

A Closer Look at the IBM Personal Computer

Gregg Williams
Senior Editor

What microcomputer has color graphics like the Apple II, an 80-column display like the TRS-80 Model II, a redefinable character set like the Atari 800, a 16-bit microprocessor like the Texas Instruments TI 99/4, an expanded memory space like the Apple III, a full-function uppercase and lowercase keyboard like the TRS-80 Model III, and BASIC color graphics like the TRS-80 Color Computer? Answer: the IBM Personal Computer, which is a synthesis of the best the microcomputer industry has offered to date. It has a

number of interesting features and a few flaws, but it is easily the best-designed microcomputer to date. In this article, I will take a closer look at the IBM Personal Computer, inside and outside.

An Overview

The IBM Personal Computer (photos 1 and 2) is housed in two units, the keyboard and the System Unit. The keyboard (photo 3) has a standard typewriter layout with the addition of a numeric keypad to the right, a set of function keys to the

left, and miscellaneous other keys to bring the total number to 83. It is connected by a coiled cable to the System Unit, which houses the Intel 8088 microprocessor, the 40 K-byte extended Microsoft BASIC in ROM (read-only memory), up to 64 K bytes of dynamic memory, up to two disk drives, a cassette interface, a built-in speaker, and five expansion slots. (Extra dynamic memory cards placed in expansion slots can bring the total up to 256 K bytes.)

Other peripherals include the IBM Monochrome Display (shown in photo 2) and the IBM 80 CPS (characters per second) Matrix

Photo 1: *The IBM Personal Computer System with a non-IBM color monitor.*



Printer (shown with the optional printer stand in photo 1).

What's It Going to Cost?

The IBM Personal Computer is an impressive unit. But how much is it going to cost? Although the component prices in the "At a Glance" text-box look reasonable (the System Unit and keyboard are only \$1265), the price of a usable configuration is somewhat higher. The higher cost is due to a marketing technique called *unbundling*, which is common in the computer industry and a trademark of IBM in particular. When a system is unbundled, components that usually are priced as one are priced separately. In the case of the IBM Personal Computer, the main unit needs one of two video-display adapter cards, a monitor or television set, a cable, and perhaps an external radio-frequency (RF) modulator.

Table 1a shows several sample configurations of the IBM Personal Computer, and tables 1b and 1c show the list prices of comparable Apple II and Radio Shack TRS-80 Model III units with 48 K bytes of memory and one disk drive. The IBM unit is somewhat more expensive than the standard configurations (note that the Apple II Plus is less expensive if you want only 40-column uppercase output). Still, you get a *lot* more for your money.

Video-Display Options

One thing not commonly understood about the IBM Personal Computer is that you must choose between two separate ways of getting video output. The Monochrome Display and Printer Adapter gives high-quality black-and-white output only, while the Color/Graphics Monitor Adapter can produce color graphics or text. Each takes one of the five expansion slots available on the IBM motherboard (called the System Board by IBM). While you could have both kinds of output by using both adapter cards, most people will not want to tie up the extra slot (more on that later).

The monochrome adapter card is most suited to IBM machines that will be used in an office environment only. The adapter card gives you a



Photo 2: The IBM Personal Computer System with the IBM Monochrome Display.



Photo 3: The IBM Personal Computer keyboard unit.

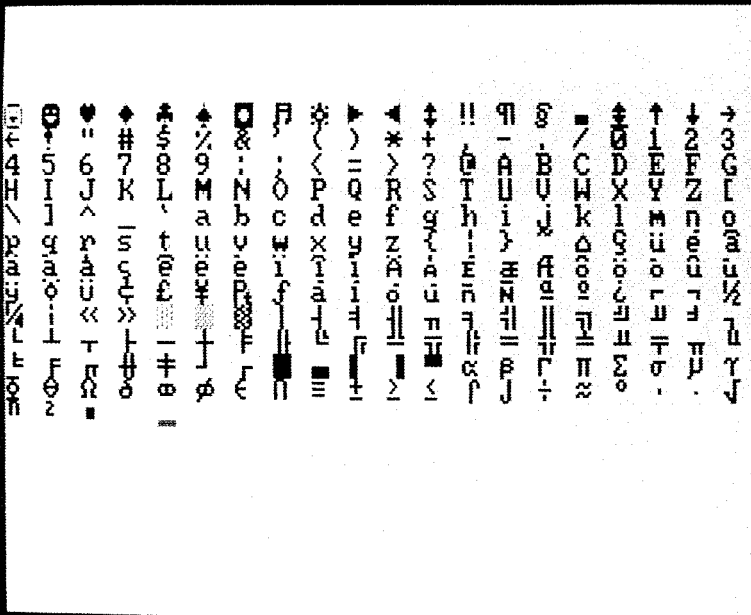


Photo 4: The 256 characters available on the IBM Personal Computer video display.

sharp 25-line by 80-column display with well-formed characters. A 9 by 14 dot matrix is used, and characters are displayed within a 7 by 9 dot matrix; this makes for an extremely readable screen, an important factor if you are using the computer for long periods of time.

Among the 256 characters available are miscellaneous graphics characters (musical note, male and female symbols), all standard upper-case and lowercase letters, numbers, punctuation, some familiar foreign-language, Greek, and mathematics symbols, and a set of rectangular shapes that can be combined to create rectangles and lined tables. A display of the full 256-character set is shown in photo 4.

Although you can use a suitable monitor if you want, the IBM Monochrome Display is also available. The IBM monitor has a green-phosphor tube and matches the appearance of the rest of the system.

The monochrome adapter card contains 4 K bytes of on-board memory. (In this article, 8 bits will be referred to as a "byte," as opposed to a 16-bit "word.") The on-board display memory prevents the available system memory from being steadily decreased by peripheral

cards. In addition, due to the architecture of the 8088 microprocessor, the on-board memory itself does not reduce the main memory address space available to the IBM microcomputer; in contrast, the memory taken by the video display of an 8-bit microcomputer always reduces its 64 K-byte work-space.

The manuals will set the standard for all microcomputer documentation in the future.

Twenty-five lines of 80 characters each amounts to only 2000 characters, yet the on-board display memory has 4096 bytes. The reason for this is that the IBM Personal Computer *always* uses two bytes per stored character, regardless of the adapter card used. When the monochrome adapter card is used, individual characters can have any of the following characteristics: invisible (white-on-white, black-on-black), blinking, high-intensity, or underline. The permissible combinations of these are shown in figure 1.

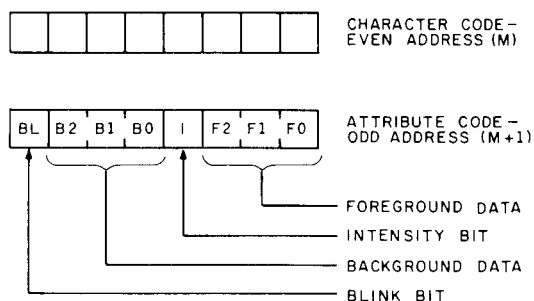
One final advantage of the monochrome adapter card is that it includes an interface to the IBM 80 CPS Matrix Printer, which saves you the expense of an IBM Printer Adapter card (around \$150) and one expansion slot.

Of course, the main disadvantage of the monochrome adapter card is that it does not produce color graphics. As you can see from photos 5a through 5d, this is some disadvantage. The graphics available through the color/graphics adapter card are very good—slightly better than color graphics on existing microcomputers, and they are more versatile and easier to use.

Color/Graphics Monitor Adapter

Residing in one of the five expansion slots in the System Unit, the Color/Graphics Monitor Adapter has 16 K bytes of on-board memory, can display two kinds of text and two (actually, three) kinds of graphics, and allows you to connect to a black-and-white monitor or to a color monitor with composite or RGB (red-green-blue) input, or to a color television. The color pictures accompanying this article were made with a \$1000 RGB color monitor, so don't expect such stunning graphics to come from your composite monitor or an ordinary color television. (RGB monitors are more expensive and produce better images because they have separate red, green, and blue inputs to get a more detailed image. For an RGB monitor to work properly with the color/graphics adapter card, it must accept the following signals: red, green, blue, intensity, horizontal drive, vertical drive, and ground. RGB monitors that do not have an intensity signal can display only 8 of the possible 16 colors.)

Let's consider graphics first. The IBM color/graphics adapter card has three color-graphics resolutions, only two of which are supported by the system software in ROM. The first mode, the IBM low-resolution mode, is unsupported by IBM. It gives you a display of 100 rows and 160 pixels (picture elements), each of which can be any of the standard 16 colors (for the color list, see table 2). IBM



BACKGROUND			FOREGROUND			RESULTING CHARACTER
B2	B1	B0	F2	F1	F0	
0	0	0	0	0	0	NON DISPLAY (BLACK ON BLACK)
0	0	0	0	0	1	NORMAL UNDERLINED CHARACTER
0	0	0	1	1	1	NORMAL CHARACTER
1	1	1	0	0	0	REVERSE CHARACTER (BLACK ON WHITE)
1	1	1	1	1	1	NON DISPLAY (WHITE ON WHITE)

Figure 1: Character storage within the monochrome adapter board.

representatives told me that the only way to use this mode is to directly address the Motorola 6845 CRT Controller, which is at the heart of both the monochrome and color/graphics adapter cards. (For both units, the 6845 device is addressed through two ports: hexadecimal 3D4 and 3D5; more information on this is given in *Technical Reference*, the IBM Personal Computer manual.)

