The IBM Color/Graphics Adapter

A PROGRAMMER'S GUIDE TO THE COLOR DISPLAY ADAPTER: FILLING IN THE INFORMATION GAP LEFT BY THE AVAILABLE IBM LITERATURE

THOMAS V. HOFFMANN

"I can't come back, I don't know how it works!"—The Wizard of Oz

There's no doubt the Wizard of Oz was clever. To reward them for their watery withering of the Wicked Witch of the West, he concocted courage for the Lion, bestowed brains on the Scarecrow, and gave a hearty handout to the Tin Woodsman. But when he tried to deliver Dorothy to Kansas in his used balloon, he was helplessly carried away, and the old gasbag

was never seen again. His final exclamation, reproduced above, illustrates that cleverness is more ingenuity than knowledge, and of considerably less use in dealing with high technology.

So while it's interesting to read on page 2-46 of the Technical Reference Manual that the IBM Color/Graphics Monitor Adapter is "highly programmable," and that "many additional modes are possible with clever programming," it's not very helpful. And knowing that 16 color 160 by 100 low resolution graphics is "not supported in ROM" and "requires special programming" is tantalizing, but hardly informative. Is clever programming more difficult than special programming? If we knew how the adapter works, perhaps ordinary programming would suffice.

This article is a programmer's guide to the color display adapter, filling in much of the information gap left by the available IBM litera-

ture. The additional information comes from various sources: logic diagrams and program listings in the Technical Reference Manual, component data sheets, experiments, and the display card itself.

We will begin with a brief overview of the color adapter hardware organization, then review the standard alpha and graphics modes available from BASIC. Next there is a detailed description of the programmable features of the adapter, including the Motorola 6845 CRT controller. Along the way we present two different techniques for 16 color graphics programming, which IBM only hints about: the one used in the Microsoft Flight Simulator that only works with composite monitors or TV sets, and another low resolution technique that works on any color display.

Hardware Overview

The Color Graphics Monitor Adapter is a single printed circuit card, which fits into one of the expansion bus slots in the PC motherboard. IBM recommends that it always be placed in slot number two. Since the color card is deeper than other cards, there is the possibility that pressure on the top of the case could damage the card or components under it if it were in slot three. The Monochrome Display Adapter should go in slot three if your system has both adapters.

> *WHILE IT'S INTERESTING THAT THE IBM COLOR/GRAPHICS MONITOR ADAPTER IS "HIGHLY PROGRAMMABLE," AND THAT "MANY ADDITIONAL MODES ARE POSSIBLE WITH CLEVER PROGRAMMING," IT'S NOT VERY HELPFUL.*

The major elements of the color adapter are the Motorola 6845 CRT controller chip, a 16K byte display buffer memory, a ROM character generator, and mode, color, light pen control, and status registers. Figure 1 is a block diagram which shows the major data paths connecting these elements.

The 6845 CRT controller is the heart of the adapter. It provides the basic horizontal and vertical video timing signals and generates the addresses for accessing the display buffer and character generator.

The display buffer RAM resides in the 8088 CPU address space at segment &HB8OO, and can be accessed by both the CPU and the adapter's video generation logic. [Note: Hexadecimal numbers in the text are preceded by &H, the standard BASIC notation.] Data stored in the buffer by the CPU are read out two bytes at a time into two 8-bit data latches. From there, the data passes to the serializer logic (going through the character generator in alpha mode), which extracts one picture element, or pixel, at a time, and then to the color encoder where it is turned into RGB or composite video information for output to the display.

The character generator is an 8K byte read-only memory, which contains the patterns used to generate dot matrix characters in alpha modes. This ROM actually contains three sets of patterns, each defining 256 characters, and is the same chip used in the Monochrome Display Adapter. On the color card only 2K bytes are used, eight bytes per character, with each of the eight bits in a byte representing a dot in the 8x8 character cell. This ROM cannot be read by the CPU, so there is a copy of the first 128 character patterns in BIOS ROM, which is used in graphics modes to generate characters under software control.

There is a minor mystery surrounding the character generator. The Technical Reference Manual mentions a jumper that can select either a 5x7 single dot font or 7x7 character double dot font. The terms single and double dot usually refer to the width of each displayed pixel; characters look better on low resolution displays if the dots are wider. My adapter has no jumper, and the logic diagrams don't show one either, but the card generates 7x7 characters, which is consistent with no jumper being installed. It's not clear what IBM means by single and double dot fonts.

The mode and color control registers determine the operating mode and overall color attributes, and are described in detail in the programming section below.

Theory of Operation

The adapter has two major operating modes, alphanumeric and graphics. In both modes, information stored in the display buffer is continuously read out, interpreted, and displayed. The display image is composed of 200 lines of 320 or 640 dots called picture elements or pixels. The difference between alpha and graphics mode is in the interpretation of the stored data.

