The Hercules Graphics Card

This card connects directly to the display and provides high-resolution graphics for the IBM PC

by Tom Wadlow

When I bought an IBM Personal Computer (PC), I spent quite a bit of time comparing the merits of the monochrome display with those of the color display. One side of the debate asked: How can you have a home computer without some sort of graphics display? Look at all the flexibility you'd have, writing color-coded software and making graphs and charts. And what about games? The other side of the debate was, unfortunately, much more pragmatic: Are you going to be able to stare at color characters on the display all day?

Pragmatism won.

But now there's a product for people who want crisp, attractive text as well as graphics. Hercules Computer Technology's Graphics Card (see photo 1) directly replaces the IBM monochrome card; it plugs into a slot in the IBM backplane and connects to the PC display (i.e., the green-phosphor video monitor). The card produces a display that is indistinguishable from that of the standard monochrome board. Twiddle a few bits, however, and the card provides a 720- by 348-point graphics display as well. In addition, the Hercules card has two displayable graphics pages. Because you can write into one page while displaying the other, some types of animation are possible.

The Hercules board looks and feels well constructed. It's a full-sized PC board that comes with an extra card guide. The board's connectors duplicate those of the IBM monochrome card—one for connection to the IBM monitor and one for the parallel printer interface. To use this board with a standard video monitor, you would prob-

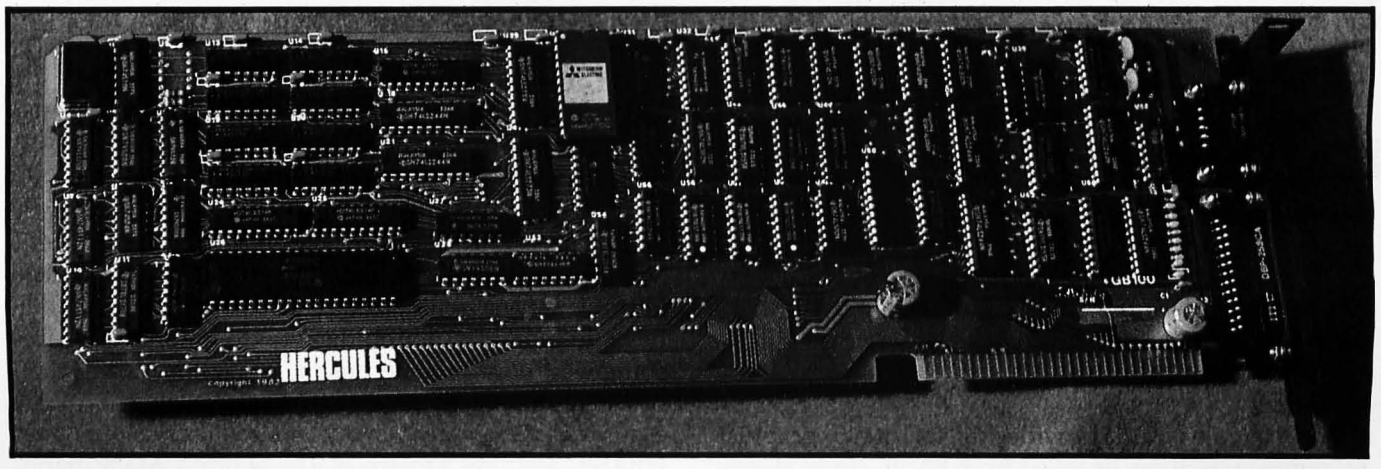

Photo 1: *The Hercules Graphics Card plugs into a slot on the IBM PC's backplane, replacing IBM's monochrome card.*

Command	Operation
GMODE	Enters graphics mode
TMODE	Enters text mode
CLRSCR	Clears the current graphics page
GPAGE	Sets the current graphics page for writing (either 0 or 1) but doesn't affect the display
LEVEL	Sets the intensity level for subsequent writing; 0 writes black (off), 1 writes white (on), 2 applies the exclusive-OR function to what is on the screen
DISP	Sets the currently displayed graphics page
PLOT	Writes one pixel to the screen
GETPT	Gets one pixel from the screen
MOVE	Sets an endpoint to be used for line drawing
DLINE	Draws a line from the point set by the last MOVE to the current point
BLKFIL	Fills a rectangle according to the currently set intensity
TEXT	Puts a character or characters on the screen
ARC	Draws a quarter circle
CIRCLE	Draws a complete circle
FILL	Fills in an irregular polygon