The IBM medium-resolution mode is comparable to what Apple calls its high-resolution mode. It allows 200 rows of 320 pixels each, with four possible colors. (The Apple II allows four colors plus black and white.) The colors are referred to in memory as colors 0 through 3. Color 0 can be any of the 16 colors available, while colors 1 through 3 are set by choosing one of two three-color sets. Set 1 produces cyan, magenta, and white, while set 2 produces green, red, and brown; only the colors from one set are available at any one time. Each byte represents 4 pixels; the mapping scheme is shown in figure 2.

The IBM high-resolution mode uses a white-on-black image and gives you control of 200 rows of 640 pixels each. (Although it is not a well-known fact, the Apple II can display a resolution of 192 by 560 on a black-and-white monitor, although there are some limitations to pixel locations

and the mode must be supported by user-supplied software.) In the IBM high-resolution mode, the mapping of graphics bytes to video scan lines is the same as for medium-resolution graphics, but each byte represents 8 pixels.

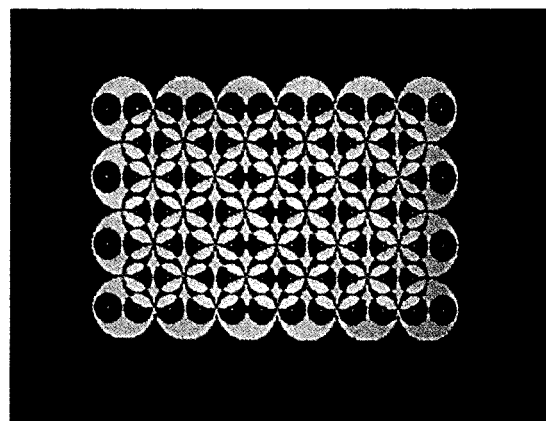
Photos 6a and 6b show one edge of the screen to highlight the differences between IBM medium-resolution and high-resolution graphics. As you would expect, corresponding lines in the IBM high-resolution mode are finer drawn, but I can't see that much difference between the two modes.

The color/graphics adapter card supports two text formats: the first, suitable for color televisions and composite monitors, is 25 rows of 40 characters each, while the second, usable by RGB monitors only, is 25 rows of 80 characters each. The card displays characters in an 8 by 8 dot matrix, with characters being drawn in a 5 by 7 dot matrix.

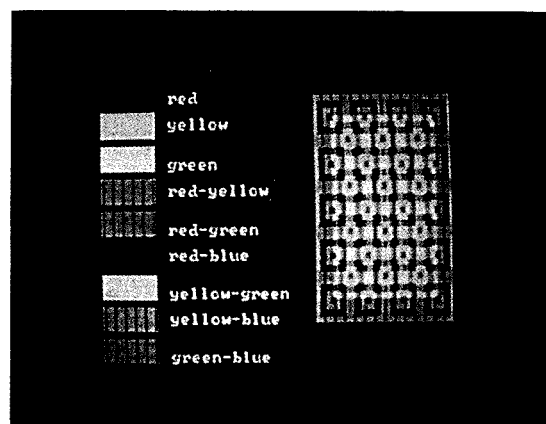
Although the IBM microcomputer has separate text and graphics modes, text can be displayed while in the graphics mode. If you are in graphics mode and want to print text, you simply give the appropriate command (for example, PRINT when in BASIC) and the computer draws the

Photos 5a-5d: Four examples of IBM medium-resolution color graphics.

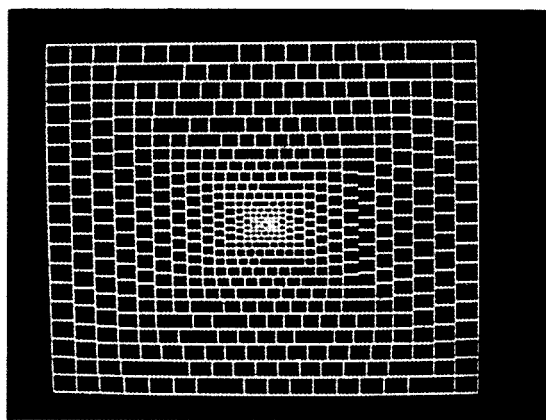
5a



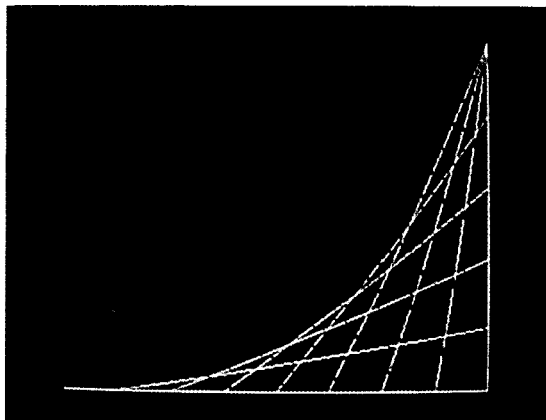
5b



5c



5d



characters on the graphics screen automatically. An example of this is shown in photo 7. While using a text screen, you have access to the same 256-character set used by the monochrome adapter card. If you are using a graphics screen, you have access to only the bottom 128

characters (some symbols, all punctuation, digits, uppercase and lowercase letters). The top 128 characters can be user defined by pointing interrupt vector hexadecimal 1F (contained in hexadecimal memory locations 7C through 7F) to the beginning of a 1 K-byte area that defines the

dot pattern of the top 128 characters, 8 bytes per character.

In the text modes, each character can be one of sixteen colors, with the background of that character being one of eight colors, or the text can be displayed without a color signal (for black-and-white monitors). This is done automatically in BASIC with the COLOR statement. The data that cause a given combination are stored in the attribute byte for each character. Figure 3 shows the layout of the data in the attribute byte, and photo 8 shows an example of multiple background and foreground colors used with text.

Since the color/graphics adapter card has 16 K bytes of memory and the two kinds of text pages take only 2000 and 4000 bytes, respectively, you can store up to four 80-column pages of text or eight 40-column pages at once. In addition, you can specify the display of a page independent of the page actually being written to at the moment. In BASIC, all this is available from the SCREEN statement.

Inside the Main Unit

The IBM Personal Computer is as well designed on the inside as it is on the outside. As shown in photo 9a, the five expansion slots are in the upper left corner, the memory and an internal speaker are in the lower left corner, and the floppy-disk drives (if any) are in the lower right corner. Figure 4 shows the signals on the IBM expansion slot, and table 3 gives the full names of the signals. The bus allows four DMA (direct-memory access) channels, one of which is used to refresh the dynamic memory, the others for high-speed DMA data transfer between memory and peripheral cards. In addition, the bus supports eight levels of interrupts, six of which are available to the user.

The system memory is shown in detail in photo 9b. The set of eight large integrated circuits with gold faces is the 40 K-byte extended Microsoft BASIC in ROM. Notice the empty socket at the bottom of the same row; this can house an 8 K-byte ROM or EPROM (erasable program-mable read-only memory). Just to the

1a IBM Personal Computer (suggested retail prices)

48 K-byte cassette-based unit with color/graphics adapter card	\$1745
all the above, plus one floppy-disk drive, adapter card, and DOS software	\$2575
all the above, plus 16 K bytes more (total, 64 K bytes) and game adapter card	\$2720
all the above, plus a second disk drive	\$3290
all the above, plus one 64 K-byte card (total, 128 K bytes)	\$3830

1b Apple II Plus

48 K-byte Apple II Plus with one floppy-disk drive and DOS software	\$2175
all of the above, plus Videx Videoterm and Enhancer II (to modify Apple for 80-column display and upper- and lowercase keyboard)	\$2788

1c Radio Shack TRS-80 Model III

48 K-byte unit with one floppy-disk drive and DOS software	\$1995
--	--------

Table 1: Prices for several versions of the IBM Personal Computer and roughly comparable Apple II Plus and Radio Shack TRS-80 Model III microcomputers. The versions to be compared are shaded.

Intensity	Red Bit	Green Bit	Blue Bit	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	Light Gray
1	0	0	0	Dark Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	White

Table 2: The 16 available colors on the IBM Personal Computer, and their representation in memory. When only the first eight colors are available (intensity=0), they can be represented with only the bottom three bits.

right of the ROMs are four rows of 4116 dynamic memory rated with an access time of 250 ns. Only the first row is filled in a 16 K-byte IBM microcomputer; successive rows are filled to bring the microcomputer to 64 K bytes before additional memory is added through the expansion slots.