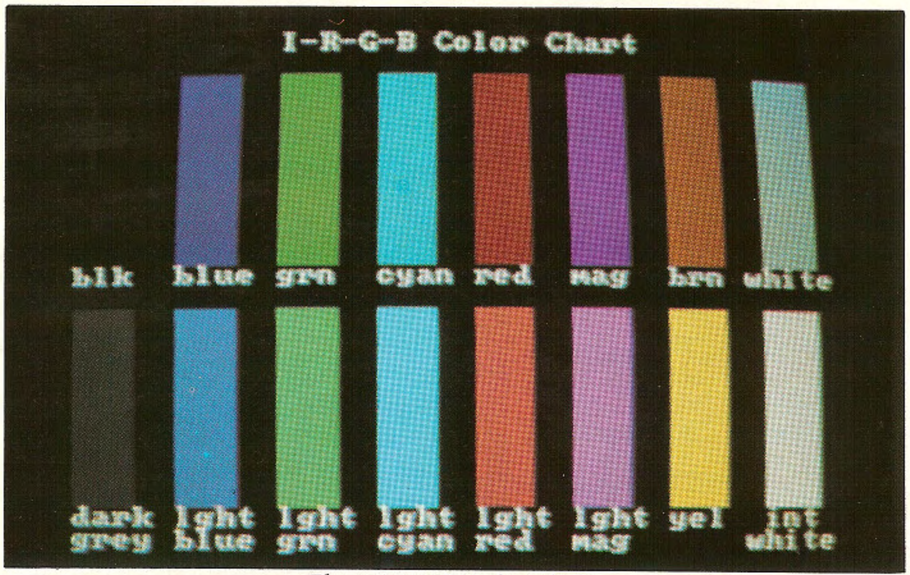


Photo 1: IRGB Color Chart

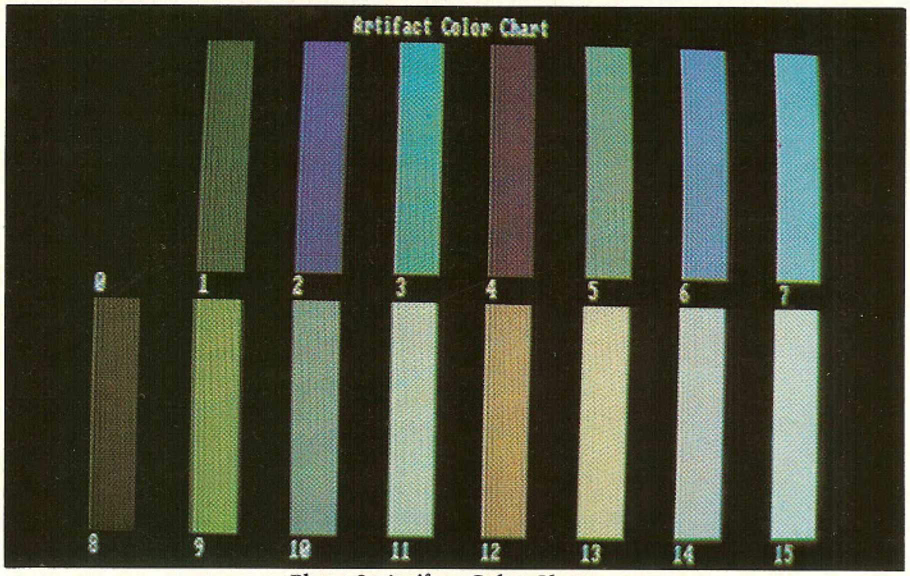


Photo 2: Artifact Color Chart

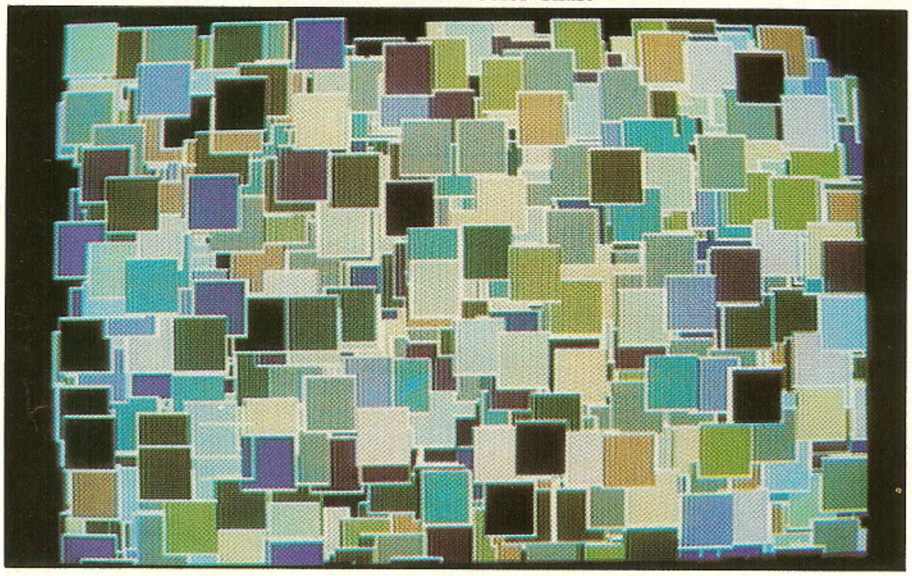
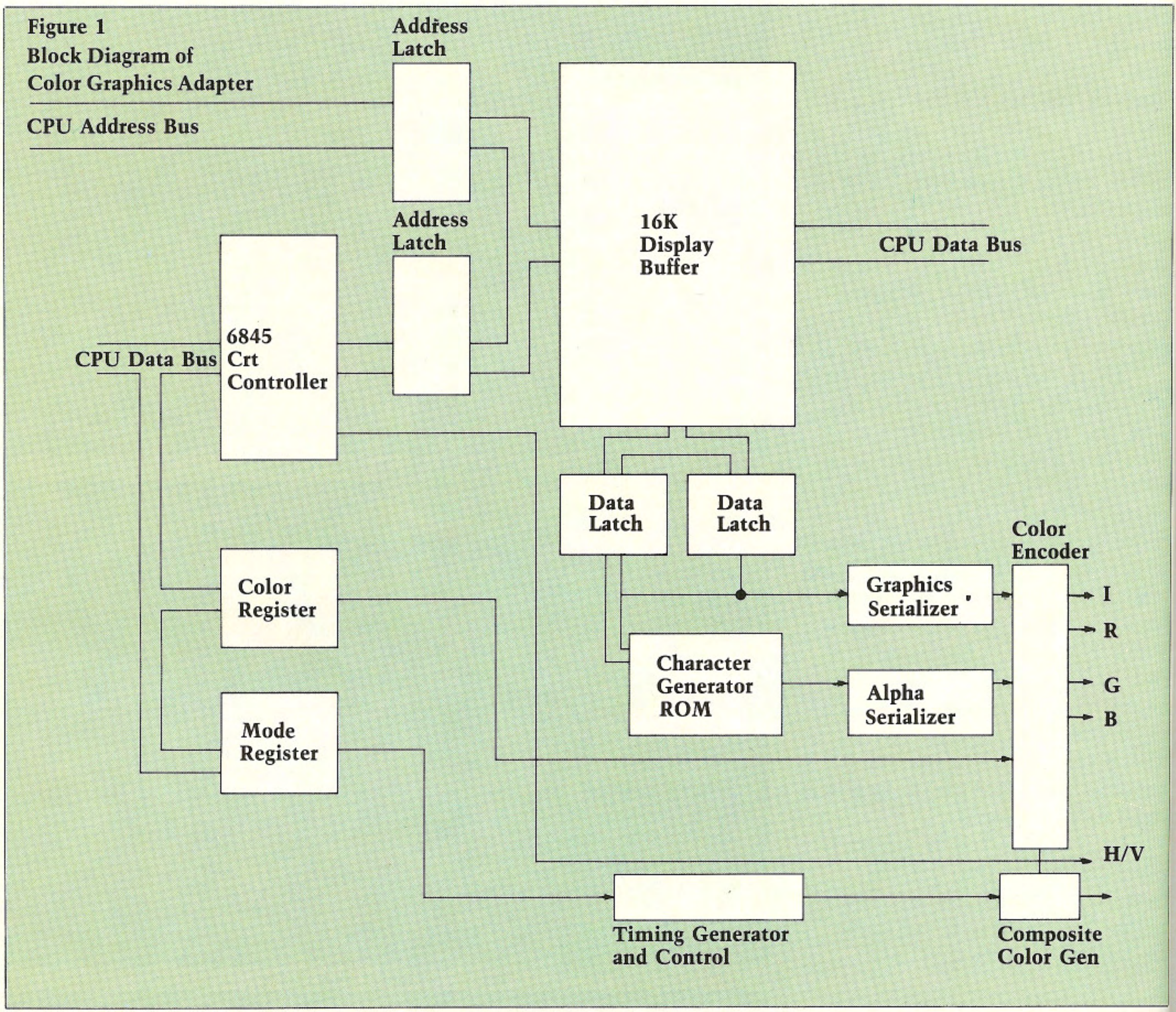


Photo 3: BOXES

Figure 1
Block Diagram of
Color Graphics Adapter



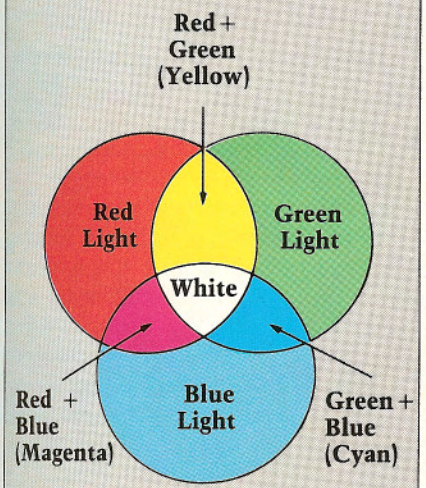
THE COLORS ARE DETERMINED BY INDEPENDENT BITS FOR EACH ADDITIVE PRIMARY COLOR—RED, GREEN, AND BLUE AND A FOURTH BIT FOR INTENSITY.

Alpha Mode

In alpha mode, each pair of bytes determines an 8x8 pixel region of the display. The first byte, which always has an even address, contains a character code that selects one of 256 patterns from the character generator ROM. The second byte called the attribute byte, determines the foreground color (used where the character generator pattern bits are '1') and background color (character pattern bits are '0') for the character.

The colors are determined by independent bits for each additive primary color—red, green, and blue—and a fourth bit for intensity. The three primaries can be mixed in eight combinations (see figure 2) to form the basic RGB colors. The intensity control gives an additional eight colors, each a brighter version of its non-intensified counterpart. Photograph 1 shows all 16 IRGB colors; table 1 shows the composition of each color.

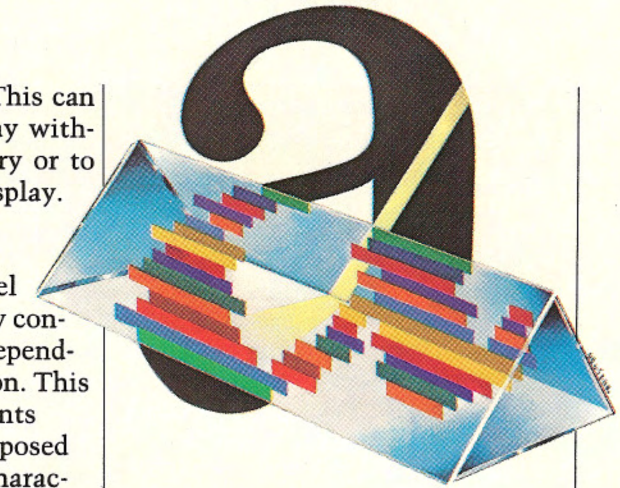
Figure 2: Additive Color Mixing



The three additive primary colors, red, green, and blue, combine to form white light. In pairs they form the complementary colors cyan, magenta, and yellow. Black is the absence of any light.