Table 1: *The graphics primitives supplied with the Hercules card.*

At a Glance

Name
Hercules Graphics Card

Use
Replaces the IBM monochrome display card; has both 80- by 25-character display and 720- by 348-point graphics display

Manufacturer
Hercules Computer Technology, 3200 Adeline St., Berkeley, CA 94703, (415) 799-9354

Size
Uses one slot on the IBM PC backplane

Features
Text display is indistinguishable from that of the IBM monochrome display; includes a printer port to allow replacement of the monochrome card

Hardware required
IBM monochrome video display; **cannot** be used in conjunction with a color display

Software
BASICA, in order to run HBASIC; can be used with high-level languages as well, although none is provided

Documentation
General description of hardware, low-level graphics interface routines, and BASIC

Price
$499, includes Graph X and HBASIC software

ably need an adapter cable.

Replacing my monochrome board with the Hercules presented no problem because the operations manual contained explicit instructions. The Hercules card worked the first time, pretending to be the standard 80- by 25-character text display. With daily use, however, I have noticed an occasional failure to properly initialize on power-up; specifically, the cursor does not appear as it should, and the system does not boot. Each time this problem occurred, I solved it by turning off the system power switch, waiting a few seconds, and then turning the system back on.

I made working copies of the two Hercules disks so I could try out the graphics functions. A message on the disk envelope warns the user that only *one* backup copy is permitted, although this rule is not enforced by the software. It seems ridiculous to place such an unenforceable restriction on a software package, especially considering that the graphics functions are likely to be used in several ways on the same system. I'm sure that few people will endure the shuffling of floppy disks that's necessary to comply with this rule.

Graphics

The Hercules card provides a set of 15 graphics primitives, which can be called from BASIC, from assembly language, or from other high-level languages. Using these primitives as a starting point, you can write sophisticated programs to manipulate a screen image. The functions supplied are listed in table 1.

The Graph X software manual, which describes the graphics operations, contains excellent examples of the assembly-language interfaces needed to call these functions from an assembler program. An object file can be used with the IBM linker to enable compiled languages to use Graph X. Graph X functions can be called from BASIC—both IBM's BASICA (the Advanced BASIC interpreter provided by IBM for the PC) and compiled BASIC—through a series of steps.

HBASIC

Probably the easiest way to start producing graphics with the Hercules card is using Hercules BASIC (HBASIC). When I first heard about HBASIC, I had visions of trying to keep track of which .BAS files were written for BASICA and which were written for HBASIC. My fears, however, were unfounded. The program called HBASIC is *not* a new BASIC. It loads BASICA and modifies it in memory (not on disk) to do graphics, but it is somewhat different. Because it is slightly slower than BASICA, timing loops must be recalculated. BASICA assumes an individual character size of 8 by 8 pixels, yet HBASIC assumes a matrix of 9 by 14 pixels. Many graphics commands intended for the IBM color/graphics

Text continued on page 352

Listing 1: *This IBM Macro Assembler program contains all the code needed to call any Graph X function. The program was used to generate a series of screens, from which some of the photos in this article were generated.*

```
TITLE Hercules  -- Exerciser for the Hercules Graphics Card
COMMENT *
        Written by Tom Wadlow
        *
video   macro   func            ; Do BDOS Screen output function func
        mov     ah,func
        int     10h
        endm

gmode   macro                   ; Enter graphics mode
        video   40h
        endm

tmode   macro                   ; Enter text mode
        video   41h
        endm

clrscr  macro                   ; Clear the screen
        video   42h
        endm

gpage   macro   bufpage         ; Change page to be written into
        mov     al,bufpage      ; (0 or 1)  Doesn't affect display
        video   43h
        endm

level   macro   i               ; Set intensity level
        mov     al,i            ; 0 - black, 1 - white, 2 - XOR
        video   44h
        endm

disp    macro   bufpage         ; Set current display page
        mov     al,bufpage      ; (0 or 1)
        video   45h
        endm

plot    macro   x,y             ; sets, clears or xors a pixel
        mov     di,x
        mov     bp,y
        video   46h
        endm

getpt   macro   x,y             ; reads a pixel
        mov     di,x
        mov     bp,y
        video   47h
        endm

move    macro   x,y             ; set new endpoint
        mov     di,x
        mov     bp,y
        video   48h
        endm

dline   macro   x,y             ; draws from last endpoint to x,y
        mov     di,x            ; sets new endpoint at x,y
        mov     bp,y
        video   49h
        endm
```

