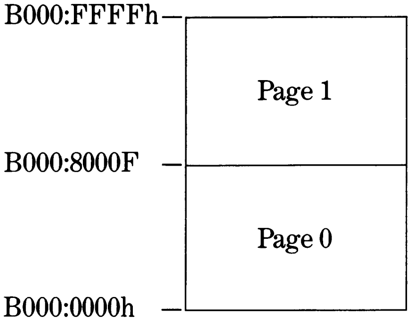
Underline and Overstrike:

The colors of these two attributes are programmable independently of the character foreground and background colors. When the same scan line is selected for any combination of overstrike, underscore, and cursor, the color priorities are cursor, overstrike, and then underline.

Operations

Graphics Mode

Graphics Mode Memory Mapping



In graphics mode the 64k buffer area is divided into two 32k pages of four planes each. Data is written directly to the bit-map, with the soft-latch and read/write control registers determining how the data will be represented on the individual planes.

The 64k-byte buffer area is divided into two 32k-byte pages, with these offsets:

Page 0 = B0000-B7FFF
Page 1 = B8000-BFFFF

With the Hercules InColor Card in bit-mapped (graphics) mode, the display buffer stores two screens (one screen per page). These two pages can be alternately displayed.

Each page is made up of four 8k-byte frames. Since the resolution of the graphics mode display is 720 pixels wide by 348 lines deep, each frame contains 87 lines (348 lines divided by four frames). Each line requires 90 bytes of memory (720 pixels per line divided by 8 pixels per byte). The total storage space used in each frame is 7830 bytes (90 by 87). The remaining memory (the highest 362 bytes per frame) are not used; writing to these high bytes has no effect on the display.

While each successive 90 bytes of a frame contains a specific line, the lines within an individual frame are not contiguous on the display. Instead, the frames are interleaved so that filling consecutive lines in a frame results in every fourth line on the display being illuminated.

This table summarizes the organization of lines, frames and pages of the display buffer.

Graphics Mode Display Buffer

Page	Starting Address (Hex)	Maps To Screen At Lines
0	B0000	0,4,8, ...344
	B2000	1,5,9, ...345
	B4000	2,6,10, ...346
	B6000	3,7,11, ...347
1	B8000	0,4,8, ...344
	BA000	1,5,9, ...345
	BC000	2,6,10, ...346
	BE000	3,7,11, ...347

Updating and Displaying Pages

Each page can be updated independently. When either page is being displayed, alterations to its display buffer appear on-screen immediately. Changes made to the display buffer of the page not being displayed are shown only when that page is brought to the screen.

Page selection is performed using bit 7 of the Display Mode Control Port:

0 = Display page 0
1 = Display page 1

Addressing a Specific Screen Location

To determine the memory offset of a specific screen position **(x,y)** (where x is an integer value between 0 and 719 and y is an integer value between 0 and 347), use the following formulas:

Byte offset = [2000h * (Y MOD 4)] + [90 * INTEGER (Y/4)] + [INTEGER (X/8)]

Bit position = 7 – (X MOD 8)

Example:

The offset of the position **(300,250)** is:

[2000h * (250 MOD 4)] + [90 * INTEGER (250/4)] + [INTEGER (300/8)]

= [2000h * 2] + [90 * 62] + 37

= 4000h + 5617

= 4000h + 15F1h

= 55F1h

And the bit position is:

7 – (300 MOD 8) = 7 – (4) = 3

Chapter 5—Hardware Features

Hercules InColor Card—Input/Output Ports

6845 Index Register (03B4h—write only)

```
7 6 5 4 3 2 1 0
x x x └─┴─┴─┴─┘  Index of the 6845 register to access through data
                 port.
```

6845 Data Register (03B5h read/write)

```
7 6 5 4 3 2 1 0
└─┴─┴─┴─┴─┴─┴─┘  Data for 6845 register pointed to by index
                 register.
```

Display Mode Control Port (03B8)—write only)

```
7 6 5 4 3 2 1 0
│ x │ x │ x │ x
│   │   │   └──── 0 = text mode, 1 = graphics mode (see note)
│   │   └──────── 0 = disable video, 1 = enable video
│   └──────────── 0 = blink off, 1 = blink on
└──────────────── 0 = graphics page zero, 1 = graphics page one
```

Note: Be particularly careful when changing between text and graphics modes. The 6845 must be programmed with the proper parameters (the first 12 parameters listed on page 48) whenever bit 1 of the Display Mode Control Port is changed. When software switches between text and graphics modes, the monitor is subject to some undefined horizontal and vertical frequencies. It is recommended that you do not use high-level languages to control the Display Mode Control Port, bit 1 or to initialize the 6845 timing registers. See the sample code for an illustration of the correct way to change modes.

Set Light-Pen Flip-Flop (03B9h—write only)

```
7 6 5 4 3 2 1 0
x x x x x x x x   Any write to this port sets the light-pen flip-flop.
```

Display Status Port (03BAh—read only)

```
7 6 5 4 3 2 1 0
|  |  |  |  |  x  |  └─  0 = normal character, 1 = Sync
|  |  |  |  |     └────  Light Pen Flip-Flop
|  |  |  |  └──────────  0 = dots off, 1 = dots on
|  └──┴──┴─────────────  Version I.D. bits (see note)
└───────────────────────  0 = vertical sync, 1 = active display
```

Note: I.D. Mask = 01110000b
I.D. Code = 01010000b

Reset Light-Pen Flip-Flop (03BBh—write only)

```
7 6 5 4 3 2 1 0
x x x x x x x x   Any write to this port resets the light-pen
                  flip-flop.
```

Printer Data Port (03BCh—read/write)

```
7 6 5 4 3 2 1 0
└─┴─┴─┴─┴─┴─┴─┴─  8 bits of data to send to the printer
```

Printer Status Port (03BDh—read only)

```
7 6 5 4 3 2 1 0
| | | | | x x x
| | | | └──────── 0 = printer error, 1 = normal operation
| | | └────────── 0 = printer not attentive, 1 = printer is listening
| | └──────────── 0 = normal operation, 1 = out of paper
| └────────────── 0 = accepting data, 1 = ready for more
└──────────────── 0 = printer is busy, not selected or in error, 1 =
                    normal
```

Printer Control Port (03BEh—read/write)

```
7 6 5 4 3 2 1 0
x x x | | | | └── 0 = strobe printer to accept data, 1 = release
      | | | |       strobe
      | | | └──── 0 = auto line feed, 1 = remote line feed
      | | └────── 0 = initialize the printer, 1 = release the printer
      | └──────── 0 = deselect the printer, 1 = select the printer
      └────────── 0 = mask off IRQ7, 1 = enable IRQ7 when ready
                    for data
```

Configuration Port (03BFh—write only)

```
7 6 5 4 3 2 1 0
x x x x x x | └── 0 = prevents graphics mode, 1 = allows graphics
            |       mode
            └──── 0 = freeze graphics page one from
                    reading/writing
                  1 = allows random access to graphics page one
```

Hardware Features

The 6845 Registers

The 6845 CRT controller chip is found on many common IBM-compatible display adapters, including the IBM monochrome and color cards and the Hercules Graphics Card Plus. The function of the 6845 chip is to control the operating frequencies and display format of the monitor, as well as manage the cursor and light-pen. In addition to these standard functions of the 6845, a number of features that are unique to the Hercules InColor Card have been implemented as "extensions" of the 6845. RamFont mode, color control, latch control, and the palette are all accessed via the standard 6845 index and data registers.

To properly initialize a 6845 register, you must send the desired register number to the 6845 index register (03B4h) and then the value for that register to the 6845 data register (03B5h).

The table on the following page represents the programmer's view of the 6845 on the Hercules InColor Card. The registers with indices 0 through 11h are the standard registers found on any 6845-based video board. Of these 18 registers, the first 12 govern the CRT timing and the cursor, and must be initialized by software whenever there is a mode change (bit 1 of the Display Mode Control Port). They are loaded into the 6845 one by one via a single input/output port. Another output port is used to specify which parameter will be loaded next. **It is extremely important that when the 6845 timing registers are programmed, all of the parameters are initialized. Improper programming of the 6845 could result in severe damage to the monitor.** No assumptions should be made about the previous states of the 6845, **all** of the first 12 programmed values should be initialized whenever there is a mode change. Registers 14h through 1Ch are the proprietary Hercules registers and can be accessed individually as needed.