The low order four bits of the attribute byte determine the foreground color, and the high order bits determine the background. The adapter may be programmed for the high order attribute bit to control either background intensity, allowing sixteen background colors on the screen simultaneously, or foreground blinking. Standard IBM software selects the blinking function, restricting the display to eight background colors. Table 2 shows the format of the character code and attribute bytes.

With two bytes per displayed character, a 40x25 display uses 2000 bytes and an 80x25 display uses 4000 bytes. The display 16K byte display buffer is thus big enough for either eight or four complete pages of text. Normally, the adapter displays characters starting from the beginning of the buffer, but the 6845 can be programmed to start at any even address. This can be used to scroll the display without moving data in memory or to switch rapidly to a new display.



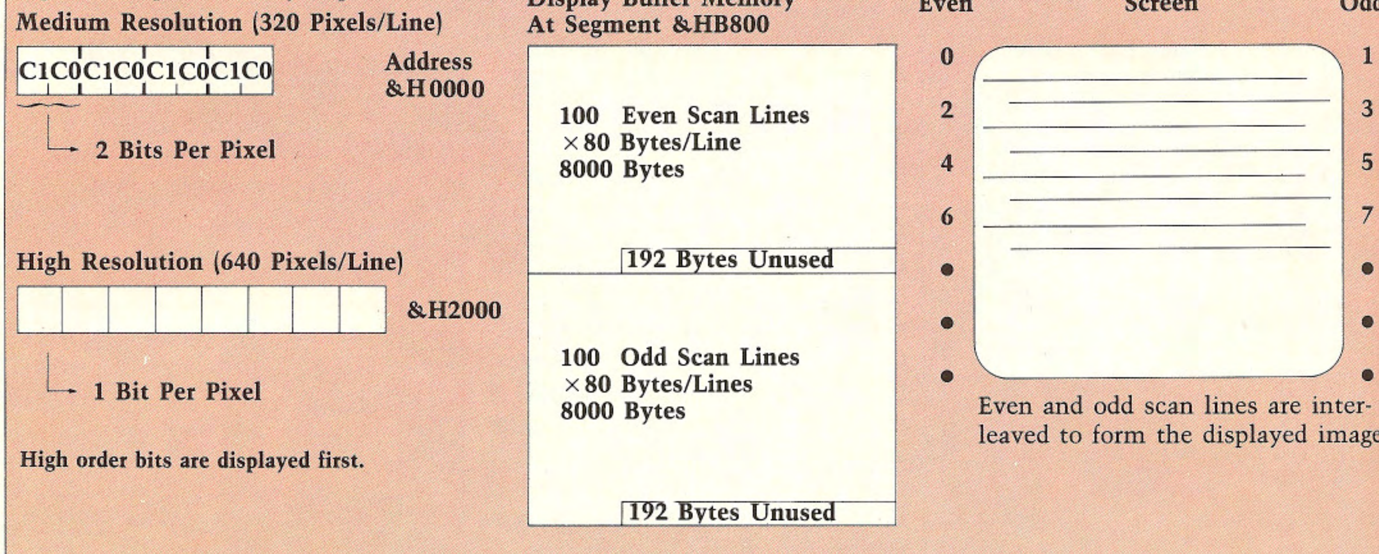
Graphics Modes

In graphics modes, each pixel of the display is individually controlled by one or two bits, depending on the selected resolution. This is sometimes called "all points addressable" graphics, as opposed to the "alpha mosaic" or "character" graphics available in alpha modes. All images, even characters, are formed by individually programmed pixels. The PC's ROM BIOS contains a character generator table for the first 128 character codes with the same patterns as those in the color adapter's ROM. In graphics modes, the software must read the patterns from the table and turn on the appropriate pixels. To read characters back from the screen memory in graphics modes, the BIOS programs actually match the patterns in the display buffer against those in the character table to determine the character displayed.

The increased flexibity in displayable images comes at the price of more memory. For example, in high resolution 640x200 graphics mode, 64 bits are required for each 8x8 pixel character cell; in alpha mode, 8 bits indirectly specify the entire 8x8 pattern. Both resolutions use 16,000 bytes of display buffer to represent the screen image.

In high resolution graphics two colors can be displayed, but, like Henry Ford's Model T, one of them—used for border and background—is always black. The foreground is controlled by the color select register.

Figure 3: Graphics Memory Map



In graphics modes, the 16,384 byte Display Buffer is divided into two 8,192 byte halves. The first half contains pixel data for even numbered scan lines, the second half contains data for the odd scan lines.

Medium resolution 320x200 graphics uses two bits per pixel, providing four colors. The background color, selected when both bits are 0, can be any one of 16 colors programmed into the background/border bits of the color select register. The other three colors are chosen from pre-selected color sets shown in table 3.

The display buffer is organized into two banks of 8K bytes each. Even numbered scan lines are displayed from the lower bank, odd numbered lines from the upper bank. This interleaving is unfortunate from the programmer's viewpoint, but results from a hardware/software tradeoff forced by the 6845's inability to address more than 128 character rows. This is explained more fully in the 6845 programming discussion.

Figure 3 summarizes the graphics memory map and pixel formats.

Programming the Color Graphics Adapter

In addition to the 16K display buffer memory, the adapter has several I/O ports through which its operation can be controlled and monitored. Table 4 summarizes the I/O device and bit assignments. Their operation is detailed below.

THE THREE PRIMARIES CAN BE MIXED IN EIGHT COMBINATIONS TO FORM THE BASIC RGB COLORS.

6845 Address Register (&H3D4)

This 5-bit write-only register is used to select one of the 18 internal data registers of the 6845 CRT controller by writing the register number to this port. The selected register is then read or written through the 6845 Data Register.

6845 Data Register (&H4D5)

This port is used to access the internal data register previously selected through the 6845 Address Register. The function of the various data registers is explained in the 6845 programming section below.

Mode Register (&H3D8)

The mode register is a 6-bit write-only register. Each bit controls one aspect of the operation of the display electronics, and together they establish the basic operating mode for the adapter. Table 5 summarizes the mode bits and standard settings for each of the IBM-supported video modes.

HOFFMANN: COLOR

Bit 0

High Resolution Dot Clock

Selects either a 7 or 14 megahertz dot clock, which determines both the rate at which dots from the character generator are sent to the screen, and when data is read from the display buffer. In alpha mode this selects 40 or 80 columns. A '1' selects the 14 MHz (80 column) clock.

Bit 1

Graphics Select

Selects between alpha and graphics modes. In alpha mode, successive bytes are interpreted as character code/attribute pairs, with the actual display patterns read from the character generator ROM. In graphics mode, pixels are directly determined by adjacent bits, or groups of bits, from successive bytes in the display buffer. A '1' selects graphics mode.

Bit 2

Black and White Select

Selects color or black and white mode for composite monitors or TV receivers. A '1' disables the color burst signal, giving a black and white image. With RGB monitors, this bit selects a variant color palette in 320x200 medium resolution graphics mode. Otherwise, this bit has no effect on RGB monitors.