Notice that there are nine integrated circuits per row. The device on the extreme left is used as a parity bit. To increase the reliability of the system, IBM has made all user memory (i.e., all the memory used for programs and data) 9 bits wide. When a parity error is detected, the IBM microcomputer issues the appropriate error message and stops whatever program is running; this prevents an application program from continuing if it has read the memory incorrectly.

In the middle of the right half of the board are two DIP (dual inline package) switches that set certain parameters of the system. The positions of these switches tell the IBM microcomputer how many disk drives are installed, what kind of video device is attached, and how much memory is in the system. These switches are usually hidden by the

floppy-disk-drive cables, as shown in photo 9a.

Photo 9c shows the Intel 8088 microprocessor (the large device in the center) and, above it, an integrated circuit socket identified by IBM only as an "auxiliary processor socket." An IBM representative would only say that the slot could house "any architecturally compatible processor," but it is probable that the device to go in that slot is an Intel 8087, a mathematics coprocessor device. With the appropriate software, the Intel 8087 or something similar could improve the performance of the IBM microcomputer.

Photo 9d shows one of the IBM peripheral cards, the 64KB Memory Expansion Option. This card is interesting in that it uses two modified 4116 16 K-bit dynamic memory devices "piggybacked" into each 18-pin socket. IBM was buying a lot of these two years ago—now we know where they went.

The Intel 8088 itself is functionally equivalent to the 16-bit Intel 8086 microprocessor, except that all 16-bit input/output (I/O) is done 8 bits at a time, with the help of a few extra support devices. Even though the 8088

has the same instruction set as the 16-bit 8086 microprocessor, the necessity of funneling all data through an 8-bit path degrades the 8088's performance to the point

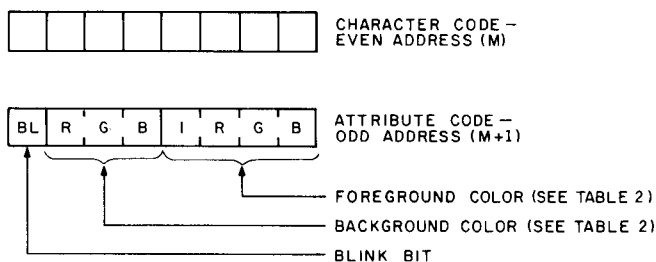


Figure 2: IBM medium-resolution-graphics storage within the color/graphics adapter board.

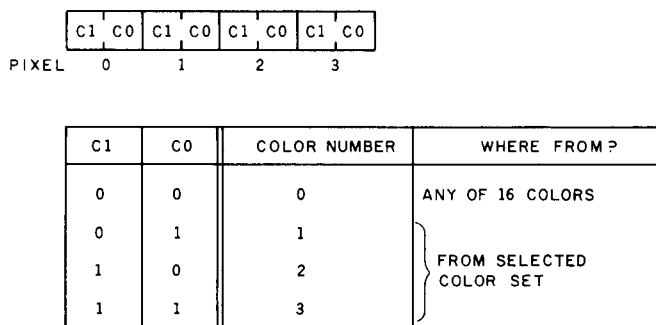


Figure 3: Character storage within the color/graphics adapter board.

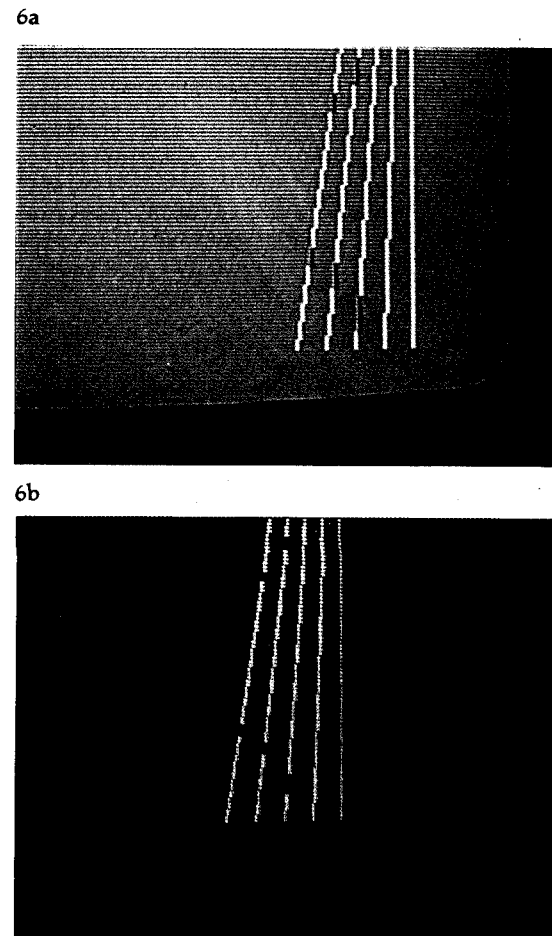


Photo 6: Close-up views of equivalent screen images using IBM medium-resolution (photo 6a) and high-resolution (photo 6b) graphics.

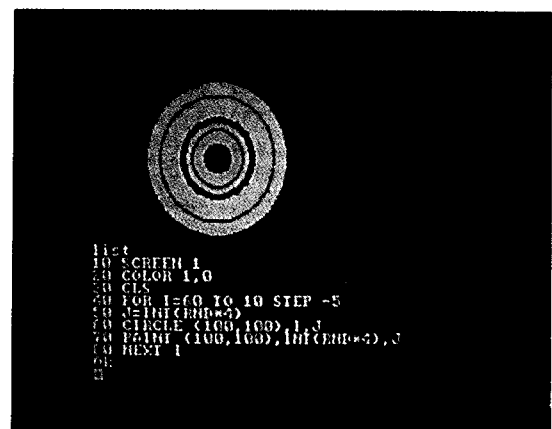


Photo 7: An example of combining text and graphics on the same video screen. The program shown, when run, generates the circular image just above it.

Signal Name

Description

OSC	14.31818 MHz oscillator signal
CLK	4.77 MHz system clock
RESET DRV	reset driver; resets system logic
A0 through A19	address bits 0 (low) through 19 (high)
D0 through D7	data bits 0 through 7
ALE	address latch enable
I/O CH CK	I/O channel check
I/O CH RDY	I/O channel ready
IRQ2 through IRQ7	interrupt request 2 (highest priority) through 7 (lowest)
IOR	I/O read command line
IOW	I/O write command line
MEMR	memory read command line
MEMW	memory write command line
DRQ1 through DRQ3	DMA request 1 through 3
DACK0 through DACK3	DMA acknowledge 0 through 3
AEN	address enable
T/C	terminal count

Table 3: Signal names and descriptions for the IBM Personal Computer System Board I/O Channel (expansion slot). See also figure 4.

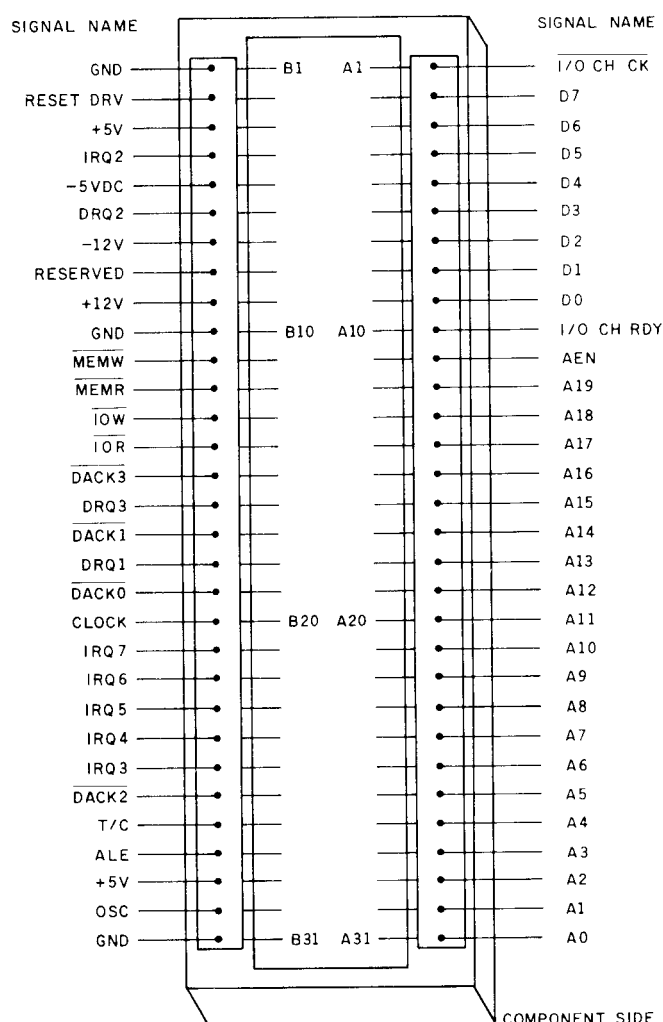


Figure 4: Electrical signals on the IBM System Board I/O Channel (expansion slot). See table 3 for signal descriptions.

where it is more like a fast 8-bit microprocessor with an extended instruction set than it is a 16-bit microprocessor. After all, how much processing can you do on a number without accessing memory again?