The table on the next page lists the 6845 registers and Hercules extensions and the recommended values for the first 12 timing and cursor format registers for both text and graphics modes. The rightmost column lists the unit (characters, rasters, or character rows) with which the timing value is calculated. Timing values in graphics mode are based on a character size of 16 pixels wide by 4 rasters tall. In text mode, characters are 9 pixels wide by 14 rasters tall. RamFont mode timing values vary depending upon the character matrix being employed and are listed separately.

For detailed information and functional descriptions of the standard 6845 registers, consult a 6845 data book.

Hardware Features

6845 Registers

Index	Read	Write	7	6	5	4	3	2	1	0	NAME	TEXT	GRAPHICS	UNIT
0	X	0									Horizontal Total	61h	35h	char
1	X	0									Horizontal Displayed	50h	2Dh	char
2	X	0									Horizontal Sync Position	52h	2Eh	char
3	X	0	0	0	0	0					Sync Width	0Fh	07H	char width
4	X	0	▒								Vertical Total	19h	5Bh	char ht
5	X	0	▒	▒	▒						Vertical Total Adjust	06h	02h	scan line
6	X	0	▒								Vertical Displayed	19h	57h	row
7	X	0	▒								Vertical Sync Position	19h	57h	row
8	X	0	0	0	0	0	▒	▒	1	0	Interlace & Skew	02h	02h	n/a
9	X	0	▒	▒	▒						Max Raster Addr. (per row)	0Dh	03h	scan line
0Ah	X	0	▒	0							Cursor Start Raster	0Bh	00h	scan line
0Bh	X	0	▒	▒	▒						Cursor End Raster	0Ch	00h	scan line
0Ch	0	0	▒	▒							Start Address (high)			
0Dh	0	0									Start Address (low)			
0Eh	0	0	▒	▒							Cursor Location (high)			
0Fh	0	0									Cursor Location (low)			
10h	0	X	▒	▒							Light-Pen Position (high)			
11h	0	X									Light-Pen Position (low)			
12h	X	X	▒	▒	▒	▒	▒	▒	▒	▒	unused			
13h	X	X	▒	▒	▒	▒	▒	▒	▒	▒	unused			
14h	X	0	▒	▒	▒	▒					xMode Register			
15h	X	0									Underscore Register			
16h	X	0									Overstrike Register			
17h	X	0	▒	▒							Exception Register			
18h	X	0									Plane Mask Register			
19h	X	0	▒								Read/Write Control Register			
1Ah	X	0									Read/Write Color Register			
1Bh	X	0									Latch Protect Register			
1Ch	0	0	▒	▒							Palette Register			

0 = yes
X = no
▒ = unused bit
0 = must be zero
1 = must be one

Hercules InColor Card—6845 Register Extensions Summary

xMode Register (R14h—write only)

```
7 6 5 4 3 2 1 0
x x x x x | | └─ 0 = ROM generator on, 1 = RamFont selected
          | └─── 0 = 9 bit wide matrix, 1 = 8 bit wide matrix
          └───── 0 = 4k RamFont, 1 = 48k RamFont
```

Underscore Register (R15h—write only)

```
7 6 5 4 3 2 1 0
| | | | └─┴─┴─┴─ Underscore Position
└─┴─┴─┴───────── Underscore Color
```

Value 0000 for underscore color cannot be specified. If it is programmed in, value 0111 will be used instead.

Overstrike Register (R16h—write only)

```
7 6 5 4 3 2 1 0
| | | | └─┴─┴─┴─ Overstrike Position
└─┴─┴─┴───────── Overstrike Color
```

Exception Register (R17h—write only)

```
7 6 5 4 3 2 1 0
x x | | └─┴─┴─┴─ Cursor Color
    | └───────── 0 = Palette disabled, 1 = Palette enabled
    └─────────── 0 = Alternate attributes, 1 = Normal attributes
```

Value 0000 cannot be specified for cursor color. If it is programmed in, value 0111 will be used instead.

Hardware Features

Plane Mask Register (R18h—write only)

```
7 6 5 4 3 2 1 0
| | | | └─┴─┴─┴─ Display Planes— 0 = not visible, 1 = visible
└─┴─┴─┴───────── Freeze Planes—0 = writable, 1 = read only
```

Note: In the "Display Planes" bits, the plane number (0-3) corresponds to the bit number. In the "Freeze Planes" bits, the planes are represented by the correspondingly significant bit (plane 0 by bit 4, plane 1 by bit 5, etc.).

Read/Write Control Register (R19h—write only)

```
7 6 5 4 3 2 1 0
x | | | └─┴─┴─┴─ Don't care — 0 = care, 1 = don't care
  | └─┴─────────── Write Mode (see text)
  └─────────────── Mask Polarity—0 = set bit in BBM if equal to
                                    background
                                1 = clear bit in BBM if equal to
                                    background
```

Read/Write Color Register (R1Ah—write only)

```
7 6 5 4 3 2 1 0
| | | | └─┴─┴─┴─ Foreground Color
└─┴─┴─┴───────── Background Color
```

Hardware Features

Latch Project Register (R1Bh—write only)

7 6 5 4 3 2 1 0

Protect pixel 0 (rightmost)
Protect pixel 1
Protect pixel 2
Protect pixel 3
Protect pixel 4
Protect pixel 5
Protect pixel 6
Protect pixel 7 (leftmost)

Palette Register—Read (R1Ch—read only)

7 6 5 4 3 2 1 0
x x x x x x x x Reset Palette Index

Palette Register—Write to the current Index and then increment Index (R1Ch—write only)

7 6 5 4 3 2 1 0
x x

Blue
Green
Red
Secondary Blue
Secondary Green
Secondary Red

Hercules InColor Card—6845 Register Extension Descriptions

xMode Register (R14h—write only, Power On Default = 00h)

Bit 0: "Character Mode"

0 = ROM Generator—Character set obtained from character generator chip.

1 = RamFont On—Character set(s) obtained from character storage memory starting at B4000h.

Bit 1: "Character Width"

0 = 9 Bit Wide—In RamFont mode, the 720-dot horizontal resolution of the screen is divided into 80 columns of 9 dots each. In 9 bit wide mode the hardware automatically inserts a blank ninth dot between characters, except in the ASCII range C0h—DFh where the contents of the eighth bit is duplicated into the ninth bit position to facilitate the connecting graphics characters.

1 = 8 Bit Wide—In RamFont mode, the 720-dot horizontal resolution of the screen is divided into 90 columns of 8 dots each. The hardware does not automatically handle the ninth dot.

Bit 2: "RamFont Mode"

0 = 4k Ramfont—256 characters stored in the first "Type" of character storage RAM are accessed using an 8 bit character/8 bit attribute code.

1 = 48k RamFont—Up to 12,288 characters held in character storage RAM can be accessed with a 12 bit character/4 bit attribute code.

Underscore Register (R15h—write only, Power On Default = xx)

Bits 0-3: "Underscore Position"

In 48k RamFont mode only, indicate the position in the character matrix where an underscore will be placed. A value in the range 0—0Fh is appropriate here.

Bits 4-7: "Underscore Color"

Hold a color value for the underscore line if not equal to 0. 0 value produces a color value of 0111.

Overstrike Register (R16h—write only, Power On Default = xx)

Bits 0-3: "Overstrike Position"

In 48k RamFont mode only, indicate the position in the character matrix where an overstrike will be placed. A value in the range 0—0Fh is appropriate here.

Bits 4-7: "Overstrike Color"

Hold a color value for the overstrike line if not equal to 0. 0 value produces a color value of 0111.

Exception Register (R17h—write only, Power On Default = 20h)

Bits 0-3: "Cursor Color"

Hold a color value for the cursor in text and RamFont modes if not equal to 0. A 0000 value results in the cursor color of 0111 if the character foreground is not intensified, 1111 if it is.