Bit 3

Video Enable

Enables the video signal for the displayed area of the screen. When this bit is '0', the adapter's internal registers, which contain pixel or character and attribute data, are forced to '0', thus turning off the video signal. IBM suggests disabling the display when changing modes or reprogramming the 6845 CRT controller. A '1' enables the video signal. This bit does

Table 1 — Standard I-R-G-B Colors

Color Number	I R G B	Color Name	Composition
0	0 0 0 0	Black	
1	0 0 0 1	Blue	Blue
2	0 0 1 0	Green	Green
3	0 0 1 1	Cyan	Green + Blue
4	0 1 0 0	Red	Red
5	0 1 0 1	Magenta	Red Blue
6	0 1 1 0	Brown	Red + Green
7	0 1 1 1	White (Light Gray)	Red + Green + Blue
8	1 0 0 0	Dark Gray	Int
9	1 0 0 1	Light Blue	Int + Blue
10	1 0 1 0	Light Green	Int + Green
11	1 0 1 1	Light Cyan	Int + Green + Blue
12	1 1 0 0	Light Red	Int + Red
13	1 1 0 1	Light Magenta	Int + Red + Blue
14	1 1 1 0	Yellow	Int + Red + Green
15	1 1 1 1	Intense White	Int + Red + Green + Blue

Table 2 — Character Code/Attribute Formats

In alpha modes, each display position is defined by a character code/attribute pair. The character code is always the even-addressed byte, the attribute is the next higher odd-addressed byte.

Character Code Byte

7 6 5 4 3 2 1 0

Attribute Byte

7	6	5	4	3	2	1	0
I/ BL	Red	Grn	Blu	I	Red	Grn	Blu
Background						Foreground	

Attribute Byte Format

Bits 0-3	Foreground Color
Bits 4-6	Background Color
Bit 7	Background Intensity (Mode Bit 5 = '0') or Foreground Blink (Mode bit 5 = '1')

The COLOR statement in BASIC establishes the foreground, background, and border colors. Border color is changed immediately; foreground and background take effect for subsequently displayed characters.

COLOR [foreground] [,[background] [,border]]

Foreground may be 0 to 31, and sets attribute bits 7 and 0-3.
Background may be 0 to 7 and sets attribute bits 4-6.
Border may be 0-15, and sets bits 0-3 of the color select register.

THE DESCRIPTIONS IN THE IBM TECHNICAL REFERENCE MANUAL DON'T PROPERLY EMPHASIZE THE INDEPENDENCE OF EACH MODE BIT FROM THE OTHERS.

Table 3 — Medium Resolution (320 × 200) Color Sets

Colors in medium resolution are formed by combining bits from the color select register with the two pixel bits — C1 and C0 — from the display buffer. When C1 and C0 are both '0', the background color is displayed from CSR bits 0-3.

CSR Bit 4 (Int)	Pixel Data C1 (Red)	C0 (Green)	CSR Bit 5 (Blue)	IRGB Color Number	Color Name
(0)	0	0	(0)	??	Background
0	0	1	0	2	Green
0	1	0	0	4	Red
0	1	1	0	6	Brown
(1)	0	0	(0)	??	Background
1	0	1	0	10	Light Green
1	1	0	0	12	Light Red
1	1	1	0	14	Yellow
(0)	0	0	(1)	??	Background
0	0	1	1	3	Cyan
0	1	0	1	5	Magenta
0	1	1	1	7	White
(1)	0	0	(1)	??	Background
1	0	1	1	11	Light Cyan
1	1	0	1	13	Light Magenta
1	1	1	1	15	Intense White

In medium resolution black and white graphics modes (mode register bit 2 set to '1'), RGB monitors will display the following colors.

CSR Bit 4(Int)	Pixel Data C1 Red	C0 (Green)	(Blue)	IRGB Color Number	Color Name
(0)	0	0	(0)	??	Background
0	0	1	1	3	Cyan
0	1	0	0	4	Red
0	1	1	1	7	White
(1)	0	0	(0)	??	Background
1	0	1	1	11	Light Cyan
1	1	0	0	12	Light Red
1	1	1	1	15	Intense White

Note: Bit values in parentheses have no effect on the background color.

not affect the video signal for displaying the screen border area.

Bit 4
640 Dot Graphics
In conjunction with bit 1, enables high resolution graphics, with one bit per pixel. A '1' selects 640 1-bit pixels, a '0' 320 2-bit pixels per line.

Bit 5
Blink Enable
In alpha modes, this bit determines whether the high order attribute bit controls background color intensity or foreground blinking. A '1' enables blinking for characters with attribute bit 7 set, and restricts the background to eight colors. In this mode, bit 4 of the Color Select Register at I/O address &H3D9 controls the intensity of all background colors. A '0' inhibits blinking and allows all 16 background colors. This bit has no effect in graphics modes (i.e., when bit 1 is '0').

The descriptions above differ slightly from those in the IBM Technical Reference Manual, which don't properly emphasize the independence of each mode bit from the others. For example, IBM calls bit 0 the "80 x 25 mode alpha mode" bit, presumably because it is one of the bits that must be set to achieve that mode. But bit 0 does not determine alpha versus graphics, and none of the mode bits have anything to do with the number of lines displayed: that's determined by the 6845.

IBM's version is no doubt well-intentioned; after all, "80 x 25" makes more sense to most people than "high resolution dot clock," but it can lead to misunderstandings about how the adapter works.

COLOR SELECT REGISTER (&H3D9)

This 6-bit write-only register controls the screen border color in alpha modes, background color and foreground color set in medium resolution (320x200) graphics modes, and the foreground color in high resolution (640x200) graphics modes.

IN HIGH RESOLUTION GRAPHICS, TWO COLORS CAN BE DISPLAYED, BUT, LIKE HENRY FORD'S MODEL T, ONE OF THEM IS ALWAYS BLACK.

Bits 0-3
Alpha border, 320 Background, 640 Foreground
In alpha modes, these bits determine the color of the screen border area. In 320x200 graphics, they select the background color displayed for pixel values of '00'. In 640x200 graphics, they select the foreground color displayed for pixel values of '1'. The background and border are always black in high resolution graphics modes. The bits are arranged in the same order as the color attribute bits in alpha mode:

Bit	3	2	1	0
	Intensity	Red	Green	Blue

Bit 4
Alpha Background / 320 Graphics Foreground Intensity
This bit selects the intensity for background colors in alpha mode when blink is enabled (mode register bits 1 and 5 are both '0'). In medium resolution (320 pixel) graphics, this bit controls the intensity of the foreground color set.

Bit 5
Medium Resolution Graphics Color Set (Blue Control)
Determines the color set used for foreground colors in medium resolution (320x200) graphics by controlling the presence or absence of blue. Red and green are selected by the high and low order bits of the pixel — C1 and C0, respectively. The resulting combinations are shown in table 3. Bit 5 has no effect in modes other than medium resolution graphics.

Table 4 — Color/Graphics Adapter I/O Devices

I/O Port Address	Device Name	Read/ Write	Active Data Bits
&H3D4	6845 Address Register	W	- - - 4 3 2 1 0
&H3D5	6845 Data Register	R/W	- - - 4 3 2 1 0
&H3D8	Mode Select Register	W	7 6 5 4 3 2 1 0
&H3D9	Color Select Register	W	- 5 4 3 2 1 0
&H3DA	Status Register	R	- - - - 3 2 1 0
&H3DB	Clear Light Pen Latch	R/W	- - - - - - - -
&H3DC	Set Light Pen Latch	R/W	- - - - - - - -

Mode Register (&H3D8)

Bit 0	High Res Dot Clock (80 Character alpha)
1	Graphic Select
2	Black & White Select
3	Enable Video
4	640 Graphics Select
5	Alpha Blink Enable

Color Select Register (&H3D9)

Bit 0	Blue	Alpha Border/Graphics Background
1	Green	Alpha Border/Graphics Background
2	Red	Alpha Border/Graphics Background
3	Intensity	Alpha Border/Graphics Background
4	Intensity	Alpha Background/Medium Resolution Foreground
5	Medium Resolution Foreground Color Select (Blue)	

Status Register (&H3DA)

Bit 0	Display Inactive
1	Light Pen Trigger Set
2	Light Pen Switch Open
3	Vertical Sync

Notes: 1. The 6845 also responds to other even/odd I/O addresses in the range &H3D0 to &H3D7. The addresses shown are the recommended standard used by all IBM software.
2. The 6845 Data Register may have from 4 to 8 active bits, depending on the internal register selected by the 6845 Address Register.
3. The Clear and Set Light Pen functions are activated by any I/O read or write to the appropriate port.