Still, the IBM microcomputer combines the architecture of a 16-bit machine with the cost advantages of using familiar 8-bit memory and system design. The 8088 microprocessor in the IBM microcomputer runs at 4.77 MHz.

The disk drives are soft-sectored, double-density, single-sided drives that use MFM (modified frequency modulation) encoding. The floppy-disk drive uses 40 tracks per disk, with eight 512-byte sectors per track. This results in 163,840 bytes of storage per drive. The drive has a motor-start time of 500 ms, a track-to-track seek time of 8 ms, and a data transfer rate of 250 K bits (not bytes) per second.

The IBM Personal Computer includes a cassette-recorder interface that connects to any good-quality cassette recorder through a user-supplied cable. The IBM microcomputer can be configured to use either the microphone or the auxiliary input of the recorder by changing a jumper on the bottom of the main printed-circuit board in the System Unit. The data-transfer rate is between 1000 and 2000 bits per second (bps), depending on the content of the data. The signals used to control a cassette recorder are motor control, ground, data in, and data out.

The right side of the back panel of the main unit (photo 10) contains whatever sockets are made available by the peripheral cards plugged into the expansion slots. Unused slots are masked by metal plates to prevent the escape of any RF radiation. The bottom left corner of the panel contains the power plug to the IBM Monochrome Display and the plug for the main power supply. In the bottom center of the panel are 5-pin DIN plugs that go to the keyboard (left) and the cassette tape recorder (right).

The Keyboard

The keyboard (see photo 3) is one

of the most important components of any computer because it is the primary device through which you give instructions to the computer. Most existing microcomputers have something wrong with their keyboard design; the most common errors are functions unavailable from the keyboard and poor keyboard layout. With one exception, the IBM keyboard seems to be faultless. It is, bar none, the best keyboard on any microcomputer.

The IBM keyboard abounds with good features. The keys have a nice "feel" to them and give tactile and audible feedback when pressed. The keyboard is a separate unit that can be placed up to several feet away from the main unit. It is light enough to rest and use in your lap. The keys themselves are "sculpted"—that is, an imaginary plane touching all the key tops has a slight concave curve to it. The keyboard has two plastic feet that can be used to tilt it up when it is used on a flat surface. A plastic ledge just above the top row of keys can be used to prop an open book between the video display and the keyboard.

Several keyboard features deserve more description. The right side of the keyboard contains a numeric keypad that doubles, in certain situations, as a set of text and cursor-movement keys. The left side contains ten function keys, whose functions can change with the application. (The twenty-fifth line of the video display can be used to illustrate their current use, and you can redefine these keys at any time from BASIC.)

Three keys must be pressed simultaneously to restart the system: Ctrl, Alt, and Del; it takes two hands to do this. Depressing the Ctrl and Scroll Lock/Break keys interrupts a running BASIC program. The up-arrow (shift) and PrtSc keys cause the text contents of the video display to be printed. Ctrl plus Num Lock causes the executing BASIC program to pause; the next key pressed causes it to resume.

The Alt key lets you generate any extended ASCII value from 1 to 255, even if that code is not otherwise generated by the keyboard. By holding down the Alt key and typing

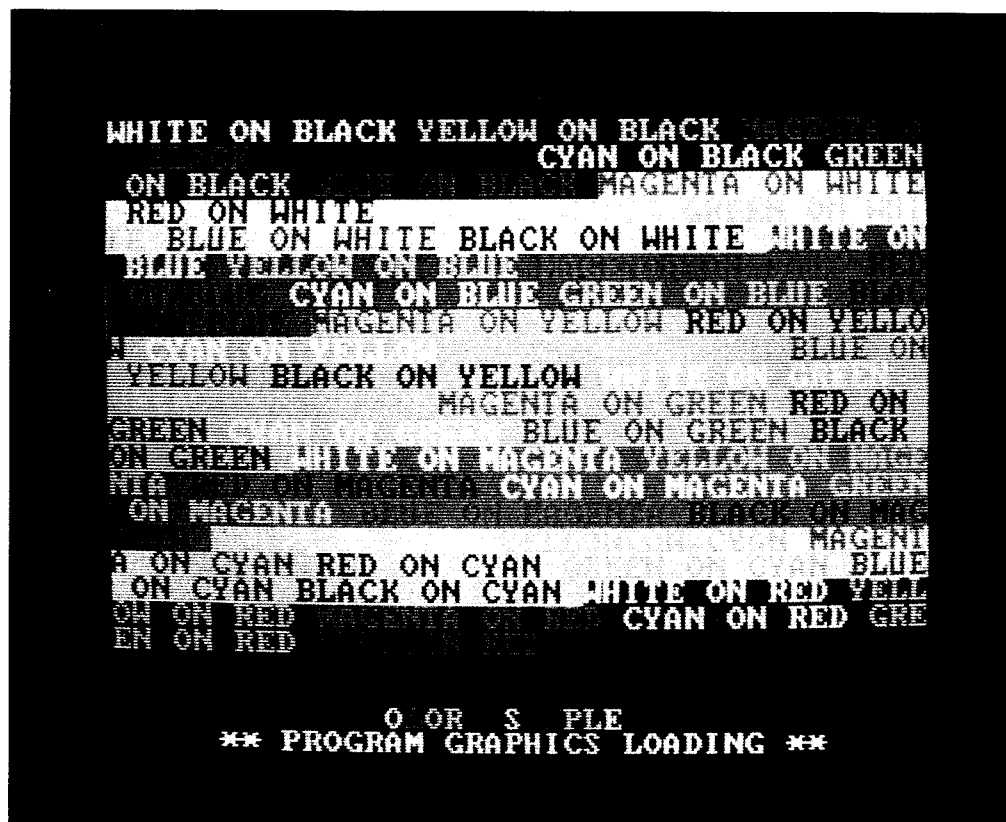


Photo 8: An example showing the independence of foreground and background colors when using the text mode of the color/graphics adapter board.

a number between 1 and 255 on the numeric keypad, that code is generated when the Alt key is released. (For some reason, the IBM unit I tried would not generate 0 with the Alt key. However, 0 could be generated by Ctrl plus the 2 key on the top row of the keyboard.)

By not having a full product line, the IBM Personal Computer may fall prey to hardware and software incompatibility.

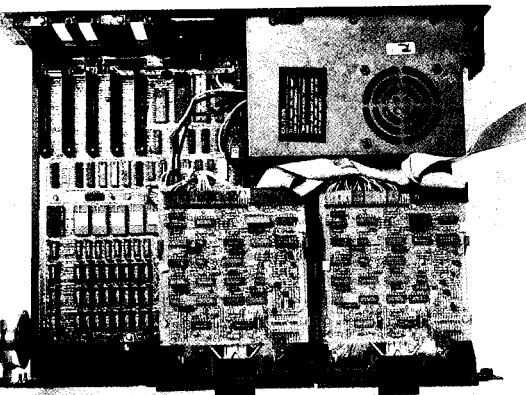
Another nice feature of the IBM keyboard is its 10-character type-ahead buffer, which keeps you from losing keystrokes if you type information into the IBM microcomputer before it is ready to receive it. In addition, the system software is written such that every key has an auto-repeat feature; i.e., every key repeats its function if it is held down for more than half a second.

My one complaint against the keyboard is minor. The right and left shift keys are one key farther away from the center of the board than most people are used to. This means that, until you get used to reaching for the shift keys, you will accidentally type the slash and reverse-slash keys instead. This problem is minor, however, compared to some of the gigantic mistakes made on almost every other microcomputer keyboard. The IBM Personal Computer is a delight to use largely because of its keyboard.