Bit 4: "Palette Enable"

0 = Disable the palette, the four bit color code from the memory planes determines the final color for the screen.

1 = Enable the palette, the four bit color code from the memory planes is used to index into the palette's 16 register file to determine final color for the screen.

Bit 5: "Attribute Selector Bit"

0 = Alternate attribute set—for text mode and 4k RamFont mode the attribute byte is interpreted as consisting of four (most significant) bits of background color and four (least significant) bits of foreground color. In 48k RamFont mode the alternate attribute set interprets the four bit attribute code as a mask of the memory planes. By using the alternate attribute set in 48k RamFont mode, software can manipulate four different fonts (or four different colors) for each "Type."

1 = Normal attribute set—for text mode and 4k RamFont mode the normal attribute set consists of normal, high-intensity, reverse video, underline, blink, and blank. The normal 48k RamFont mode attributes consist of either normal, high-intensity, blink, overstrike, and underscore; or normal, boldface, reverse video, overstrike, and underscore depending upon the setting of the blink bit (bit 5 of the Display Mode Control Port).

Plane Mask Register (R18h—write only, Power On Default = 0Fh)

Bits 0-3: "Display Planes"

Form a mask of the four memory planes determining which will be displayed. A high bit in the mask indicates that the plane will be visible.

Bits 4-7: "Freeze Planes"

Form a mask of the four memory planes determining which will be "frozen", or protected against modification. A high bit in the mask will freeze the indicated plane, a low bit enables modification.

Read/Write Control Register (R19h—write only, Power On Default = 40h)

Bits 0-3: "Don't Care Planes"

Form a mask determining which of the four memory planes will contribute to the Background Bit Mask. When a don't care bit is set, the result (background bit mask) of reading the video buffer will not reflect the value in the indicated plane.

Bits 4, 5: "Pixel Write Mode"

Determines the write mode for buffer access. When a byte of data is written to a video buffer segment offset, the color of the pixels will vary depending upon the setting of the pixel write mode bits. The data byte will be interpreted as follows:

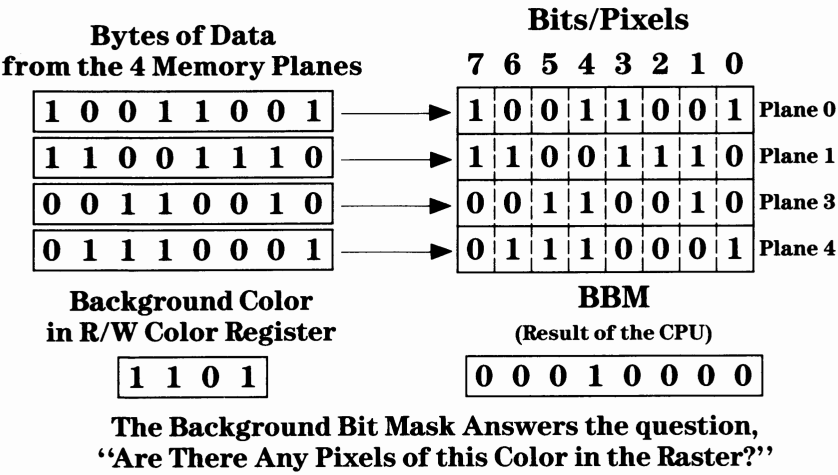
(Setting of Mode Bits)				(State of bit in data byte)	
Bit 5	Bit 4			Low Bit	High Bit
0	0	=	Write	Background Color	Foreground Color
0	1	=	Write	Source Latch	Foreground Color
1	0	=	Write	Background Color	Source Latch
1	1	=	Write	– Source Latch	Source Latch

Bits 6: "Mask Polarity"

Determines the polarity of the byte result, the "Background Bit Mask," obtained when the video buffer is read.

0 = The result bit in the Background Bit Mask will be set if the pixel equals the background color value held in the Read/Write Color Register (R1Ah, bits 4-7).

1 = The result bit in the Background Bit Mask will be cleared if the pixel equals the background color value held in the Read/Write Color Register (R1Ah, bits 4-7).



Background Bit Mask

Read/Write Color Register (R1Ah—write only, Power On Default = 0Fh)

Bits 0-3: "Foreground"

Hold a value specifying the foreground color for the pixel write operation.

Bits 4-7: "Background"

Hold a value specifying the background color for the pixel write and pixel read operations.

Source Latch Protect Register (R1Bh—write only, Power On Default = 00h)

Bits 0-7: "Protect Pixel"

Forms a pixel-wise mask of the 8 pixels in the latch such that:

0 = Allows the pixel in the source-latch at the indicated position to be updated by a subsequent pixel read operation.

1 = Protects the pixel in the source-latch at the indicated position from being updated by a subsequent pixel read operation.

Palette Register Read (R1Ch—read only)

A read of register 1Ch will reset the palette index to 0, facilitating a subsequent update of the palette.

Palette Register Write (R1Ch—write only, Power On Default—undefined

The Palette is a 16 register file which is writable with post-write auto-increment of the index for the next write.

Usage example:

```
        mov     dx,index_reg        ; 6845 index register—03B4h
        mov     al,palette          ; this register—1Ch
        out     dx,al               ; 6845 index register points at palette
        inc     dx                  ; point DX at 6845 data reg.—03B5h
        in      al,dx               ; read 1Ch—reset the palette pointer
        cld                         ; go in the right direction
        mov     cx,16               ; 16 registers in the palette
        lea     si,palette_table    ; a table of desired colors for palette

load_palette:                       ; get byte in AL
        lodsb                       ; send value, palette automatically
        out     dx,al               ; increments to point at next register
        loop    load_palette        ; do all of them
```

Read Operation and the Background Bit Mask (BBM)

The Background Bit Mask is so named because its contents reflect the state of the four planes of memory in relation to the background color value held in the Read/Write Color Register. The instruction "MOV AL,ES:[DI]" can be said to yield the Background Bit Mask in register AL. The BBM is not itself a unique location. It is a representation of what is held in the four memory planes in relation to the background color value. The BBM is simply a name for the information returned to the CPU when a CPU read operation accesses video memory.

The polarity of the BBM is determined by the "Mask Polarity Bit" (R19h, bit 6). For example, if the background color in the Read/Write Color Register (R1Ah) is set to the value 1 (we'll call it blue), and the Mask Polarity bit is 0 (set if equal to background), the above instruction will return a high bit in AL for each bit position in the raster containing a blue pixel. If the Mask Polarity bit is 1 (clear if equal to background), all bits in AL which correspond to blue pixels in the raster will be cleared, and bits corresponding to any color other than blue will be set.

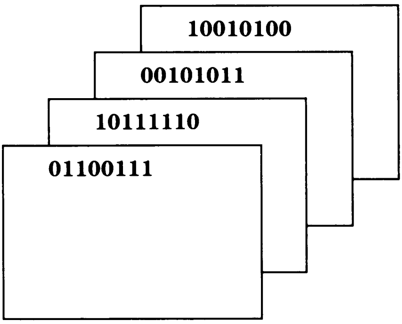
To put it another way, the BBM holds the truth values for a raster of eight pixels in answer to the query: "Are there pixels of the background color in this raster?"

The Don't Care Planes value (R19h, bits 0-3) will force a high bit from a particular memory plane, and so will not contribute to the truth value result returned in the BBM. The anatomy of a Read operation is:

```
        7  6  5  4  3  2  1  0
BBM     x  x  x  x  x  x  x  x
                             |
                     BBM[0] = {(Plane[0] = Back[0]) OR Don't Care[0]}
                              AND
                              {(Plane[1] = Back[1]) OR Don't Care[1]}
                              AND
                              {(Plane[2] = Back[2]) OR Don't Care[2]}
                              AND
                              {(Plane[3] = Back[3]) OR Don't Care[3]}
```

Note that whenever a read operation occurs, a 32-bit latch, called the **Source Latch**, is updated with the values of the four planes of memory. The contents of the Source Latch itself are internal to the InColor Card and cannot be directly accessed by the CPU.