Listing 1 continued on page 350

Listing 1 continued:

```
blkfil  macro   x,y,w,h         ; fills a rectangle w,h whose lower left
        mov     di,x            ; corner is at x,y
        mov     bp,y
        mov     cx,w
        mov     bx,h
        video   4ah
        endm

putchr  macro   x,y,c           ; puts a character at x,y
        mov     di,x
        mov     bp,y
        mov     al,c
        video   4bh
        endm

arc     macro   x,y,r,q         ; Draws a quarter circle centered at x,y
        mov     di,x            ; radius r, in quadrant q  2  1
        mov     bp,y            ;                          3  4
        mov     bx,r
        mov     al,q
        video   4ch
        endm

circ    macro   x,y,r           ; Circle at x,y radius r
        mov     di,x
        mov     bp,y
        mov     bx,r
        video   4dh
        endm

fill    macro   x,y             ; Fill irregular shape where x,y is inside
        mov     di,x
        mov     bp,y
        video   4eh
        endm

wait    macro                   ; Wait for any key to be struck
        mov     ah,0
        int     16h
        endm

SSEG    SEGMENT STACK
        DW      32 DUP(?)
SSEG    ENDS

CSEG    SEGMENT
        ASSUME  CS:CSEG

MAIN    PROC    FAR
        PUSH    DS              ; Save PC-DOS return information
        SUB     AX,AX
        PUSH    AX

; Setup
        gmode                   ; Enter graphics mode
        gpage   0               ; Select the graphics page to write into
        disp    0               ; Select the graphics page to be displayed
        clrscr                  ; Clear the page

; Fill entire screen to determine usable area
        fill    359,173
        wait                    ; Pause for measurement
        clrscr
```

Listing 1 continued on page 352

Listing 1 continued:

```
; Draw a circle
        circ    359,173,100
        wait                        ; Pause for a photo

; Fill it
        fill    359,173             ; Fill the inside of the circle
        wait                        ; Pause for a photo

; Clear the screen and try something harder
        clrscr
        circ    359,173,100         ; Draw a circle like before
        circ    359,173,50          ; and then a smaller, concentric one
        wait                        ; Pause for a photo
        fill    270,173             ; Pick point inside and try to fill the donut
        wait                        ; Pause for a photo

; Draw a square around the circle
        move    254,68              ; Set the starting point
        dline   464,68              ; Draw a square 210 pixels on a side
        dline   464,278             ; centered around the center of the circle
        dline   254,278
        dline   254,68
        wait                        ; Pause for a photo

; Set to XOR mode and fill the circle
        level   2                   ; Set the level to XOR-mode
        blkfil  254,278,210,210     ; Fill the square
        wait                        ; Pause for a photo

; Clear the screen and return to text mode
        clrscr
        tmode

        RET                         ; Return to PC-DOS
MAIN    ENDP
CSEG    ENDS

        END
```

Text continued from page 344:

adapter work under HBASIC, though, and the HBASIC documentation provides a "cookbook" method of converting color BASICA programs to run under HBASIC.

Many of the Hercules card's features cannot be used from interpretive HBASIC, however. Linkages enable a program written in compiled BASIC to use the full set of Graph X routines, and there are easy ways to use Graph X from assembly-language programs and other compiled languages such as Pascal.