STATUS REGISTER (&H3DA)

This 4-bit read-only register provides two signals for monitoring the video timing, and two for the light pen interface.

Bit 0
Display Inactive
This bit is the inverted display enable bit from the 6845 CRT controller. It is '1' during the horizontal and vertical blanking intervals, and '0' during the active display interval. When this bit is '1', CPU accesses to the display buffer will not interfere with the display. When the high resolution dot clock is selected, as in 80 column alpha mode, unsynchronized direct access to the display buffer can cause "snow" in the picture if the display is enabled.

(continued on page 45)

(continued from page 41)

Bit 1

Light Pen Trigger Set

When '1', this bit indicates the light pen trigger has been set. The buffer memory address at the time the trigger was set can be read from the 6845 light pen register. This bit can be set by the light pen input going high or by an OUT to port &H3DC. It can be reset by an OUT to port &H3DB.

Bit 2

Light Pen Switch Status

This bit shows the state of the light pen switch. A '1' means the switch is open, a '0' means it is closed.

Bit 3

Vertical Sync

This bit can be used to synchronize with the start of a vertical retrace, which begins each field 60 times per second. It is used in diagnostics (along with bit 1) to check that the video timing signals are being generated correctly. A transition from '0' to '1' marks the beginning of the vertical sync pulse.

Light Pen Latch Reset (&H3DB) and Set (&H3DC)

Any output to these ports—the data doesn't matter—resets or sets the light pen latch as indicated. The latch must be cleared before the 6845 can read the light pen again. These ports can also be used to fake light pen input, either for diagnostic purposes or to synchronize a program with the display refresh.

Table 5 — Standard Video Models

Mode Register (&H3D8)

BIOS Mode	Description	Bit 5 Blink Enable	4 640 Dot Graphics	3 Enable Video	2 B/W	1 Graphics	0 Hight Res	Hex	Dec
0	40 × 25 Alpha B/W	1	0	1	1	0	0	2C	44
1	40 × 25 Alpha Color	1	0	1	0	0	0	28	40
2	80 × 25 Alpha B/W	1	0	1	1	0	1	2D	45
3	80 × 25 Alpha Color	1	0	1	0	0	1	29	41
4	320 × 200 Graphics Color	(1)	0	1	0	1	0	2A	42
5	320 × 200 Graphics B/W	(1)	0	1	1	1	0	2E	46
6	640 × 200 Graphics B/W	(0)	1	1	1	1	0	1E	30

To set these modes from BASIC, the following statements can be used:

BIOS Mode	BASIC Statement(s)
0	SCREEN 0,0: WIDTH 40
1	SCREEN 0,1: WIDTH 40
2	SCREEN 0,0: WIDTH 80
3	SCREEN 0,1: WIDTH 80
4	SCREEN 1,0
5	SCREEN 1,1
6	SCREEN 2

Notes: 1. Mode 1 or 3 is selected by default at power on, depending on the setting of switches 1-5 and 1-6 on the system board.

2. Bit 5 (blink enable) has no effect in graphics modes. The values shown for modes 4, 5, and 6, are those actually written to the mode register by BIOS.

3. BIOS saves the current mode register and color register values in segment 0, location &H465 and &H466, respectively. value in.

4. The color register is initialized to &H30 for modes 0 to 5, and to &H3F for mode 6. This results in white foreground against a black background.

Standard Operating Modes

Let's look at the seven standard operating modes supported by IBM for the Color/Graphics Adapter. Table 5 shows the four alpha and three graphics modes, numbered as they are for the BIOS video I/O Set Mode function (interrupt &H10, AH = 0, AL = mode number) and the associated contents of the CGA mode register.

The mode register values for the standard modes are easily understood as straightforward combinations of the bits that specify the desired features. The major decisions are graphics or alpha (bit 1), and color or black and white (bit 2). In graphics mode we must further select 320 or 640 pixels per line (bit 4). In alpha mode we must choose between 40 or 80 columns per line (bit 0), and 8 background colors with blinking characters or 16 background colors with no blinking (bit 5).

How many modes are there? There are 64 possible combinations of the six mode bits, but half of them have video enable (bit 3) turned off. Of the 32 remaining visible combinations, half are alpha and half are graphics. Of the 16 possible visible alpha combinations, the 8 with 640 dot graphics turned

Table 6 — 6845 Data-Register Summary

Register	R/W	Units	Programmed Value	Max Value
00 H. Total	W	Char	Total chars-1	255
01 H. Displayed	W	Char	NR displayed	255
02 H. Sync Pos.	W	Char	Nr chars to sync	255
03 H. Sync Width	W	Char	Nr char	15
04 V. Total	W	Char Row	Total rows-1	255
05 V. Adjust	W	Scan Line	Nr scan lines	31
06 V. Displayed	W	Char Row	Nr rows	127
07 V. Sync Pos.	W	Char Row	Nr rows to sync	127
08 Interlace Mode	W	---------	(Note 1)	3
09 Max Scan Line	W	Scan Line	Lines per row-1	31
10 Cursor Start	W	Scan Line	First line (Note 2)	127
11 Cursor End	W	Scan Line	Last line	31
12 Start Addr (H)	W	---------	High 6 bits (Note 3)	63
13 Start Addr (L)	W	---------	Low 8 bits	255
14 Cursor Addr (H)	R/W	---------	High 6 bits (Note 3)	63
15 Cursor Addr (L)	R/W	---------	Low 8 bits	255
16 Light Pen (H)	R	---------	High 6 bits (Note 3)	63
17 Light Pen (L)	R	---------	Low 8 bits	255

Notes:

1. Interlace modes 0 and 2 are non-interlace. Mode 1 is interlace sync (duplicates data on even and odd fields). Mode 3 is interlace sync and video (even scan lines displayed on even fields, odd scan lines on odd fields).

2. Bits 0-4 specify first scan line. Bits 5 and 6 control cursor display and blinking as follows:

Bit 6	5	Cursor Display Mode
0	0	Non-blinking cursor
0	1	Cursor not displayed
1	0	Cursor blinks rapidly (1/16 field rate)
1	1	Cursor blinks slowly (1/32 field rate)

The color adapter has its own external cursor blinking logic, so in practice bit 6 should always be programmed to '0'. Bit 5 can be used to disable the cursor display.

3. These registers are paired to form a 14 bit display buffer address. This address is one half of the address of the even (character code) byte in each character/attribute pair.

OF THE 16 POSSIBLE VISIBLE ALPHA COMBINATIONS, THE 8 WITH 640 DOT GRAPHICS TURNED ON DON'T TURN OUT TO BE USEFUL. TRY IT, YOU'LL SEE.

on don't turn out to be useful (try it—you'll see). The blink enable bit has no effect in graphics mode, so only eight real graphics modes remain. Total sensible combinations: eight alpha and eight graphic. So why does IBM only have seven modes? Where are the other nine? What are they hiding?

The four IBM alpha modes all have blink enabled, for "normal operation." To get the other four, the mode register must be set directly to turn off bit 5. Since BIOS saves the value written to the mode register in absolute location &H465, it's easy enough to read it, change it, put it back, and output it. The following BASIC program does just that.

```
10 '--Change Alpha Mode to
   Non-Blink
15 '-- 16 Background Colors
20 MODESAVE = &H465
25 MODEREG = &H3D8
30 BLINKENABLE = &H20
35 DEF SEG = 0
40 SCREEN 0,1 '--Set Alpha
   Color
45 MODE = PEEK (MODE-
   SAVE) AND NOT
   BLINKENABLE
50 POKE MODESAVE, MODE
55 OUT MODEREG, MODE
```

It's good practice to keep the MODESAVE location up to date when experimenting with non-standard modes, because BIOS will sometimes restore the mode register from that location when you're not expecting it. One such time is when CTRL-NumLock is pressed to pause execution of a program. If MODESAVE isn't correct, you may find the display back in the last "standard" mode.

What about "secret" graphics modes? Four of the eight combinations have the high resolution dot clock (bit 0) set. In alpha mode this bit means 80 columns are displayed, requiring 160 bytes from the display buffer for each line (80 character code/attribute pairs). In graphics mode the same 160 bytes are fetched, effectively doubling the number of pixels per line. Unfortunately, twice as many pixels on each of the 200 lines would require 32K bytes of buffer memory, which the adapter doesn't have.