4 Planes



32-Bit Latch

Bits/Pixels

7	6	5	4	3	2	1	0	
1	0	0	1	0	1	0	0	Plane 0
0	0	1	0	1	0	1	1	Plane 1
1	0	1	1	1	1	1	0	Plane 2
0	1	1	0	0	1	1	1	Plane 3

CPU Read

Every CPU read operation updates a 32-bit latch with the contents of the four planes of memory (8 bits per plane)

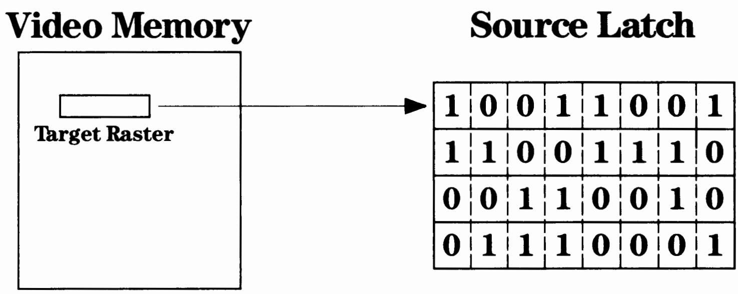
Source Latch

Write Operation and Write Modes

The efficiency of the InColor Card can be appreciated when the read operation settings are combined with the "Write Mode" settings; for example, when the CPU operation **OR ES:[DI],AL** reads the video memory, modifies it, and places the data back into video memory. The four write modes of the InColor Card each interpret the data byte from the CPU differently. Each write mode will modify video memory in one way in response to a high bit in the data byte, and another way in response to a low bit in the data byte. The write mode determines the source of the color value to be placed into memory. There are only three source locations to select from: the latch, the background value in the read/write color register, and the foreground value in the read/write color register.

		Bit in CPU Data = 0	Bit in CPU Data = 1
Write Mode 0	writes	background value	foreground value
Write Mode 1	"	source latch	foreground value
Write Mode 2	"	background value	source latch
Write Mode 3	"	—source latch	source latch

Remember that each CPU read operation updates the source latch with the color values for the raster being modified. The Hercules InColor Card is an efficient vehicle for color graphics because the write phase of the read-modify-write cycle can operate on all pixels in a raster independent of their color value.



R/W Color Register

1 0 0 1	0 0 0 1
Background Color	Foreground Color

CPU Data To Write

1 0 0 0 0 1 1 0

		0	1 In CPU Data
Write Mode 0	= Write	Back	Fore
Write Mode 1	= Write	Latch	Fore
Write Mode 2	= Write	Back	Latch
Write Mode 3	= Write	–Latch	Latch

Write Mode Determines Where Pixel Values Are Coming From When Writing to the Four Planes

Write Mode

Hardware Features

The operations covered below all involve instructions which are commonly found in monochrome bit-mapped graphics. On the InColor Card, the color control is placed in the I/O map rather than the bit-map so that existing monochrome graphics can be run unmodified, and adapted to color without changing the bit map manipulation.

As presented here, color manipulation is facilitated by setting up the Read/Write Color Register to the object (the item of interest) color value. For instance, to OR a magenta object onto an unknown background, set the background and foreground colors in register 1Ah to magenta. This is not essential for the XOR and NOT operations since in XOR and NOT, the Don't Care bit is set for all four planes.

For a logical OR of a register into a screen buffer location:

19h = 00h	Don't Care (bits 0 – 3)	= 0000 — care all planes
	Write Mode (bits 4 + 5)	= 01—write latch, fore
	Mask Polarity (bit 6)	= 0 — set if equal to background

For a logical AND of a register into a screen buffer location:

19h = 70h	Don't Care (bits 0 – 3)	= 0000 — care all planes
	Write Mode (bits 4 + 5)	= 10 — write back, latch
	Mask Polarity (bit 6)	= 1 — clear if equal to background

For a logical NOT of a register into a screen buffer location

19h = 7Fh	Don't Care (bits 0 – 3)	= 1111 — don't care all planes
	Write Mode (bits 4 + 5)	= 11 — write — latch, latch
	Mask Polarity (bit 6)	= 0—set if equal to background

For a logical XOR of a register into a screen buffer location:

19h = 3Fh	Don't Care (bits 0 – 3)	= 1111 — don't care all planes
	Write Mode (bits 4 + 5)	= 11 — write — latch, latch
	Mask Polarity (bit 6)	= 0 — set if equal to background

The material covered in this section is demonstrated in detailed examples found in Section 3, Appendix C, beginning on page 76.

Hardware Features

Power-On Conditions (monochrome compatibility)

During the power-on initialization process, the following registers are initialized to the default settings specified below. These values are the appropriate settings for compatibility with the monochrome text, graphics, and RamFont modes.

Exception Register (17h)—

Cursor Color	= 0000 (black)
Palette Enable/Disable	= 0 (disabled)
Attribute Selector	= 1 (normal)

Plane Mask Register (18h)—

Display Planes	= 1111 (display all planes)
Freeze Planes	= 0000 (freeze no planes)

Read/Write Control Register (R19h)—

Don't Care Planes	= 0000 (care all planes)
Pixel Write Mode	= 00 (write background/foreground)
Mask Polarity	= 1 (clear if equal to background)

Read/Write Color Register (R1Ah)—

Foreground Color	= 1111 (white)
Background Color	= 0000 (black)

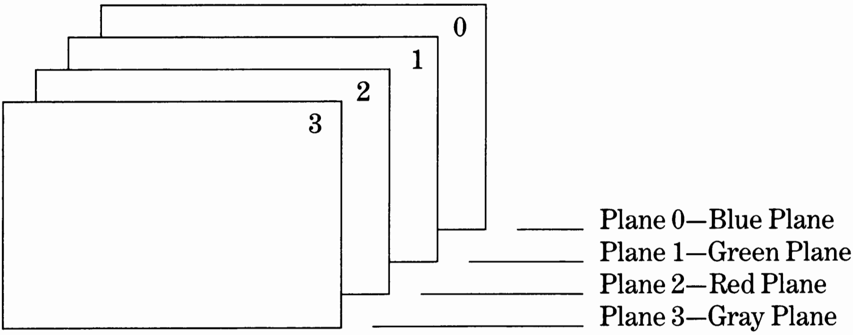
Source Latch Protect Register (R1Bh)—

Protect Pixels	= 00000000 (no bits are protected)

Hardware Features

Color Mapping

The Hercules InColor Card controls an enhanced RGB monitor to simultaneously display up to 16 out of a possible 64 colors. The following illustration shows the color mapping of the memory planes:



These are also the bit positions which specify a color whenever there is a 4-bit color code required. The least significant bit represents the blue plane, and the most significant bit represents the gray plane.

When the palette is disabled the colors resulting from the 16 permutations of 4 bits are:

Value	Color Code	Monitor Gun Value	Final Color
	YRGB	R G B R G B	
0	0 0 0 0	0 0 0 0 0 0	Black
1	0 0 0 1	0 0 0 0 0 1	Blue
2	0 0 1 0	0 0 0 0 1 0	Green
3	0 0 1 1	0 0 0 0 1 1	Cyan
4	0 1 0 0	0 0 0 1 0 0	Red
5	0 1 0 1	0 0 0 1 0 1	Magenta
6	0 1 1 0	0 0 0 1 1 0	Brown
7	0 1 1 1	0 0 0 1 1 1	White
8	1 0 0 0	1 1 1 0 0 0	Gray
9	1 0 0 1	1 1 1 0 0 1	Light Blue
10	1 0 1 0	1 1 1 0 1 0	Light Green
11	1 0 1 1	1 1 1 0 1 1	Light Cyan
12	1 1 0 0	1 1 1 1 0 0	Light Red
13	1 1 0 1	1 1 1 1 0 1	Light Magenta
14	1 1 1 0	1 1 1 1 1 0	Yellow
15	1 1 1 1	1 1 1 1 1 1	Intensified White

When the palette is enabled, the color code value is an index into the 16 register color palette which can be programmed with 6 bit color codes for the screen.