I found Graph X remarkably easy to use with the IBM assembler. Listing 1 is an IBM macro assembler program I wrote to generate some of the photographs in this article. The Graph X manual describes in detail the syntax for calling each function from assembly language, Pascal, and BASIC. My only complaint is that some of the examples are inconsistent. On page 13 of the Graph X manual is an example of drawing a circle (the comments on the right are mine).

```
MOV     AH, 4DH     Set the function to CIRCLE (hexadecimal 4D)
MOV     DI, X       Set the x-location of the center
MOV     BP, Y       Set the y-location of the center
MOV     BX, RADIUS  Set the radius of the circle
INT     10H         Call Graph X to draw the circle
```

Every example of individual functions uses a similar form, and the calling sequence is always the same: first, set the function code; second, initialize the registers; and finally, call Graph X.

However, when the time comes to give an example of an entire assembly-language program, the circle-drawing portion of that program is written as follows (the comments on the right are mine):

```
MOV     BX, 120     Set the radius
MOV     DI, 359     Set the x-location
MOV     BP, 173     Set the y-location
MOV     AH, 4DH     Select the CIRCLE function
INT     10H         Call Graph X to draw the circle
```

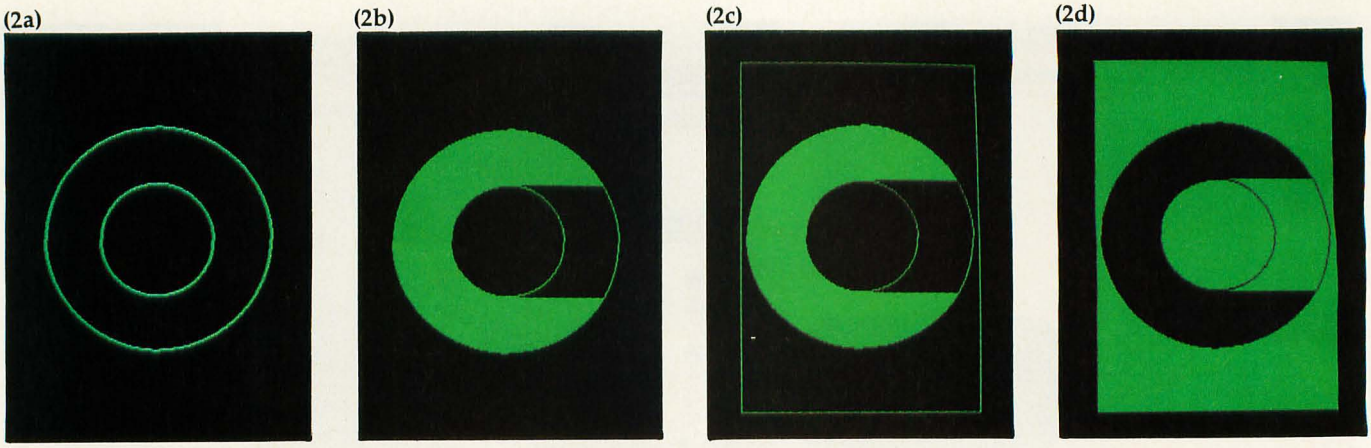

Photo 2: *The concentric circles in 2a were produced with the CIRCLE function. Then, in 2b, FILL (the convex-polygon fill function) was applied. Next, in 2c, the card's line-drawing function produced the rectangle framing the figure. In 2d, the rectangle was colored in with BLKFIL, set at an intensity level of 2 (the XOR function).*

Worse yet, the manual's only comment on those five lines of code was in the first line, simply "draw a circle." Consistency is very important in computer documentation, and I wish there were more in this set of manuals.

Despite the vagueness of the manuals, I experienced no difficulty producing a set of assembly-language and HBASIC programs to exercise the capabilities of the Hercules card. In the process, I found many usable features and three fairly important flaws. Most of the good features as well as the flaws are in the Graph X software and not in the Hercules board, so they can be corrected.

The manufacturer has provided two features, FILL and BLKFIL, that are indispensable in a graphics package. The first of these two features, an irregular-polygon fill (called a convex-polygon fill by the vendor), enables a programmer to select a point inside "an object with no interior holes and no peninsulas protruding into it." The interior of the polygon is filled by applying this function to the selected point, turning a hollow object into a solid one. Using FILL on an empty screen turns the entire screen green; using FILL on an object already filled erases the object entirely. Graph X performs this function with blinding speed. Photos 2a through 2d illustrate some of the capabilities of this software. The graphics were produced using the program in listing 1.