That leaves us with four useful graphics modes, where IBM supports three. The missing combination is 640 by 200 *color* graphics. Can the adapter really do that? The answer is a qualified yes.

(continued on page 13

HOFFMANN: COLOR

(continued from page 46)

Color with High Resolution Graphics

Remember that in 640 by 200 graphics, each pixel is represented by a single bit in the display buffer. Each pixel has two states, '0' and '1', which ordinarily display as black and white, respectively. The horizontal resolution of most composite displays is many fewer than 640 dots, in fact it's closer to 300. This means that two adjacent pixels in 640 dot mode will appear as a single brighter dot rather than as two distinct dots.

With the color burst enabled (mode bit 2 set to '0'), a composite color monitor or receiver will interpret portions of the high frequency video signal as color information, giving rise to "artifact" or "false" colors. Groups of four adjacent pixels can be treated as one large pixel with 16 possible color values. These artifact colors are not the same as the 16 "true" IRGB colors (see photograph 1). The artifact colors formed by each of the 16 four-bit values, aligned on half-byte boundaries, are shown in photograph 2.

The qualifications mentioned above are two: this is not really *high resolution* color, because only 160 effective pixels are possible per line, and it only works on composite monitors or TV sets. Even so, the results are quite pleasing and very simple to achieve, even from BASIC.

The first step is to enable the color burst signal, by setting mode register bit 2 to '0'. In BASIC this should be possible with a SCREEN 2,0 statement, but it isn't; BASIC ignores the color burst parameter for high resolution mode. The manual even states that "since black and white are the only colors in high resolution graphics (mode = 2), this parameter will not have much effect." The parameter may have no effect, but the colors are just over the rainbow. Enter the following little BASIC program, RUN it, then LIST it.

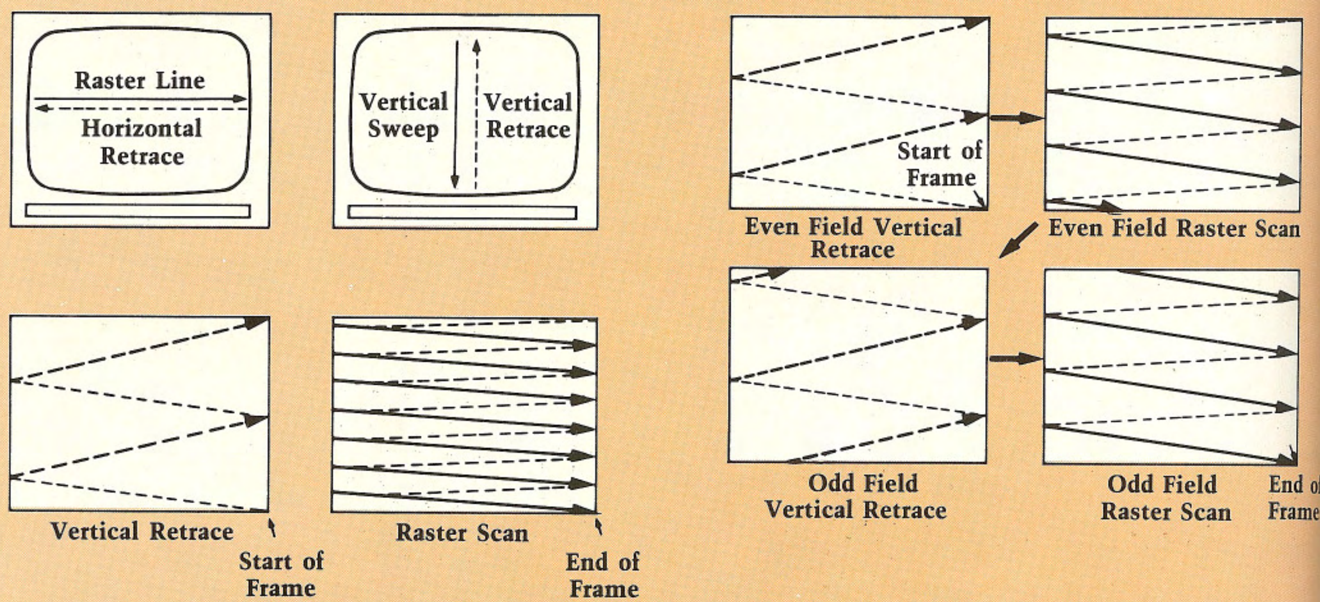
Table 7 — 6845 Register Values for Standard Modes
(All values are in decimal)

	Register	Units	R/W	40×25 Alpha	80×25 Alpha	320/640×200 Graphics
00	Horizontal Total	Char	W	56	113	56
01	Horizontal Displayed	Char	W	40	80	40
02	Horiz. Sync Position	Char	W	45	90	45
03	Horiz. Sync Width	Char	W	10	10	10
04	Vertical Total	Char Row	W	31	31	127
05	Vert. Total Adjust	Scan Line	W	6	6	6
06	Vertical Displayed	Char Row	W	25	25	100
07	Vert. Sync Position	Char Row	W	28	28	112
08	Interlace Mode	---	W	2	2	2
09	Max Scan Line Addr.	Scan Line	W	7	7	1
10	Cursor Start	Scan Line	W	6	6	6
11	Cursor End	Scan Line	W	7	7	7
12	Start Addr. (High)	---	W	0	0	0
13	Start Addr. (Low)	---	W	0	0	0
14	Cursor Addr. (High)	---	R/W	--	--	--
15	Cursor Addr. (Low)	---	R/W	--	--	--
16	Light Pen (High)	---	R	--	--	--
17	Light Pen (Low)	---	R	--	--	--

```
10 '-- Set Up High Resolution
       Graphics
15 '-- With 16 Colors
20 MODESAVE = &H465
25 MODEREG = &H3D8
30 BWENABLE = &H04
35 DEF SEG = 0
40 SCREEN 2,0 '-- Set High Res
      Graphics
45 MODE=PEEK(MODE-SAVE)
      AND NOT BWENABLE
50 POKE MODESAVE, MODE
55 OUT MODEREG, MODE
```

EACH PIXEL HAS TWO STATES, '0' AND '1', WHICH ORDINARILY DISPLAY AS BLACK AND WHITE.

Figure 4: Raster Scanning Patterns



A. The independent horizontal and vertical sweeps of the elector beam generate a regular raster pattern when they are synchronized by the CRT timing circuits.

B. When alternate fields are delayed by half the horizontal time interval, an interlaced raster pattern results.

Another method for filling a rectangular area with a color pattern uses the LINE statement to draw vertical lines in adjacent columns, skipping columns where the color code has a '0' bit. This is somewhat faster than POKEing, and allows us to deal in pixel coordinates rather than buffer addresses, a welcome simplification that avoids the messy business of interleaving scan lines. The following subroutine assumes a coordinate system of 160x200 and fills a block specified by its upper left and lower right corners.