6845 RamFont Programming Considerations

The size of the characters that are displayed in RamFont mode is determined by two separate factors: the character definitions in the RamFont storage area and the values that have been programmed into the registers of the 6845 CRT controller. Sixteen bytes are allotted in the RamFont storage area for each character. The values with which the 6845 is programmed determine how many of these sixteen bytes are displayed when a character is mapped to the screen. If a sixteen-bit-high character set were stored in the RamFont storage area, but the 6845 were programmed with values appropriate for eight-bit-high characters, for instance, only the upper eight bits of each character would be displayed.

The registers that must be initialized for differing character sizes perform three distinct functions: horizontal timing, vertical timing and cursor shape. These registers are all WRITE ONLY registers. When a character size other than the 9 ×14 text-mode matrix is desired, the 6845 will need to be programmed with appropriate parameters.

Listed in the table below are 6845 timing values that are recommended when programming various character sizes.

Note: **When programming the 6845, you should always re-program all of the registers.** This is because inappropriate values in the 6845 can, in some cases, cause damage to your monitor. Thus it is far safer to make no assumptions about previous states of any of the 6845's registers; instead, program them all with the correct values for the mode you intend to use.

6845 Timing Parameters for RamFont Mode
Horizontal Timing Registers

Reg. No.	Character width (in pixels)	
	8	9
r0	109	97
r1	90	80
r2	92	82
r3	15	15

Vertical Timing Registers

Reg. No.	Character height (in pixels)												
	4	5	6	7	8	9	10	11	12	13	14	15	16
r4	92	74	61	52	45	40	36	32	29	27	25	23	22
r5	2	0	4	6	2	1	0	7	10	6	6	10	2
r6	88	70	58	50	43	38	35	31	29	26	25	23	21
r7	89	70	59	51	44	39	35	32	29	27	25	23	22

Section Three

Appendices

3

Appendix A—Troubleshooting

If you encounter any problems in using the Hercules InColor Card, please check the following procedures.

Problem	Try
No Image on Screen	Go back through the steps for installing the InColor Card (chapter 2, page 10). Check switch settings to be sure they are set for a monochrome display. You may want to refer to your computer owner's manual. Make sure you don't have a monochrome text card, Hercules Graphics Card, Hercules Graphic Card Plus or other non-compatible display card installed in your system (see Appendix B). Make sure you have the correct monitor. The Hercules InColor Card requires an enhanced RGB monitor that accepts sync frequencies of 60 Hz vertical and 22.0kHz horizontal. The Hercules InColor Card cannot be used with monochrome monitors or standard RGB or composite color monitors.
Software doesn't work with the card	Make sure you have the Hercules version of the graphics program you want to run. Check the installation procedure for the software you are using. Many graphics programs require the user to perform a one-time procedure to tell the program what video board you are using. Make sure you are in the FULL configuration before attempting to run Lotus 1-2-3, version 1A. If you are using one of the drivers included with the Hercules InColor Card, make sure that the version of the program you're using matches the driver and that you've followed the instructions correctly.

Make sure your Hercules InColor Card is not in FULL mode if you also have an IBM Color Graphics Adapter (or certain other graphics adapters) installed in your system. You can never run a graphics program (including 1-2-3, Symphony, and AutoCad) that uses page one of graphics memory when such an adapter is also installed. You can use the FULL configuration if the other card in your system is a Hercules Color Card.

Printer not working/ working improperly	Be sure you don't have any printer port conflicts (see Appendix B). The ability to print alternate fonts is controlled by software, not the Hercules InColor Card. To print alternate typestyles, you must make sure your software is capable of this task.

Appendix B—Compatibility with Other Systems Components

Computer/Software Compatibility

The Hercules Software and Hardware Compatibility Guide (included with your card) lists those computers and software products known to work with Hercules graphics cards, including the InColor Card.

If you have any questions about compatibility with your computer, consult this guide before installing the InColor Card in your system.

The Hercules InColor Card will run any text-mode software that runs on the IBM Monochrome Card.

The Hercules InColor Card will run graphics software specially written or modified to support the Hercules Graphics Card, Graphics Card Plus, or InColor Card. It will not run graphics software written to run on the IBM Color Card. To run such software with the InColor Graphics Card installed in your system, you'll need to add a Hercules Color Card.

Although the Compatibility Guide is updated frequently, it may not contain every Hercules-compatibile program that is currently available. Before purchasing a program not listed in the guide, check the program's documentation or consult the program's publisher for compatibility information.

Parallel Port Compatibility

If you have another parallel printer port installed in your computer as LPT1 (occupying addresses 03BC-03BE), you will need to disable either the port on the Hercules InColor Card or the other port in your system. See your system owner's manual for information on any parallel ports installed in your system.

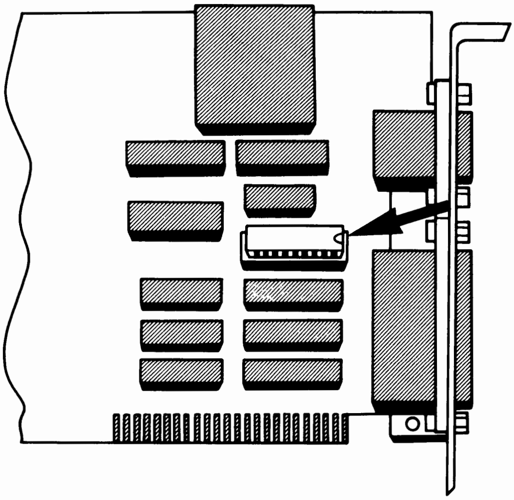
If the existing printer port has a lower priority address (0378 or 0278) it will now be designated as LPT2, and no conflict with the parallel port on the InColor Card will exist. If you have just one printer, connect it to the InColor Card's port.

Note: When the Hercules InColor Card is installed in a system with a Hercules Color Card, the parallel port on the InColor Card will be designated as LPT1 and that on the Color Card will be designated LPT2. If you have a single printer, connect it to the InColor Card's port.

Compatibility with Other Systems Components

To Disable the InColor Card's Parallel Port

1. Use a chip remover or small flathead screwdriver to carefully lift off the socketed chip HCT LPT112 (located as shown below).



2. Carefully store the chip on the piece of conductive foam (included with your InColor Card). You may re-install it later if you wish to re-enable the printer port.

Display Card Compatibility

Co-existence with the Hercules Color Card

The Hercules Color Card (Model GB200) is the only video card we know of that can co-exist under all circumstances in the same system with a Hercules InColor Card.

When Installing the InColor Card

Be sure to set your computer's configuration switch for a monochrome display. The InColor Graphics Card will always be the active display board on power-up.

Compatibility with Other Systems Components

Display Card Compatibility

Co-existence with the Hercules Color Card

The Hercules Color Card (Model GB200) is the only video card we know of that can co-exist under all circumstances in the same system with a Hercules InColor Card.

When Installing the InColor Card

Be sure to set your computer's configuration switch for a monochrome display. The InColor Card will always be the active display board on power-up.

To Access a Hercules Color Card Installed with the InColor Card

First, be sure that the InColor Card is not in FULL configuration. Then:

On a Dual-Floppy Disk System, place your DOS disk in drive A and type:

A> MODE CO80 [ENTER]

On a Hard-Disk System, make sure DOS is in your current directory and type:

C> MODE CO80 [ENTER]

To Return to the InColor Card

On a Dual-Floppy Disk System, place your DOS disk in drive A and type:

A> MODE MONO [ENTER]

On a Hard-Disk System, make sure DOS is in your current directory and type:

C> MODE MONO [ENTER]

Co-existence with Other Graphics Cards

The Hercules InColor Card can co-exist under some circumstances in your system with a limited number of other video cards. Be aware that the presence of any video board other than a Hercules Color Card will limit the operation of the InColor Card substantially.