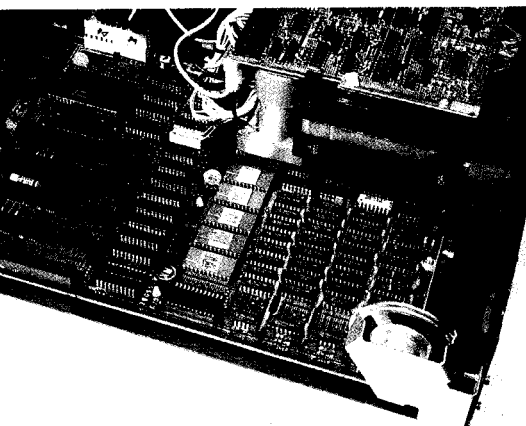
System Startup

When the IBM Personal Computer is first turned on, a series of fourteen tests are performed on the system, and any errors are reported immediately. These include tests of the 8088 microprocessor, the internal ROM, the main memory, the video-display adapter card, the keyboard, the cassette recorder (if attached), and the floppy-disk system. The memory test includes five different read/write tests of the entire user-memory area, each using a different

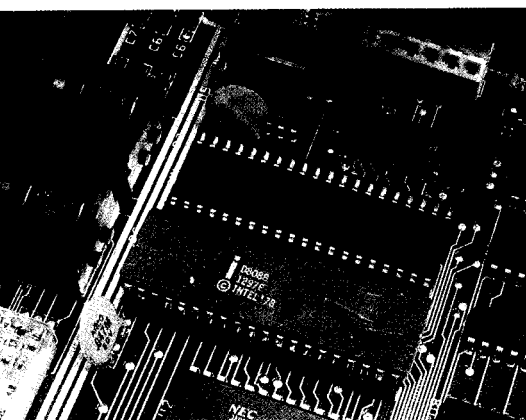
9a



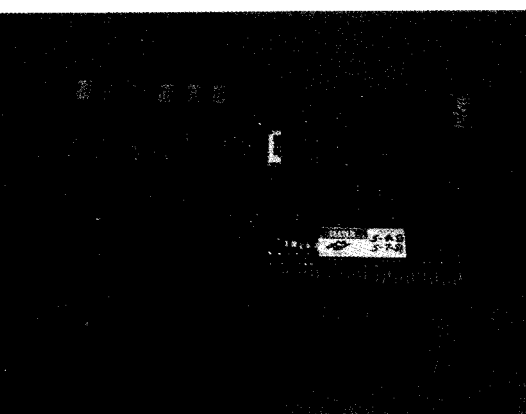
9b



9c



9d



bit pattern for testing. Because of this, the initial startup of the IBM microcomputer may take between 5 seconds and about 1.5 minutes, depending on how much memory is in the machine. For example, in my test a 64 K-byte, disk-based machine took about 18 seconds to complete its initial tests and about 25 seconds more to complete the bootup of the machine. When the system is restarted from the keyboard with the Ctrl-Alt-Del triad of keys, the system tests are omitted, thus greatly reducing the delay associated with rebooting.

Three Levels of BASIC

Because BASIC is the most commonly used programming language, I plan to describe some of the features of the 40 K-byte extended Microsoft BASIC in great detail. But before I

start talking about the more prominent features of BASIC, I'll look at the three kinds of BASIC that are available with the IBM Personal Computer.

Cassette BASIC is the simplest BASIC you can get. It is available on *every* IBM microcomputer, and it is contained in the 40 K bytes of ROM mentioned before. In addition to the standard features that are associated with Microsoft BASIC, Cassette BASIC gives you the ability to plot points and draw lines in both the IBM medium- and high-resolution modes, to make sounds through the internal speaker, and to use light pens and joysticks.

Disk BASIC, which requires at least 32 K bytes of memory and one floppy-disk drive, occupies extra user memory which can be added to the ROM version of BASIC. The IBM

Command

Description

AUTO	generates line numbers automatically
BLOAD	load machine-language (binary) program
BSAVE	save machine-language program
CLEAR	clear program variables
CONT	continue interrupted program
DELETE	deletes a range of BASIC lines
EDIT	edit a line of BASIC
FILES	list all or selected files on disk
KILL	delete a disk file
LIST	list BASIC lines
LLIST	list BASIC lines to printer
LOAD	load a BASIC program file
MERGE	merge a BASIC program file into an existing program
NAME	rename a disk file
NEW	erase current program
RENUM	renumber BASIC program
RESET	close all disk files
RUN	load and run program
SAVE	save current program
SYSTEM	exit BASIC and return to DOS
TRON, TROFF	turn tracing option on and off

Table 4a: A summary of IBM BASIC commands.

Photo 9: Inside the IBM Personal Computer System Unit. Photo 9a shows the overall interior of the unit; the floppy-disk drives are in the lower right corner, and the expansion slots are in the upper left. Photo 9b is a detailed shot of the expansion slots (left), the BASIC in ROM (large devices with gold faces, center), and the workspace memory (right). Photo 9c shows the Intel 8088 microprocessor (bottom) and the empty "auxiliary processor socket" (just above the 8088). Photo 9d shows the IBM 64KB Memory Expansion Option card, which holds 64 K bytes of memory. Notice that two 4116-like devices are "piggybacked" into one socket.

DOS (disk operating system) takes 12 K bytes of user memory, and disk BASIC adds about another 12 K bytes (depending on certain options). Disk BASIC adds a large number of disk input and output options, access to a date and time-of-day clock, the ability to store and redraw rectangular areas of graphic images, communications support using a standard RS-232C port, and software support for two extra printers. Disk BASIC is called by typing "BASIC" from the DOS prompt.

Advanced BASIC, which requires at least 48 K bytes of memory and one floppy-disk drive, occupies an additional 5 K bytes of user memory (for a total overhead of about 29 K bytes). Advanced BASIC adds event trapping, some advanced graphics commands, and an advanced music-playing command (all of these are covered in greater detail later). Advanced BASIC is called by typing "BASICA" from the DOS prompt.

Tables 4a, 4b, and 4c list the commands, statements, and functions of IBM BASIC.

The BASIC Program Editor

The BASIC Program Editor, common to all the versions of IBM BASIC, allows you to make changes far more quickly and easily than is possible on other microcomputers. It is a full-screen editor in that changes can be made to a program line by use of the four arrow keys and the Ctrl (control), Ins (insert), Del (delete), and End keys. If the new line (enter) key is pressed while the cursor is *anywhere* on the program line where changes have just been made, the changed line takes the place of the old line. With the BASIC Program Editor, changing a program is as easy as it would be if the text of the program were being manipulated by a word processor.

In addition, the Alt key has a special function within BASIC. Simultaneously pressing Alt and a letter of the alphabet causes a pre-defined BASIC keyword to be printed on the screen. For example, Alt plus C causes the word "COLOR" to be printed. This "shorthand" method is often helpful when you are typing in

a long BASIC program.

Along the same lines, all levels of IBM BASIC have the AUTO (automatic line numbering), RENUM (re-number a BASIC program), and MERGE (merge two programs) commands—all very useful commands that are often absent or awkward to use in other microcomputers.

Graphics in BASIC

The following summarizes most of the graphics commands available from BASIC:

- **COLOR** (all BASICs) is used to choose the four colors available in the IBM medium-resolution mode. As stated before, color 0 can be any of

At a Glance

Product Name

The IBM Personal Computer

Manufacturer

International Business Machines Corporation
Information Systems Division
Entry Systems Business
POB 1328
Boca Raton, FL 33432

Components

System Unit

Size: width 20 inches, depth 16 inches, height 5.5 inches; weight (without disk drives) 21 pounds, (with two disk drives) 28 pounds

Electrical needs: 120 VAC

Processor: Intel 8088

Cycle Time: main storage, 410 nanoseconds; access, 250 nanoseconds

Memory: 40 K bytes of built-in ROM (read-only memory),

16 K bytes of user RAM (random-access read/write memory); expandable to 256 K bytes

Standard: keyboard for data and text entry; audio-cassette recorder connector; five expansion slots for memory, display, printer, communications, and game adapters; built-in speaker for music programming; power-on automatic self-test of system components; BASIC-language interpreter; 16 K bytes of user RAM (all user RAM includes parity bit)

Keyboard: total of 83 keys for data and text entry; includes 10 keys for numeric entry and cursor control, 10 special function keys, and ASCII characters and special graphics characters (total 256 characters); automatic repeat on all keys; adjustable typing angle; detachable six-foot coil cable

Disk drives: up to two 5-inch floppy-disk drives, 160 K bytes each (will accommodate 4 drives in future)

Operating Systems

IBM Personal Computer DOS (Microsoft)

Software Available for IBM Personal Computer DOS

BASIC interpreter (Microsoft) standard; extended BASIC interpreter (Microsoft) \$40; Pascal compiler (Microsoft) \$300; VisiCalc (Personal Software) \$200; EasyWriter (Information Unlimited Software) \$175; General Ledger, Accounts Receivable, Accounts Payable (Peachtree Software) \$595 each; asynchronous communications support \$40; Adventure (Microsoft) \$30; Advanced diagnostics package \$155

Hardware Prices

System Unit, 16 K-byte RAM, keyboard	\$1265
System Unit, 48 K-byte RAM, keyboard, single floppy-disk drive, disk-drive adapter	2235
Monochrome video display	345
Combination monochrome-display adapter and printer adapter	335
Color-graphics-monitor adapter	300
16 K-byte memory-expansion kit	90
32 K-byte memory-expansion kit	325
64 K-byte memory-expansion kit	540
Disk-drive adapter	220
Disk drive (5-inch floppy disks)	570
Asynchronous communications adapter	150
Game-control adapter	55
Keyboard	270
Printer	755
Printer adapter	150
Printer cable	55
Printer stand	55

the 16 available colors, while colors 1 through 3 are chosen from two available color sets. (In the text mode, COLOR sets the foreground, background, and border colors.)

●LINE (all BASICs) allows you to draw a line, outline a box, or fill in a box in whatever colors are available.

●SCREEN (all BASICs): "SCREEN *n*" is used to choose text mode (*n*=0), IBM medium-resolution graphics (*n*=1), or IBM high-resolution graphics (*n*=2). In text mode, SCREEN can also generate a black-and-white text image and choose which of several pages are to be independently viewed and written to.