THE COLOR RESULTS ARE QUITE PLEASING AND VERY SIMPLE TO ACHIEVE, EVEN FROM BASIC.

```
1000 '-- Fill Block (x1,y1)-
         (x2,y2) with color C
1010 FOR X = X1 TO X2
1020 X4 = X * 4 '--Trans-
         form X to 640 Coords
1030 CMASK = 8
1040 FOR I = 0 TO 3
1050 IF (CMASK AND C)
         = 0 THEN 1080
1060 XC = X4 + I
1070 LINE (XC,Y1)-
         (XC,Y2)
1080 CMASK =
         CMASK / 2
1090   NEXT I
1100 NEXT X
1110 RETURN
```

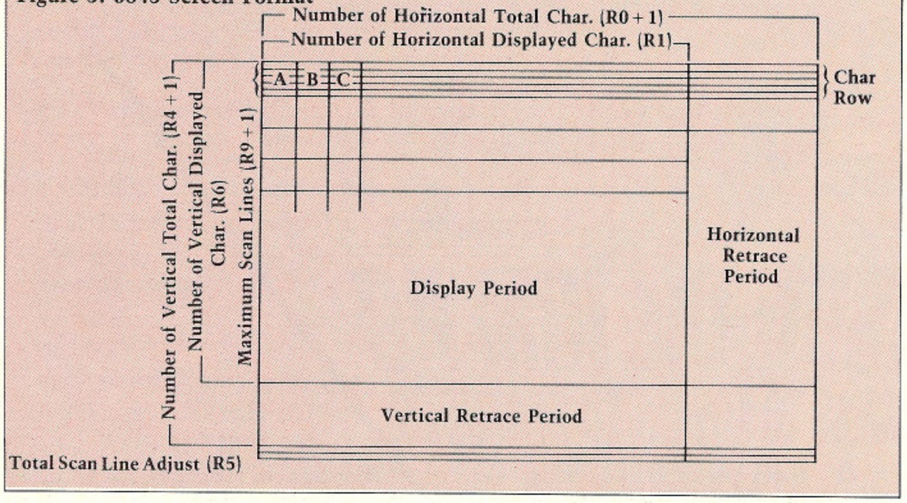
The program "BOXES" (page 183) displays randomly positioned boxes drawn in high resolution graphics, then PAINTS them with a color pattern using the BASIC 2.0 extension for tiling an area with a repeating pattern. This is much simpler and faster than either of the previous two methods, but requires BASICA version 2.0. A typical display is shown in photograph 3. It might be fun to try variations using circles or arbitrary polygons.

TELEVISION FUNDAMENTALS

Some basic knowledge of the way television images are formed on the screen is necessary to fully understand the operation of the display adapter, and especially the role of the 6845 CRT controller.

The image never really exists all at once on the screen. What you see is the illuminated trail left by a continuously moving electron beam which repeatedly traces a rectangular pattern of regularly spaced horizontal lines, called a *raster*. As the beam moves, its intensity at the instant it passes a point determines the brightness of the picture at that point. Color displays have three beams that move together, but have independently controlled intensities, one for each of the primary colors red, green, and blue.

Figure 5: 6845 Screen Format



A horizontal deflection circuit in the monitor sweeps the beam from the left edge of the screen to the right, then more quickly back to the left. The rightward motion traces a *scan line*, and the leftward return is called the *horizontal retrace*. A similar, but independent, vertical deflection circuit sweeps the beam from the top edge to the bottom and back again. The downward motion is called *vertical sweep*, and the upward return is the *vertical retrace*. When the horizontal and vertical sweeps occur simultaneously and are synchronized, the beam traces a raster on the screen. The scan lines are tilted downward to the right due to the vertical sweep. Figure 4 illustrates these various scanning patterns.

The time from the end of one horizontal or vertical sweep to the end of the next, including the associated retrace, is called the *H-interval* or *V-interval*, respectively. Monitors are designed so that the V-interval is an exact multiple of the H-interval. This makes it possible for the electron beam to start in the lower right corner of the screen, zig-zag its way to the top right corner for the start of another scan of the entire raster, and end up in the lower right corner again to repeat the cycle.

The display adapter must synchronize the horizontal and vertical deflections with each other and the information to be displayed. This is done by means of horizontal and vertical sync pulses, which define the start of the respective retraces, and the video signal, which defines the display intensity and color.

The rate at which the video information changes determines the number of dots, or pixels (picture elements), per scan line. The color graphics adapter operates at a dot rate of either 14.31818 MHz or 7.15909 MHz. Standard U.S. television monitors and receivers make 15,750 complete horizontal scans per second, giving a scan line time of 63.49 microseconds. At the 14 MHz rate that gives 909 potential dots per line, but only 640 are displayed. The remaining time is allocated to the screen border and horizontal retrace interval.

Vertical timing is calculated in a similar fashion. A complete vertical scan, including retrace, occurs 60 times per second. Dividing the V-interval by the H-interval gives 262.5 lines per vertical sweep. In normal broadcast television, the extra half line causes the beam to begin every other downward pass at the center of the screen instead of the upper left corner, tracing a path exactly between the scan lines of the previous pass. This is called interlace and results in a complete 525 line frame 30 times per second, each composed of successive even and odd fields of 262.5 lines each.

The color graphics adapter normally operates in non-interlaced mode, with identical 262-line fields scanned 60 times per second. This gives a steadier image by sacrificing vertical resolution for more frequent refresh. Only 200 lines are used to display the image, with the other 62 devoted to border and vertical retrace.

Motorola 6845 CRT Controller

The 6845 CRT controller coordinates the access to the display buffer with the horizontal and vertical timing of the display monitor. The controller is fed by the basic 7 or 14MHz dot clock, which is used to generate sets of output signals: video timing control, consisting of horizontal and vertical sync and a display enable signal; 13 memory address lines, used to address up to 8K pairs of bytes in display buffer; and three row lines used to address the character generator ROM. In graphics modes the low order row address line is used as the highest order refresh memory address line. This causes even and odd scan lines to be fetched respectively from the lower and upper halves of the display buffer.

Types of Displays

The color graphics adapter supports two types of display devices: direct drive monitors and composite monitors, including television receivers used with an RF modulator. Either type of display may be monochrome or color.

Direct drive monitors have separate vertical sync, horizontal sync, and video inputs. Monochrome monitors have a single video signal. Direct drive color monitors, often called RGB monitors, have three video inputs, one for each of the red, green, and blue electron guns. If these inputs are digital (they are independently on or off, with no intermediate values) the monitor can display eight colors. Some digital RGB monitors have a fourth input signal that controls the intensity of all three guns, allowing them to display 16 colors. In analog RGB monitors the intensity of the beam is proportional to the voltage applied to the video inputs. These monitors are much more expensive than digital monitors but have the advantage of being able to display a practically infinite number of colors. The IBM Color Graphics Adapter only generates binary RGB and intensity signals, so it cannot take advantage of the capabilities of analog color monitors. They can still be used, however, with proper signal level converts.

Composite monitors accept a combined signal that contains both sync and video information. They are generally less expensive and have poorer resolution than direct drive monitors.

Television receivers are essentially composite monitors that extract the composite video signal from a modulated radio frequency (RF) carrier. Because the CGA generates only unmodulated, or baseband, composite video, an external RF modulator is required to produce the television frequency signal required by a receiver.

Despite their poorer picture quality, composite monitors and TV sets have several advantages over direct drive monitors. The first is price. Even with the added cost of the RF modulator — about $30 to $70 — televisions are the least expensive displays available for the IBM PC. Composite monitors are only slightly more expensive, and of much better quality than standard TV sets. A related advantage is multiple use. It's a lot easier to convince your spouse to let you buy another TV than it is to watch the 11 o'clock news on an RGB display. Composite monitors can also be used with video cassette recorders and disk players. Finally, only composite color displays can be used to get multiple colors in 640x200 high resolution color mode, such as the Microsoft Flight Simulator, look much worse on expensive direct drive monitors than on cheap black and white TV sets. On RGB displays, this technique produces coarse black and white images instead of uniformly colored ones.

The card has two connectors on the rear panel for attaching display monitors, and another on the card itself. A fourth connector, a 6-pin Berg strip near the rear of the card, is for a light pen. The round RCA phono jack on the rear bracket (just like the ones on the back of your stereo) is for a composite video monitor. The same composite video output is available on a 4-pin Berg strip connector on the card for attaching an RF modulator. The 9-pin D shell connector on the rear bracket is for a direct drive RGB monitor, and carries the separate red, green, blue, intensity, horizontal and vertical sync signals.

HOFFMANN: COLOR

The 6845 has 18 internal data registers that can be programmed to handle a variety of display formats. These registers are accessed by writing the data register number to the 6845's Address Register at I/O address &H3D4, then reading or writing the desired value at I/O address &H3D5.

Table 6 summarizes the 18 data registers. Table 7 shows the initial register values for the standard alpha and graphics modes.

6845 Register Descriptions

The first 10 registers define the character and screen formats. These values must be set to generate the proper timing intervals for the monitor used. The video timing is programmable in terms of character times, which are always eight dot times for the horizontal dimension and vary depending on the programmed number of scan lines per character for the vertical dimension. Figure 5 shows the CRT screen format as viewed by the 6845.