Compatibility with Other Systems Components

The InColor Card can co-exist with an IBM Color Graphics Adapter, or any other video card which uses the same display memory addresses (B8000h-BFFFFh), provided that the Hercules InColor Card is used only in its DIAG or HALF configurations. The InColor Card may never be configured for FULL graphics while a non-Hercules Color Card is in your system.

Most popular graphics applications using the InColor Card **require** it to operate in the FULL configuration. Therefore, it will be impractical to keep an IBM Color Graphics Adapter in the system if you use such applications.

The Hercules InColor Card cannot co-exist with any graphics card that uses any part of the address space B0000h-B7FFFh. If you don't know which display memory addresses your graphics card uses, consult its owner's manual.

The IBM Monochrome Card, the Hercules Graphics Card and the Hercules Graphics Card Plus cannot stay in the same system with an InColor Card.

(See the Software Manual for Details on these configurations.)

Appendix C—Read/Write Operation Examples

Programmers already familiar with plane-organized color graphics should have no difficulty understanding the implementation of the Hercules InColor Card. For those who are new to color graphics, the following examples are offered. These provide detailed descriptions of several typical operations that are useful to understand in order to successfully write color software for the InColor Card. It is assumed that the reader is conversant with assembly language graphics programming.

Generally, there is more than one way to achieve a particular result on the InColor Card. For example, to draw a red line, you could freeze all but the red plane, then draw to it. You could draw a white line, then erase it from all but the red planes. Or, you could draw a green line, then map green to red through the palette. Each of these techniques might be a valid approach for a particular context. There are some techniques that are recommended because they represent an efficient and general approach. A few of these are presented in the following examples.
The examples are conceived from a particular orientation; that of accommodating existing Hercules monochrome graphics code on the InColor Card in full color.

Example 1: Plotting a dot of a desired color on a background of unknown color

On the Hercules Graphics Card (monochrome), the 8088 instruction "OR ES:[DI],AL" combines the contents of AL(mask) with the contents of the byte of data addressed by ES:[DI] (target), which is assumed to point to video memory. The OR instruction yields a high bit in the result for any high bit in either the target byte or the mask. This is perfectly good monochrome code.

Read/Write Operation Examples

On the InColor Card, we need to do more to manipulate colors. For example, to place a magenta dot on the screen, while leaving other colors in the raster undisturbed, we'll have to make sure the following settings are in effect:

```
Background = Magenta     ; The color we are interested in displaying,
Foreground = Magenta     ; these values are held in the Read/Write
                         ; Color Register.
Mask Polarity = 0        ; Bit in BBM set if pixel in memory is
                         ; magenta.
Don't Care = 0           ; We care about the contents of all planes.
Freeze = 0               ; All planes writable.
Latch Protect = 0        ; No pixels protected against modification.
Write Mode= 1            ; In the write phase, a low bit will update
                         ; the corresponding pixel in the video
                         ; buffer with the color value held in the
                         ; source latch; a high bit will set the
                         ; corresponding pixel to the foreground
                         ; color value.
```

Here's what happens when the CPU executes the OR instruction:

The first phase of the OR instruction reads the contents of video memory pointed to by ES:[DI]. This read updates the source latch with the contents of the video buffer so that the four bit color values for each of the eight pixel positions in the raster are held in the latch. Because the mask polarity bit is cleared, the data returned to the CPU (the Background Bit Mask) contains a high bit at the position of any magenta pixel in the byte, and a low bit for a pixel of any other color.

We have asked for a truth value for eight pixels: "Are there any magenta pixels in this byte?" The **BBM is the answer** to this query. Magenta pixels are high, all others are low. The CPU OR operation now modifies the result of the read operation with the contents of AL (our magenta mask). High bits in the mask produce high bits in the data written back to the video buffer.

In the write phase of the CPU OR operation, the new data is written back into video memory. Here, the write mode setting becomes important. Write mode 1 (write latch, fore) updates the four memory planes with data from one of two sources: 1) if the bit in the data byte is low, the buffer is updated with the contents of the source latch, or 2) if the bit in the data byte is high, the buffer is updated with the value held in the foreground color portion of the Read/Write Color Register. This causes the desired effect, since the source latch has been updated in the read phase of the OR instruction to contain the color values for the byte of video memory being modified, and we have initialized the foregound color to magenta. At the end of the single CPU instruction, we have successfully updated the video memory to achieve the desired result.

Example 2: "ANDing" a mask onto an unknown background.

In monochrome graphics, the logical operation AND is often employed to place a bit pattern onto a reverse video field. In such cases, the mask is complemented; that is, the object of interest is defined by low bits, the field on which the object is to be placed is defined by high bits.

To AND a green mask onto a field of unknown color, the following settings should be in effect:

```
Background = green      ; The color we are interested in, these
Foreground = green      ; values are held in the Read/Write Color
                        ; Register.
Mask Polarity = 1       ; Bit in BBM; clear if pixel in memory is
                        ; green.
Don't Care = 0          ; We care about the contents of all planes.
Freeze = 0              ; All planes are writable.
Latch Protect = 0       ; No pixels protected against modification.
Write Mode = 2          ; In the write phase, a low bit will update
                        ; the corresponding pixel in the video
                        ; buffer with the color value held in the
                        ; background color value. A high bit will
                        ; set the corresponding pixel to the value in
                        ; the source latch.
```

The CPU AL register contains our complemented mask.

Here's what happens when the CPU executes the instruction "AND ES:[DI],AL":

The first phase of the AND instruction reads the contents of video memory pointed to by ES:[DI]. This read updates the source latch with the contents of the video buffer so that the four bit color values for each of the eight pixel positions in the raster are held in the latch. Because the mask polarity bit is set, the data returned to the CPU (the Background Bit Mask) contains a low bit at the position of any green pixel in the byte, and a high bit for a pixel of any other color.

The AND operation now modifies the result of the read operation with the contents of AL (our complemented green mask). High bits in the mask produce high bits in the data to be written back to the video buffer. Low bits in the mask force low bits in the data, representing the positions of pixels we desire to make green.

In the write phase of the AND operation, the new data is written back into video memory. Here, again, the write mode setting becomes important. Write mode 2 (write back, latch) updates the four memory planes with data from one of two sources:

1) if the bit in the data byte is low, the buffer is updated with the background color value (which has been set to green), or 2) if the bit in the data byte is high, the buffer is updated with the contents of the source latch. The source latch has been updated in the read phase of the AND instruction, so that it now contains the color values for the byte in video memory we are modifying. Again, with a single CPU instruction, we have successfully updated video memory to achieve the desired result.

Example 3: "XORing" to place and erase.

The XOR operation turns a high bit in either the mask or the background bit mask (BBM) into a high bit in the target; it will yield a low bit in the target if both the mask and the target contain a high bit. XORing a mask with a byte of high bits complements all bits in the mask.

In monochrome graphics, the logical operator XOR is often used to place and erase a dot or a mask. Monochrome makes this easy since the complement of a pixel that is on is a pixel that is off, and vice-versa.

The XOR operator is useful in color as well, but this usefulness is complicated by the fact that the complement of one color is another color, and not necessarily off (or background color). Color complements on the InColor Card are not absolute; instead, they are determined by the currently selected **color palette.** For example, the complement of the color value 0001 (which might be blue) is 1110 (which could be yellow).

Read/Write Operation Examples

A graphics cursor is an example of an item which might be desirable to manage with the XOR logical placement style. A cursor moves often, and it has to be removed from former positions as it does so. XORing a cursor mask into video memory changes the color value in a predictable fashion. XORing the mask again at the same memory location complements the colors a second time, restoring them to their former values and making the cursor disappear.

To place or erase a mask with the instruction "XOR ES:[DI],AL", our mask should be in AL and the following settings should be in effect:

```
Background = n/a         ; Background and foreground have no
Foreground = n/a         ; meaning with write mode 3.
Mask Polarity = 0        ; Bit in BBM set if true (i.e. equal to
                         ; background).
Don't Care = 15          ; We don't care about color values at all.
                         ; We just want to complement some and
                         ; not others.
Freeze = 0               ; All planes writable.
Latch Protect = 0        ; No pixels protected against modification.
Write Mode = 3           ; In the write phase, a low bit will comple-
                         ; ment the color value of the corresponding
                         ; pixel, while a high bit will set the
                         ; corresponding pixel to the value in the
                         ; source latch.
```

The first phase of the XOR instruction will read the contents of the video memory pointed to by ES:[DI]. In this instance, the Don't Care Planes value is set to indicate that we don't care about the value of any of the planes. The Read operation still updates the source latch with the contents of the four memory planes, but the BBM yields high bits for all pixel positions—a don't care on all planes forces all true.