●GET and PUT (disk BASIC): GET allows you to save the graphic image within a specified rectangular area into a BASIC array. PUT allows the stored image to be redrawn at any point on the screen, in one of five ways: PSET (replace the existing image with the stored image), PRESET (replace with the inverse of the stored image), XOR (exclusive-or the existing and stored images), OR (add the stored image to the existing image), AND (selectively restore the stored image, using the existing image as a logical mask).

●CIRCLE (advanced BASIC) allows you to draw a circle or ellipse with a given center, color, radius, and eccentricity. In addition, an arc (partial circle) may be drawn (the begin and end points of the arc can be specified); optionally, either or both endpoints can be connected to the center point. (This last feature exists but is not documented in the description of the CIRCLE command in the IBM BASIC manual. The end points must have an absolute value less than or equal to 2π . The same arc is drawn regardless of the sign of the end point; if the end point is negative, however, it is connected to the center point.)

●PAINT (advanced BASIC) lets you specify a starting point, a color, and a boundary color. PAINT then starts painting the screen the given color from the starting point outward until it reaches the boundary color.

Subroutine Interrupts in BASIC

A very unusual and useful feature of the IBM BASIC is its ability to stop

Statement	Description
BEEP	beep the internal speaker
CALL	call machine-language subroutine from BASIC
CHAIN	execute a new program, retaining values of program variables
CIRCLE	draw circle, ellipse, arc, or pie-shaped wedge
CLOSE	close data file
CLS	clear video screen
COLOR	set foreground and background colors
COM...ON/OFF/STOP	enable/disable activation of ON COM...GOSUB
COMMON	mechanism to pass variables to CHAINED program
DATA	standard DATA statement
DATE\$	set date
DEF FN...	user-defined function
DEF SEG	define current segment of memory
DEF USR	define starting address for USR call
DIM	dimension arrays
DRAW	draw a graphics command string
END	end program
ERASE	reclaim memory from arrays no longer being used
ERROR	simulate a given error condition
FIELD	defines fields within a file buffer
FOR...TO...STEP	standard FOR loop
GET (disk I/O)	get a record from a random-access file
GET (graphics)	put graphics information from screen to array
GOSUB	execute subroutine
GOTO	continue execution at specified line
IF...THEN...ELSE	standard IF statement
INPUT	read data from keyboard or data file
KEY ON/OFF	turn display of function keys on 25th line on or off
KEY	redefine one of ten function keys
KEY...ON/OFF	enable/disable activation of ON KEY GOSUB
LET	standard assignment statement (e.g., LET A = 3)
LINE	draw line, box, or solid box on graphics screen
LINE INPUT	read an entire line from keyboard or data file
LOCATE	position cursor
LPRINT	print to printer
LPRINT USING	print to printer according to a given format

Table 4b: A summary of IBM BASIC statements.

execution of a BASIC program to service an external interrupt before continuing the BASIC program. What makes this interrupt capability different from that of any other micro-computer is that the interrupt routine is *not* a machine-language program but a BASIC subroutine within the BASIC program being used. The interrupting events are: a keypress from any of the four cursor-movement keys or the ten function keys, incoming information from the IBM Asynchronous Communications

Adapter card, activation of the light pen, or a keypress from a joystick trigger button.

The form of these statements is

ON *event* GOSUB *line*

where *event* is COM*n*, KEY(*n*), PEN, or STRIG(*n*) (joystick trigger), and *line* is the beginning line number of a BASIC subroutine. Another condition for the execution of the subroutine is for the event to be *activated*, which is done by an

Statement	Description
LSET	left-justify a string within a field
MID\$	substring substitution statement
MOTOR	control cassette recorder motor
NEXT	ends FOR loop
ON COM/KEY/PEN/ STRIG...GOSUB	interrupt by given event to BASIC subroutine (see text for details)
ON ERROR GOTO	enable error-trapping routine
ON...GOSUB	standard computed GOSUB statement
ON...GOTO	standard computed GOTO statement
OPEN	open a disk or communications file
OPTION BASE	allows array subscripts to start at 0 or 1
OUT	output a byte to a port
PAINT	fill an area of the graphics screen with color
PEN ON/OFF/STOP	enable/disable activation of ON PEN GOSUB
POKE	put a specified value into a byte
PRINT	print to video display or file
PRINT USING	print to video display or file according to a given format
PRESET	plot a graphics point in the background color
PSET	plot a graphics point in a given color
PUT (disk I/O)	write a record to a random-access file
PUT (graphics)	draw a stored image onto the graphics screen
RANDOMIZE	start a new pseudo-random number sequence
READ	read information from DATA statements
REM	standard remark statement
RESTORE	reset pointer to DATA statements
RESUME	return from an error routine
RETURN	return from a subroutine
RSET	right-justify a string within a field
SCREEN	choose text or graphics screen for video display
SOUND	generate sound from the speaker
STOP	stop program execution
STRIG ON/OFF	enable/disable joystick button
STRIG...ON/OFF	enable/disable activation of ON STRIG...GOSUB
SWAP	exchange the values of two variables
TIMES	set time
WAIT	standard Microsoft WAIT statement
WEND	end WHILE loop
WHILE	program loop that executes as long as a given condi- tion is true
WRITE	output data to video screen or file

associated set of BASIC commands. For example, if the statement

PEN ON

is executed and the ON PEN statement exists in the program, the subroutine will be executed the next time the light pen is used. If

PEN OFF

is executed, the use of the light pen will *not* cause the subroutine to be executed. If the statement

PEN STOP

is executed, using the light pen causes the subroutine to be executed as soon as a PEN ON statement is executed. Similar statements are available for COM n and KEY(n), but not for STRIG(n).

With these statements, a program can immediately respond to certain events that may or may not happen.

DRAW and PLAY

One of the most innovative features of the IBM BASIC is the use of predefined BASIC strings to specify a series of draw commands (for DRAW) or note-playing com-

mands (for PLAY). These strings have their origins in the Apple II shape tables; but, by extending the syntax and allowing the "table" to take the form of standard strings that can be manipulated by the BASIC program itself, the concept has been greatly improved.

Table 5 lists the commands available within a DRAW string. To draw a long, narrow rectangle, we simply define

AS="R40;U10;L40;D10"

This draws 40 units to the right, 10 up, 40 to the left, and 10 down. If we execute the statement

DRAW AS

the rectangle will be drawn from wherever the graphics cursor happens to be at that time.

One of the most powerful features of this graphics-command language is the ability to call one string from another. For example, to rotate this box 90 degrees counterclockwise, we could simply command

DRAW "A1;XA\$;"

(A1 calls for a 90-degree rotation, and XA\$; executes string AS.) In addition, any command can take its argument from an existing variable, so that if we say

DRAW "A=I;XA\$;"

the image will be rotated an I-multiple of 90 degrees before being drawn. Note the presence of the semicolon at the end of the X command; this is necessary for the command to work.

Photo 11 shows the listing and the run of a program that first draws the string AS, then draws it in all its rotations. The PSET statement simply moves the graphics cursor to a new location before drawing.

The PLAY statement works similarly to the DRAW statement, but with a different set of commands. For example, the statement

PLAY "C1;D#2;G-4"

Function	Description
ABS	absolute value
ASC	convert ASCII character to its numeric value
ATN	arctangent
CDBL	convert to double-precision number
CHR\$	converts number to equivalent ASCII character
CINT	round to nearest integer
COS	cosine
CSNG	convert to single-precision number
CSRLIN	returns row number of current cursor position
CVD	convert string to double-precision number
CVI	convert string to integer
CVS	convert string to single-precision number
EOF	logical test for end-of-file condition
ERL	line number of an error that has just occurred
ERR	error code of an error that has just occurred
EXP	exponential function, base e
FIX	truncate to an integer value
FRE	amount of workspace left unused
HEX\$	converts number to a string containing a hexadecimal number equivalent to the original number
INKEY\$	read a character from the keyboard
INP	read 8-bit value from port
INPUT\$	read characters from a file
INSTR	find substring within a given string
INT	largest integer less than or equal to argument
LEFT\$	take substring starting with first character
LEN	length of a string
LOF	amount of space in a file
LOG	natural logarithm
LPOS	carriage position of printer
MID\$	extract a substring from a given string
MKD\$	convert a double-precision number to a string
MKI\$	convert an integer to a string
MKS\$	convert a single-precision number to a string
OCT\$	converts number to a string containing an octal number equivalent to the original number
PEEK	read value of byte in memory
PEN	read light pen
POINT	get color number point on graphics screen
POS	cursor column position
RIGHT\$	take substring ending with last character
RND	random number
SCREEN	character or color at given position (text mode only)
SGN	sign of argument
SIN	sine
SPACE\$	creates a string full of spaces
SPC	prints spaces
SQR	square root
STICK	get coordinates of joystick
STR\$	converts a number to a string
STRING\$	creates a string filled with one ASCII constant
TAB	spaces over to an absolute print position
TAN	tangent
USR	call machine-language subroutine
VAL	converts string to numeric value
VARPTR	get address of variable; or get file control block of a file

Table 4c: *A summary of IBM BASIC functions.*

plays a whole C note, a half D-sharp note, and a quarter G-flat note. Many variations are possible, including octave and tempo change, note length, pauses, substring execution, and variable command parameters. In addition, a sequence of up to 32 notes can be stored in a buffer and played in background—that is, the BASIC program continues to execute, and the music is played independently by the buffer.