Standard U.S. television monitors have a horizontal interval of 63.5 microseconds, and a vertical interval of 1/60 second or 16,667 microseconds. The following discussion uses the 40 by 25 alpha mode as an example, with characters composed of 8 scan lines of 8 dots each, and a dot clock rate of 7.15909 MHz. This gives a dot time of 139.7 nanoseconds and a character time of 1.12 microseconds.

THE 6845 HAS 18 INTERNAL DATA REGISTERS THAT CAN BE PROGRAMMED TO HANDLE A VARIETY OF DISPLAY FORMATS.

Horizontal Timing Registers (RO through R3)

Horizontal Total (RO)

This 8-bit register determines the frequency of the horizontal sync pulse, which should closely match the duration of the horizontal interval. It is programmed to the total number of character times minus one. (H-interval / CharTime = 63.5 / 1.12 = 57 - 1 = 56 Chars)

Horizontal Displayed (R1)

This 8-bit register is programmed to the number of characters displayed per horizontal line.

Horizontal Sync Position (R2)

This 8-bit register is programmed to the character position at which the horizontal sync pulse should occur. This should be approximately five microseconds after the last displayed character. A smaller value places the sync pulse closer to the last displayed character, thus moving the image to the right. Similarly, a larger value moves the image to the left. This is how the DOS MODE command adjusts the horizontal position of the display.

Horizontal Sync Width (R3)

This is a 4-bit register which determines the width of the horizontal sync pulse.

Vertical Timing Registers (R4 to R9)

Vertical Total (R4) and Vertical Total Adjust (R5)

These registers determine the frequency of the vertical sync pulse, which should match the duration of the vertical interval (1/60 second). The integer part of the calcu-

(continued on page 163)

(continued from page 145)

lated number of character line times minus one is programmed into R4, and the remainder in scan lines is programmed into R5. This allows the vertical sync timing to be quite precise, and eliminates vertical rolling of the image.

(V-interval / CharLineTime = 16667 / (8 * 63.5) = 32.81 char lines)

(R4 = 32 - 1 = 31, R5 = 8 * .81 = 6.48 = 6 scan lines)

Vertical Displayed (R6)

This 7-bit register is programmed to the number of character rows to be displayed on the screen.

Vertical Sync Position (R7)

This 7-bit register determines the position of the vertical sync pulse with respect to the first displayed row. The nominal value is about 1524 microseconds past the last displayed character row and is programmed in character row times. Smaller values will lower the displayed image; larger values will raise it.

Interlace Mode (R8)

This 2-bit register controls the raster scan pattern. A value of 0 or 2 selects normal, or non-interlaced mode. In this mode, each field traces the same raster on every vertical sweep. In interlaced modes, the vertical sync position of every other vertical sweep is offset by 1/2 of the H-interval time, resulting in two alternating sets of interlaced scan lines (see figure 4). A value of 2 selects interlaced sync mode, where each field displays the same information. The effect is to fill in the spaces between scan lines, which can make characters appear more solid. A value of 3 interlaces

A COMPLETE VERTICAL SCAN, INCLUDING RETRACE, OCCURS 60 TIMES PER SECOND.

both the sync and video signals, displaying even lines in the even field, and odd lines in the odd field. This effectively doubles the vertical resolution of the display. Both interlace modes have the disadvantage of increased image flicker, since the image is refreshed at half the non-interlaced rate. For interlaced operation, the horizontal total character count must be even (R0 must be odd). For mode 3 only, there must be an even number of displayed character rows (R6) with an even number of scan lines in each (R9 must be odd), and the cursor start and end registers must both be even or both be odd.

Maximum Scan Line Address (R9)

This 5-bit register determines the number of scan lines per character row, including blank lines for spacing between rows. It is programmed to one fewer than the number of odd scan lines in each row.

OTHER REGISTERS

Cursor Start (R10) and End (R11)

These registers determine the format of the cursor in the character block. Bit 6 of R10 is intended to enable cursor blinking, but the color adapter has its own external blinking logic. When bit 6 is '0', a '1' in bit 5 disables the cursor display, and a '0' enables the cursor. Bits 0-4 of R10 set the cursor start scan line, and the 5 bit register R11 sets the cursor end scan line.

Start Address (R12, R13)

This 14 bit register determines the address in the refresh buffer from which the first character of the frame is fetched. The upper 6 bits are written to R12 and the lower 8 bits to R13. These registers should be programmed to the number of character/attribute byte pairs from &HB8OO:O to be skipped. The actual refresh memory adress for the beginning of a frame will always be even.

Light Pen Register (R16, R17)
This 14-bit read-only register is used to store the current refresh memory address when the light pen input signal goes high. The registers are in the same format as R12 and R13, and indicate the number of byte pairs from the beginning of &HB8OO:O.

Cursor Register (R14, R15)
This 14-bit register stores the cursor location in the same refresh address format as R12 and R13. It may be read or written.

PROGRAMMING THE 6845 FOR ALPHA-GRAPICS
The 6845 always treats the display as an array of characters, whether the adapter is in alpha or graphics mode. The standard IBM graphics modes are set up as 100 rows of 40 characters each, with each character row being two scan lines high. This gives the 200 line vertical resolution. The same external logic that fetches two bytes per character column from the display buffer in alpha modes is also used in graphics modes, but the video information is formed directly from the pixel information in the buffer rather than from the character generator.

A variety of low resolution, 16 color graphics modes can be programmed with the adapter's alpha mode by changing the character dimensions of the screen. The technique uses the character code &HDE, which has a pattern of four columns of background and four of foreground. By filling every character code position with &HDE, the two nibbles of each attribute byte can be used to individually set any one of the 16 IRGB colors in each pixel. The maximum horizontal resolution in this method is 160—80 characters with two halves each. The vertical resolution can vary from 25 rows of 8 scan lines each, to 100 rows each two scan lines high.

Table 8 shows the 6845 parameters for several alpha-graphics formats. The program "KSCOPE" (page 183) uses this technique in 160x100 to generate kaleidescopic patterns, with four axes of symmetry. Obviously, there are countless variations.

The color adaptor has a problem with color in high resolution alpha mode on composite monitors or TV sets which require a bit of special handling. The Technical Reference Manual indirectly alludes to this when it says that alphanumeric mode can display up to 25 rows of 40 characters for color TVs and up to 25 rows of 80 characters on direct drive monitors. Color in 80 columns attempted in the normal fashion works fine on RGB displays but looks quite monochromatic on a TV set.

The problem is apparently caused by the way the adaptor generates the color reference information in the composite video signal. It can be overcome by setting the border color to yellow ('1110' in bits

MONITORS ARE DESIGNED SO THAT THE V-INTERVAL IS AN EXACT MULTIPLE OF THE H-INTERVAL.

COLOR

0-3 of the color select register) and adjusting the brightness and contrast controls on the set. It's still hard to read text, but the display will decode the color signals more or less correctly, which is just what we need for the alphagraphics technique described above. The only drawback is that the border will be yellow. This can be done in BASIC with

```
COLOR ,,14
```

where the third parameter is the border color. Other settings also give interesting results: setting the border to blue (1) still gives a yellow border, but complements each color in the displayed area — red, green, and blue become cyan, magenta and yellow respectively.

In the accompanying example program KSCOPE, change line 1380 to

```
1380 POKE COLORSAVE,14:
OUT COLORREG,14
```

for operation on composite monitors or TV sets.

Conclusion

The color adapter is a very versatile device, capable of much more than the standard IBM modes support. The information and techniques presented here should provide a good foundation for further experiments in the realm of the clever, or just a better appreciation of the ordinary. After all, there's no place like home. □

References

1. IBM Personal Computer Technical Reference, IBM Corporation, P.O. Box 1328, Boca Raton, Florida 33432.

2. Motorola MC6845 Advance Information AD1-465, Motorola Semiconductors, 3501 Ed Bluestein Blvd., Austin, Texas 78721.

3. Raster Graphics Handbook, Conrac Division, Conrac Corporation, 600 North Rimsdale Avenue, Covina, California 91722.

Thomas V. Hoffmann is Director of Advanced Systems Development for General Instrument Corporation and an expert in computer graphics.