The XOR operation now modifies the result of the read operation with the contents of AL (our mask). High bits in the mask will force low bits in the data to be written back to the video buffer. Low bits in the mask produce high bits in the data.

In the write phase of the XOR operation, the new data is written back into video memory. Write mode 3 (write-latch, latch) updates the four memory planes with the complement of the source latch if the bit in the data byte is low; if the bit in the data byte is high, the buffer is updated with the contents of the source latch. This accomplishes the desired effect, changing the color values in those pixel positions defined by the mask. White pixels (1111), for instance, become black (0000); red pixels (0100) become light cyan (1011). Performing the same operation at the same location again complements the color values of the pixel positions in the mask, restoring them to their former values and, in this instance, removing the cursor.

A useful effect can be achieved through the palette when you are willing to limit the number of displayable colors. If you have only eight displayable colors and a cursor color which must be maintained, map the complement of each of the palette indices of your displayed colors to the cursor color. For instance, if you want to have eight colors and a green cursor, program palette values 0-7 with your colors and program palette indices 8-15 with green. The complement of each palette color becomes the cursor color.

Example 4: "NOTing" to complement color

The NOT operator functions just like the XOR operator detailed above, except that no mask is used to modify data. NOT simply complements all the bits, then writes the new data back into video memory. NOT would be used in monochrome graphics to set a screen into reverse video or back again. In color graphics, the NOT operator is useful with the palette if the desired color value in the palette is stored in the palette index as the complement of the color value to be NOTed.

Read/Write Operation Examples

To complement the color of a raster with the instruction "NOT ES:[DI]," make sure the following settings are in effect:

```
Background = n/a        ; Background and foreground don't have
Foreground = n/a        ; meaning in write mode 3.
Mask Polarity = 0       ; Bit in BBM set if true.
Don't Care = 15         ; We don't care about color values at all.
                        ; We just want to complement some and
                        ; not others.
Freeze = 0              ; All planes writable.
Latch Protect = 0       ; No pixels protected against modification.
Write Mode = 3          ; In the write phase, a low bit will comple-
                        ; ment the color value of the corresponding
                        ; pixel, while a high bit will set the
                        ; corresponding pixel to the value in the
                        ; source latch.
```

The first phase of the NOT instruction reads the contents of video memory pointed to by ES:[DI]. The Don't Care Planes value is set to indicate that we don't care about the value of any of the planes. The Read operation still updates the source latch with the contents of the four memory planes, but because the mask polarity bit is 0, the BBM will contain high bits for all pixel positions.

The CPU now has a byte of high bits and the NOT operation complements them all, yielding a byte of low bits to be written back to video memory. Write mode 3 (write-latch, latch) updates the four memory planes with the complement of the source latch if the bit in the data byte is low, and all bits in this case are low. This has accomplished the desired effect, changing all color values to their complement. NOTing a second time restores the colors to their original values.

Read/Write Operation Examples

Example 5: Clearing the screen to a desired color.

This common operation can be accomplished in a number of ways with the InColor Card.

In this example, we are going to clear the screen to black, without concern for what was there before this operation. Since we just want to replace the contents of memory, not combine new data with what is already there, we will use a CPU MOV instruction. In this instance, the source latch is not updated because the buffer will not be read; instead, new data is simply written.

The settings needed to accomplish this are:

```
Background = black        ; The color to write.
Foreground = n/a          ; This value will not be used.
Write Mode = 0            ; Write back, fore.
Freeze = 0                ; All planes writable.
```

If CPU register AL contains the value zero, the instruction "MOV ES:[DI],AL" writes all low bits to the raster, setting them all to the background color. As is typical, clearing the buffer is accomplished with the REP operator and a string move instruction like STOSW. Note that though a 16-bit operation is suitable to use when simply transferring data to video memory, it must be understood that any operation involving a read of video memory can only be performed 8 bits at a time. The source latch holds 8 pixel values (32 bits) only.

Example 6: Copying an object using latch protect

The Latch Protect register is very useful when an object on a random background (i.e. a background whose present color value we are not concerned about) is copied to another area of the screen. The Source Latch Protect Register (R1Bh) forms a mask which will protect the values of the pixels in the latch. These positions are not updated by subsequent reads, but continue to hold the value present when the latch was protected.

Read/Write Operation Examples

If we have a multi-colored object on a green field, and we want to move it to another area of the screen, we can perform this task efficiently with the following:

Since we know that the object is on a green background, we can differentiate this area of interest by setting the mask polarity bit to 1 (clear if equal to background), then setting the background color to green, and reading the byte with"MOV AL,ES:[DI]." AL now contains a high bit for each pixel in the byte that is not green, and can now be used to protect the latch, since we want to preserve the color values of each high bit position in AL. The latch is protected simply by sending this value to register 1Bh.

Next, the byte of data that represents the target raster should be read. Nothing is done with the result obtained from the read; the purpose of this read is solely to update the latch (update pixel positions that have not been protected). Now, the latch contains values currently in the target byte, except in the positions protected. These still contain the multi-colored object. Modify the target raster (using write mode 2) by writing all high bits ("MOV ES:[DI],AL", assuming AL contains OFFh). This completes the otherwise complicated task of moving a multi-colored object, in just four simple steps:

1. Read source
2. Protect object
3. Read target to update latch
4. Modify target

A fifth, crucial step remains—unprotect the latch by sending a value of zero to register 1Bh. Forgetting this step will make the rest of your program one very large, undesirable side effect.

Appendix D—Programming Do's and Don'ts

Graphics Mode

- The easiest way to generate color is by using the Plane Mask Register (18h). This is accomplished by moving the desired bits to mask out planes into the high nibble and displaying all planes. A more elegant and versatile way is to use the Read/Write Color Register (1Ah) and Read/Write Control Register (19h) to generate color in graphics mode.
- When you are expecting the latch to be updated in video RAM, **do not** use 16-bit CPU operations. The CPU will do two reads followed by two successive writes. The contents of the latch will be data from the second read. Use only 8-bit operations when writing to or from the latch.
- Do not use string move instructions that transfer data into video memory if you require that the latch be updated with the target byte.
- When writing an object onto an unknown background, set the background and foreground colors to the object color value in the Read/Write Color Register (1Ah). This is useful when using logical operators to update color information in video RAM.
- When programming the Palette Register (1Ch), use a read operation to reset the pointer to the first position in the palette table. Then use successive writes to update the palette. The palette pointer is auto-incremented after each write. To enable the palette, set bit 4 of the Exception Register (17h).
- Be sure to restore the Source Latch Protect Register (1Bh) to its default setting after each use.
- To effectively XOR a cursor in graphics mode when moving through different color backgrounds, set the complement of all displayable colors to the color of the cursor in the palette. For example, if the cursor is green, set palette colors 8-15 to green.
- Be sure to reset all proprietary registers to default values when exiting your application.

Writing Unified Graphics Drivers

- Before you start, **think monochrome** (pixels on/off). This will ensure a unified driver when adding color routines to your existing programs.