Communications Files

If the IBM Asynchronous Communications Adapter is installed in the IBM Personal Computer, a BASIC program can interact with a remote device as if it were a simple disk file. GET and PUT can be used, as well as the I/O statements INPUT $\#f$, LINE INPUT $\#f$, INPUT\$, PRINT $\#f$, PRINT $\#f$ USING, and WRITE $\#f$. In all these cases, f is a file specification that has a device name of COM1: or COM2:. Thus more people can write programs that use remote devices, because BASIC automatically takes care of most of the communication details.

The IBM DOS

The IBM disk operating system (DOS) (written by Microsoft with help from Seattle Computer Products) bears a superficial resemblance to Digital Research's CP/M operating system. (For example, the IBM DOS gives the prompt "A>".) However, the IBM DOS is a scaled-down version of Microsoft's 16-bit Unix look-alike, the Xenix operating system. In addition, the commands are better worded than in CP/M. For example, the cryptic

```
PIP B:NEWFILE1=A:MYFILE1
```

of CP/M is replaced by

```
COPY A:MYFILE1 B:NEWFILE1
```

which copies MYFILE1 from drive A to drive B, where it will be named NEWFILE1. Other commands include ERASE (to delete a file), FORMAT (to format a floppy disk), RENAME (to rename a file), DIR (to list the directory of a disk), DATE (to set the

date, and TIME (to set the system clock).

The IBM DOS floppy disk contains BASIC and BASICA (the disk and advanced BASICs), as well as some disk utility programs. LINK combines object files (created by an assembler or compiler) into a form that can be executed. DEBUG allows you to examine both memory and disk files and debug a machine-language program. Photo 12 shows the DEBUG program tracing the execution of a program and displaying all the 8088 registers.

Another feature of the IBM DOS is the file AUTOEXEC.BAT. If a disk file with this name is present on the disk used to start the system, it is automatically executed as soon as the

IBM microcomputer is working. The ".BAT" suffix marks it as a *batch file*, which is a text file of statements that are executed sequentially as if they had been typed in from the keyboard in a manner similar to a CP/M submit (.SUB) file or an Apple II EXEC file. Because the AUTOEXEC.BAT file is a batch file, it can perform many operations before giving control to the user.

The IBM BIOS

All software interacts with the hardware of the IBM microcomputer through part of the DOS called the BIOS (basic input/output system). In the IBM microcomputer, all calls to the BIOS are done as 8088 software interrupts. There are 256 such inter-

rupts available on the 8088, of which 193 are used by DOS and BASIC.

BASIC uses many of the reserved interrupts to interact with the rest of the machine. By using the interrupts as "hooks" to the actual routines, which are stored in high memory (see table 6), the system can add new devices and change the behavior of existing ones by writing new routines in user memory and changing the appropriate interrupt vectors to point to the new code. In fact, this is how the disk and advanced BASICs add features to the cassette BASIC in ROM. In the same way, a programmer with sufficient skill can extend the behavior of the IBM Personal Computer by modifying the BIOS and placing the commands needed to patch them into the system into an AUTOEXEC.BAT file; the batch file should end with a program that executes an INT 27 interrupt, which allows the code to remain in the system until it is turned off. Much technical information (including an 80-page fully documented listing of the IBM BIOS) is included in the manual *Technical Reference*.

One interesting use of the IBM BIOS relates to the IBM keyboard. The keyboard, which contains an Intel 8048 microprocessor, does not deliver ASCII codes to the System Unit. Instead, it delivers two *scan codes* per keypress: one when the key is pressed, and a different one when the key is released. The IBM BIOS decodes the scan codes into an extended ASCII code that can return 256 one-byte codes and several two-byte codes for each keypress.

How Fast Is IBM BASIC?

Surprisingly, IBM BASIC is not much faster than its 8-bit counterparts. Table 7 compares the execution times of five BASIC programs on several popular microcomputers; the programs themselves are in listing 1. The first four benchmarks test an empty do-loop, division, subroutine jumps, and the MID\$ string function. The fifth test is a slightly modified version of Jim Gilbreath's Sieve of Eratosthenes benchmark program (see "A High-Level Language Benchmark," September 1981 BYTE, page

Command	Description
Un	move up <i>n</i> steps
Dn	move down <i>n</i> steps
Ln	move left <i>n</i> steps
Rn	move right <i>n</i> steps
En	move diagonally up and to the right <i>n</i> steps
Fn	move diagonally down and to the right <i>n</i> steps
Gn	move diagonally down and to the left <i>n</i> steps
Hn	move diagonally up and to the left <i>n</i> steps
Mx,y	move to point (x,y) or (if in relative mode) move (x,y) units from current position; plot a point
Bx,y	same as M, but no point is plotted
Nx,y	same as M, but return to original location when finished
An	set angle as a multiple of 90 degrees (<i>n</i> = 0 through 3)
Cn	set current color to <i>n</i>
Sn	set scale factor (step size)
Xstring\$	execute substring <i>string</i> \$

Table 5: Commands for the DRAW statement.

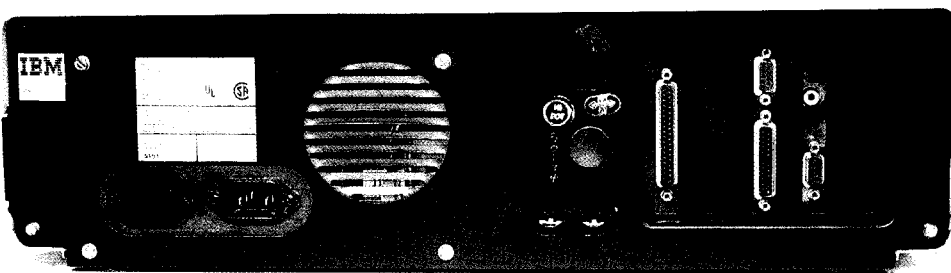


Photo 10: The back panel of the IBM Personal Computer. See the text for a description of the plugs and sockets.

180; note that the algorithm accesses lots of memory but uses only addition and subtraction.

The results of these comparisons are not encouraging. For example, IBM BASIC is somewhat faster than Applesoft, but the difference is modest, and Applesoft is one of the slower microcomputer BASICs. (All the BASICs tested are versions of Microsoft BASIC.) A comparison of IBM BASIC to Microsoft MBASIC 4.51 running on a 4-MHz Z80-based machine shows MBASIC to be faster in everything but division; this last

makes sense in that the 8088 microprocessor has a hardware divide instruction, which accounts for its better performance in the division benchmark. Still, it seems that IBM BASIC does not have a definite superiority over its 8-bit counterparts.

Although I hesitate to draw conclusions about the IBM microcomputer's performance in disk-based or machine-language programs, it is obvious that the IBM microcomputer does not gain a speed advantage from its memory access—the 8088 micro-

Listing 1: BASIC benchmark programs used in table 7. Listing 1a tests an empty do-loop; the two constants are included to allow the isolation of the features being tested in listings 1b and 1c. Listing 1b tests the division operation. Listing 1c tests a subroutine call-and-return sequence. Listing 1d tests the MIDS (substring extraction) operation. Listing 1e is the Sieve of Eratosthenes algorithm to generate prime numbers; it is used as a composite benchmark of several BASIC features working together in a short, but non-trivial, program.

1a

```
60 A=2.71828
80 B=3.14159
100 FOR I=1 TO 5000
320 NEXT I
```

1b

```
60 A=2.71828
80 B=3.14159
100 FOR I=1 TO 5000
120 C=A/B
320 NEXT I
```

1c

```
60 A=2.71828
80 B=3.14159
100 FOR I=1 TO 5000
120 GOSUB 1000
320 NEXT I
340 END
1000 RETURN
```

1d

```
80 A$="abcdefghijklm"
100 FOR I=1 TO 5000
120 B$=MID$(A$,6,6)
320 NEXT I
```

1e

```
1 SIZE=7000
2 DIM FLAGS(7001)
3 PRINT "only 1 iteration"
5 COUNT=0
6 FOR I=1 TO SIZE
7 FLAGS(I)=1
8 NEXT I
9 FOR I=0 TO SIZE
10 IF FLAGS(I)=0 THEN 18
11 PRIME=I+I+3
12 K=I+PRIME
13 IF K>SIZE THEN 17
14 FLAGS(K)=0
15 K=K+PRIME
16 GOTO 13
17 COUNT=COUNT+1
18 NEXT I
19 PRINT COUNT, " primes"
```

Address (in Hexadecimal)	Location	Type	Function
00000 00080	on System Board "	RAM "	BIOS interrupt vectors BIOS available inter- rupt vectors
00400 00500	" "	" "	BIOS data area workspace memory
10000 (decimal 64 K) 40000 (256 K)	on memory card not available now; reserved for future expansion	" "	workspace memory proposed workspace memory
A0000 (640 K) A4000 (656 K)	? on video boards	? RAM	reserved reserved for all forms of video display (note 1)
C0000 (786 K)	?	?	memory expansion area
F0000 (960 K) F4000 (976 K)	? on System Board	? ROM/PROM	reserved 8 K-byte slot available for user programs
F6000 (984 K)	"	ROM	40 K-byte BASIC in ROM
FE000 (1016 K)	"	"	BIOS code in ROM

Note 1: Not all this space is currently in use. The memory for the monochrome adapter card starts at hexadecimal B0000 (704 K bytes), and the memory for the color/graphics card starts at hexadecimal B8000 (736 K bytes).