My only complaint about this feature concerns the definition of a convex polygon. There are plenty of algorithms to fill polygons with holes and peninsulas, and it's a shame that Hercules chose not to implement them. The vendor's convex-polygon fill can indeed be used to fill objects with holes, but those holes cast "shadows" that require more filling to repair them.

The other feature I especially like, BLKFIL, is used to fill rectangles in one of three ways. If the intensity value (set with the LEVEL primitive) is set at 0, the specified rectangle is erased. If intensity is set at 1, the rectangle is entirely filled in. If intensity is set at 2, the exclusive-OR (XOR) function alters the rectangle, as shown in photo 2d. I heartily approve of this feature. XOR is a useful graphics tool for preserving information; for example, it enables you to move a cursor nondestructively over a display.

However, BLKFIL is as slow as molasses. I am willing, though unhappy, to accept this slow speed when using the XOR function but not when using BLKFIL ON (1) and BLKFILL OFF (0). No significant computation should be going on during those functions.

Another slight annoyance involves the aspect ratio, the ratio between the usable number of vertical and horizontal dots per inch. This ratio is important when you are trying to produce circles, for instance. The Hercules card provides an aspect ratio of 61/88, a rather unusual number that I expect resulted from trying to squeeze as many lines as possible onto the screen image. As the user's manual points out, this number can be approximated by 2/3, but a circle produced with an aspect ratio of 61/88 is measurably different from one produced with a ratio of 2/3. Nevertheless, most computers multiply by 61/88 as well as they do by 2/3 or 3/4. If you are working in a language such as FORTH, which is inherently integer-based, you may find this discrepancy frustrating. Listing 2, an HBASIC program that generates circles of various aspect ratios, was used to produce photo 3.

Hardware

I can offer very little criticism of the actual Hercules board because, with the exception of the occasional problem on power-up, it performs exactly as advertised. During switches between text and graphics modes, the screen gives a very distressing vertical *bounce,* which is harmless but potentially surprising the first time you see it. The switch between the two graphics pages, however, is as clean and glitch-free as you could possibly want.

A look at the documentation provided on the hardware reveals that an ambitious programmer could do many interesting things with the Hercules card. Mouse Systems has written a version of RasterOp that uses the Hercules; except for a brief demonstration at the 1983 West Coast Computer Faire, I haven't looked at it in detail. What I saw was very impressive, however.