Programming Do's and Dont's

Color RamFont Mode

- The Read/Write Color Register (1Ah) allows fonts to be stored in color in video RAM. This method allows software to access the normal attribute set. However, these fonts are stored to specific planes so as to be predefined in color and can only be changed by programming the palette.
- There are numerous ways to generate color in RamFont mode. Bit 5 of the Exception Register (17h) selects between normal attribute (bit 5 set) and alternate attribute set (bit 5 clear). The alternate attribute set interprets the attribute byte as color information.
- To store up to 48 fonts, use the Plane Mask Register (18h) to write individual planes. This is accomplished by freezing three of the four planes, then writing to video RAM. Each plane can store a maximum of 48k RamFont information.
- In 48k RamFont Mode with the individual planes storing unique character data, use the palette to select colors for each of the four planes. A maximum of four colors can be displayed when using this particular configuration.
- If you are writing a RamFont application on the InColor Card that you also want to run on the Hercules Graphics Card Plus, remember that the monochrome card has only one plane of video memory. Therefore, storing fonts into separate planes on the InColor Card will have the effect of overwriting font data on the monochrome card.
- Set all proprietary registers to default values when exiting your application.

Appendix E—Display Modes and Attributes

	Normal Attributes	Alternate Attributes
Text, 4k RamFont	Normal, underline, high-intensity, reverse, blink.	Attribute selector bit = high four bits of attribute byte hold a value specifying a background color. Low four bits of attribute byte hold a value specifying foreground color.
48k RamFont	If blink enable=0 normal, underline, overstrike, boldface, reverse. If blink enable=1 normal, underline, overstrike, high-density, blink.	Attribute bits form a mask of the four planes of memory. High bit disables plane, low bit enables plane.

Appendix F—Specifications

Resolution:

- Text Mode Resolution: 80 columns × 25 rows, using 9 × by 14 character matrix.
- Graphics Mode Resolution: 720 horizontal by 348 vertical.
- RamFont Mode Resolution: 80 or 90 columns by 21 to 87 rows, using character matrices ranging from 8 or 9 pixels wide by 4 to 16 pixels high.

Color Display:

- Color display up to 16 colors selected from a palette of 64.

Display Memory:

- 4 planes of 64k dynamic RAM multi-formatted display memory on board.
- May be used for color display or expanded RamFont display, or a combination of color and expanded RamFont.
- Ramfont Storage Capacity:

  4k RamFont Mode—One software defined font of 256 characters.

  48k RamFont Mode—3,072 software defined elements or characters, equivalent to 12 standard 256 character sets.

  Expanded RamFont Mode—12,288 software defined elements or characters, equivalent to 48 standard 256 character sets.

Specifications

Display Memory Format:

- Text Mode: 4k display buffer from B0000 to B0FFF.
- Graphics Mode: Two 32k graphics pages. Page 0 from B0000 to B7FFF. Page 1 from B8000 to BFFFF.
- RamFont Mode: 16k display buffer from B0000 to B3FFF. 4 planes of up to 48k character definition storage from B4000 to BFFFF.

Parallel Printer Port:

- Address: 03BC-03BE
- Logical Device Designation: LPT1

Attributes:

- Standard Text Attributes: Blink, reverse video, underline, high-intensity, blank and normal.
- New Text Attributes in 48k RamFont mode: Boldface, overstrike.

Sync Frequencies:

- 22.0kHz horizontal, 60Hz vertical

Current Consumption:

- 1 Amp @ 5 V

Worst Case Memory Access Time:

- 500 ns

Dimensions:

- 8.7″ long (including bracket), 4.2″ high

Appendix G—Port Pinouts

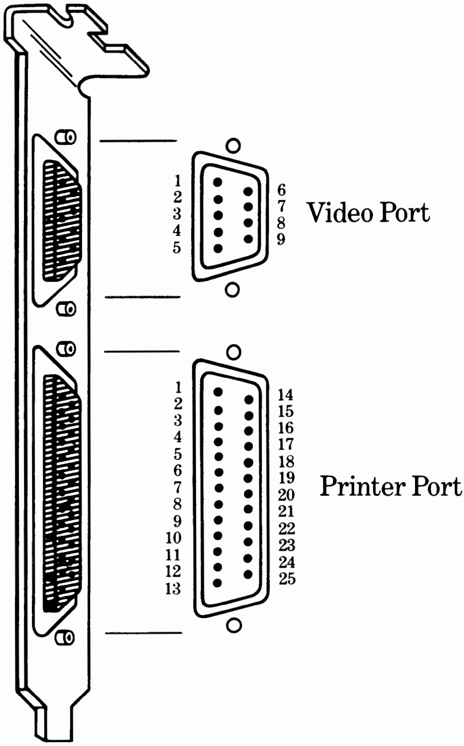
Port Pinouts

VIDEO PORT
(at standard TTL levels)

1 GROUND
2 SECONDARY RED
3 PRIMARY RED
4 PRIMARY GREEN
5 PRIMARY BLUE
6 SECONDARY GREEN
7 SECONDARY BLUE
8 + HORIZONTAL SYNC
9 – VERTICAL SYNC

PRINTER PORT
(at standard TTL levels)

1 – STROBE
2 + DATA BIT 0
3 + DATA BIT 1
4 + DATA BIT 2
5 + DATA BIT 3
6 + DATA BIT 4
7 + DATA BIT 5
8 + DATA BIT 6
9 + DATA BIT 7
10 – ACKNOWLEDGE
11 + BUSY
12 + P.END (out of paper)
13 + SELECT
14 – AUTO FEED
15 – ERROR
16 – INITIALIZE PRINTER
17 – SELECT INPUT
18-25 GROUND



Index

Index

Limited Warranty

Hercules Computer Technology (HCT) warrants this Hercules InColor Card to be in good working order for a period of two years from the date of purchase from HCT or an authorized InColor Card dealer. Should this InColor Card fail to be in good working order at any time during the two year period, HCT will, at its option, repair or replace it at no additional charge, except as set out below. Replacement parts will be either reconditioned or new, and the replaced parts will become the property of HCT. This limited warranty does not cover Graphics Cards damaged from accident, disaster, misuse, abuse or unauthorized modifications.

Limited Warranty service may be obtained by delivering this product to an authorized Hercules InColor Card dealer along with Proof of Purchase Date. If the product is mailed to HCT, you must obtain an RMA number from HCT, send it freight prepaid and you must insure the product or assume the risk of loss or damage in transit and you must return it in its original shipping container, or an equivalent.

All express and implied warranties for the Hercules InColor Card, including warranties of merchantibility and fitness for a particular purpose, are limited to two years from the date of purchase. No warranties, whether express or implied, will apply after this period. Some states do not allow limitations on how long an implied warranty lasts, so these limitations may not apply to you.

If this InColor Card is not in good working order, your sole remedy shall be repair or replacement as described above. In no event will HCT be liable to you for damage arising out of the use or inability to use this product.

Some states do not allow the exclusion or limitation of incidental or consequential damages for consumer products, so these limitations may not apply to you.

Hercules Computer Technology

2550 Ninth Street

Berkeley, CA 94710

Notice to Users

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

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

NOTE: The I/O cables that interconnect between this device and any peripheral (such as a monitor, printer, etc.) must be shielded in order for this equipment to meet Class B emission limits.

Some shielded I/O cables that have plastic connectors may have a "pig-tail" lead with a lug attached. A screw on a metal chassis of the peripheral should be located to fasten this lug to. Connecting this "pig-tail" lead to the metal chassis of the peripheral will generally reduce RF emissions that could cause interference.

Move the computing device away from the receiver being interfered with.

Relocate (turn) the computing device with respect to the receiver.

Reorient the receiver's antenna.

Plug the computing device into a different AC outlet so that the computing device and receiver are on different branch circuits.

Disconnect and remove any I/O cables that are not being used. (Unterminated I/O cables are a potential source of high RF emission levels.)

Unplug and remove any circuit boards that are not being used. (Unterminated cards can also be a source of potential interference.)

Be certain that the computing device is plugged into grounded outlet receptacles. (Avoid using AC cheater plugs. Lifting the power cord ground may increase RF emission levels and may also present a lethal shock hazard to the user.)

If you need additional help, consult your dealer or ask for assistance from the manufacturer. Customer support information is found in the "Troubleshooting" section of this manual. You may also find the following book helpful: **How to Identify and Resolve Radio-TV Interference Problems.** This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402—Stock No. 004-000-000345-4.

Hercules Computer Technology

2550 Ninth Street
Berkeley, California 94710
Telephone: 415 540-6000
Telex: 754063 HERCULES UD