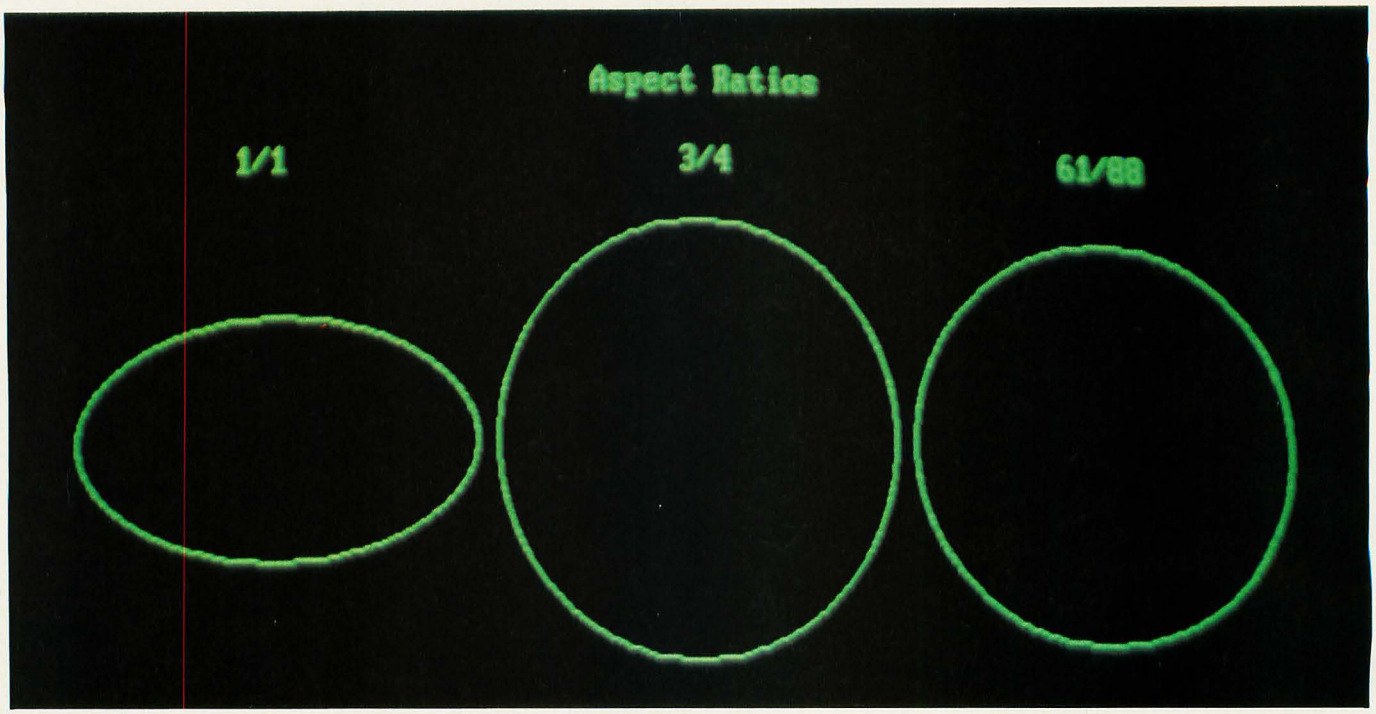

Photo 3: *A comparison of aspect ratios generated with the program in listing 1.*

Listing 2: *This HBASIC program generates circles of various aspect ratios. It was used to produce photo 3.*

```
10 REM ASPECT.BAS -- Demonstrates how an aspect ratio can affect a circle
15 REM Written by Tom Wadlow
20 CLS
30 CIRCLE(150,173),100
40 CIRCLE(360,173),100,,,,3/4
50 CIRCLE(570,173),100,,,,61/88
60 LOCATE 4,35:PRINT "Aspect Ratios";
70 LOCATE 6,15:PRINT "1/1";
80 LOCATE 6,40:PRINT "3/4";
90 LOCATE 6,62:PRINT "61/88";
100 LOCATE 21,15:PRINT "See the difference that 5/88 can make in a circle";
110 LOCATE 22,15:PRINT "The circle on the left (61/88) is correctly round"
120 REM Pause for a photo
130 LOCATE 1,1
140 INPUT " ",A$
```

The printer port on the graphics card has worked flawlessly with my Microprism 480. Hercules includes software that provides a graphics screen dump to the printer, but it works only with Epson printers equipped with Graftrax chips. Because I don't have an Epson, I couldn't test that feature. The screen dump works by replacing the built-in printer handler with a custom Hercules handler. You can do so by placing the commands INT10 and INT5E (for graphics and print handling, respectively) in your AUTOEXEC.BAT file to be executed on rebooting. Some packages, and specifically HBASIC, seem to replace the print handler themselves and thus are oblivious to preloading interrupt handlers.

Conclusion

Despite the flaws in its documentation and software, the Hercules Graphics Card is a fine product. The hardware is well built and the architecture allows for much flexibility in constructing software. The Graph X package can provide users all the power they could want in terms of graphics primitives, and for those users whose tastes run in different directions, other companies are now coming out with more sophisticated software. In addition, some firms are modifying applications packages to use Hercules graphics. The Lotus 1-2-3 Information Manager is one such package. HBASIC is a good introduction to graphics, but people with sophisticated applications will probably want to work toward a compiled language in order to take full advantage of the Hercules Graphics Card.■

Thomas A. Wadlow (POB 2755, Livermore, CA 94550) received a B.S.E.E. degree from Carnegie-Mellon University. He works as an engineer at the Lawrence Livermore National Laboratory.