Table 6: Memory map of the IBM Personal Computer.

	IBM			Applesoft		4 MHz Z80 MBASIC 4.51		Radio Shack TRS-80 Model II	
Benchmark	time	time	ratio to IBM	time	ratio to IBM	time	ratio to IBM	time	ratio to IBM
empty do-loop	6.43	6.66	1.04	5.81	0.904	7.98	1.24		
division	23.8	29.0	1.22	24.9	1.05	19.4	0.815		
subroutine jump	12.4	13.9	1.12	9.4	0.758	17.1	1.38		
MID\$ (substring)	23.0	32.3	1.40	18.6	0.809	24.8	1.08		
prime number program	190	241	1.27	151	0.795	189	0.995		

Table 7: Benchmark results for the IBM Personal Computer against several 8-bit microcomputers: an Apple II Plus running Applesoft BASIC, a 4 MHz Z80 microcomputer running MBASIC 4.51, and a Radio Shack TRS-80 Model II running Model II BASIC. All times (given in seconds) and ratios are valid to three significant digits. See listing 1 for the actual benchmark programs.

processor has to get memory one byte at a time, like the 8-bit 6502 and Z80. Still, this does not fully account for the IBM microcomputer's modest performance. Perhaps IBM's BASIC has only been translated from its 8-bit predecessors and not optimized for the 8088's instruction set.

Documentation

IBM uses the slogan "The IBM of Personal Computers" in one of its advertisements. The manuals that accompany the IBM microcomputer and various pieces of software could likewise be called "The IBM of Documentation." They will set the standard for all microcomputer documentation in the future. Not only are they well packaged, well organized, and easy to understand, but they are also *complete*. With the inclusion of the manual *Technical Reference*, the IBM Personal Com-

puter is as well documented as any existing microcomputer, and the documentation is available much earlier in the life of this machine than it has been for other machines.

Each manual is in a hardcover D-ring binder with its own slip cover. The pages are 14 by 21.6 cm (5.5 by 8.5 inches), and the binder is built so that the opened book lies flat. Included with the minimal configuration IBM microcomputer are manuals titled *Guide to Operations* and *BASIC*. A separate boxed manual is given with each software package; some IBM Personal Computer products have softcover documentation booklets.

Guide to Operations explains the capabilities of the IBM Personal Computer system and provides information to be used in the setup and initial operation of the microcomputer. A 145-page section called "Operations" describes the IBM keyboard layout and usage, the IBM 80 CPS Matrix Printer, the IBM DOS, and selected information on IBM BASIC. Other sections tell you what to do if the IBM microcomputer doesn't work, what additional peripherals are available for the system, and how to prepare the system to physically move it to another location. The manual is written in a friendly, tutorial manner and includes the basic information that most manuals take for granted (i.e., how to turn the machine on, how to start BASIC).

BASIC is 406 pages long and contains a 258-page section that fully describes each BASIC command, function, statement, and variable. Each BASIC keyword is documented under several headings: format (the syntax of the keyword), versions (the version or versions of IBM BASIC under which the keyword is available), remarks (a commentary that further explains the use of the keyword), and an example. Other sections describe the use of the BASIC Program Editor, floppy-disk I/O, communications files, and other topics.

The Reference Manual

The manual *Technical Reference* deserves special recognition simply

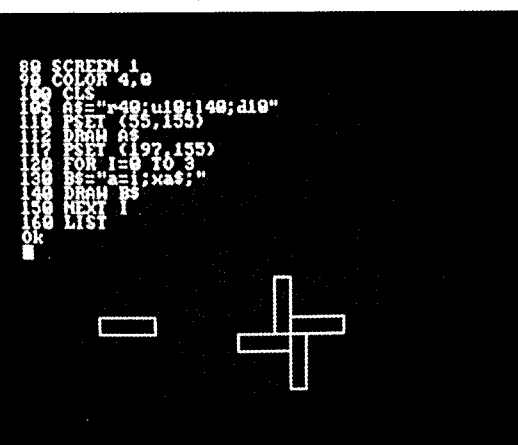
for its existence. It is 372 pages long and is in three sections, plus appendices; its price is a modest \$36. Section 1 gives a short overview of the IBM Personal Computer System and some of its internal workings. Section 2, which is 147 pages long, gives a functional specification for every piece of hardware in the IBM Personal Computer product line. This includes highly detailed specifications of the operation of the hardware, pinouts for peripheral connectors, and connection diagrams showing how to interface IBM peripheral cards with non-IBM devices. Section 3 describes the IBM BIOS. Five appendices give additional information, including a complete, commented listing of the IBM BIOS and schematics for all hardware in the system.

I'm sure that adventurous microcomputer enthusiasts will discover many more things about the IBM microcomputer as they buy and use the machine. But *Technical Reference* gives us a tremendous amount of information from the start. Most computer enthusiasts will want to have a copy of this book.

Sales and Service

Many companies are trying to become authorized IBM dealers; at the time of this writing, all Computerland stores are authorized dealers, and Sears Roebuck and Company has announced plans to sell the IBM Personal Computer through its Sears Business Systems Centers. IBM itself will sell its microcomputer through the IBM Product Centers in Baltimore, Philadelphia, and San Francisco. Since a potential dealer has to qualify as an authorized repair center before a dealership will be awarded, service will always be available from the dealer that sold you the unit.

IBM is also offering warranty extensions to increase the 90-day warranty that comes with the machine to one year, as well as annual maintenance contracts. The prices are reasonable; for example, the prices for a 48 K-byte system with one floppy disk and the monochrome display are \$154 for the warranty extension



and \$196 for the annual maintenance contract. On the other hand, the prices for the IBM 80 CPS Matrix Printer (which has a lot of moving parts) are \$141 and \$179, respectively.

Prices may become a source of potential bad feelings between you and the dealer. The prices quoted in the "At a Glance" textbox are suggested retail prices that are guaranteed to be in effect *only* at the three IBM Product Centers, listed above. Dealer prices may vary from this somewhat—expect a variation between \$10 and \$100 on most products, depending on their suggested price. However, at the time of this writing, one authorized IBM dealer is selling the Peachtree Software business packages (General Ledger, Accounts Receivable, and Accounts Payable) at \$995 each, a full \$400 above the IBM suggested price of

Advanced BASIC adds event trapping, some advanced graphics commands, and an advanced music-playing command.

\$595. Since the IBM suggested price includes a sufficient profit margin for most products, I think this price (which I confirmed with the dealer) is exorbitant.

The moral is to shop around for the best prices, if you can. However, this may be difficult for two reasons. First, IBM is probably going to authorize only one dealer per geographic area, at least initially. Second, the IBM microcomputer product line is probably not going to be available by mail for quite some time. Another problem with buying software from a dealer in a distant city is that the dealer is going to be responsible for software support. Still, for \$400, I would be tempted to buy my software in another city and make some long-distance calls when I needed software support.

Other Vendors

When the IBM Personal Computer was introduced last fall, IBM was the

sole supplier of both hardware and software. But potential hardware and software vendors have wasted no time in providing products for what they feel will be a very popular microcomputer. In particular, Lifeboat Associates announced last October that it is "making available most of its existing applications programs to serve users of IBM's new 16-bit Personal Computer." When this is accomplished, it will do a lot to ease the shortage of business and applications software that currently exists for the IBM Personal Computer.

(In a related development, Lifeboat also announced that it will be selling all its 16-bit software for the SB-86 operating system, which is its name for the IBM disk operating system. The fact that IBM, Microsoft, and Lifeboat have put their names behind this 16-bit operating system poses a serious threat to Digital Research's prospects of dominating the 16-bit market with its new CP/M-86 8086

operating system, as it has the 8-bit arena with its popular CP/M 8080 operating system.)

As for hardware, several gaps will, for the moment, be filled by outside vendors. IBM does not currently supply a high-quality RGB color monitor, a letter-quality printer, or any of the special input devices provided for in the system (joystick, light pen, paddles). IBM's position is that the potential demand for these products will cause third-party vendors to independently market them. (In the next section, I will discuss some problems with this philosophy.) In addition, the expansion slots provide the opportunity to interface the IBM microcomputer with many outside devices. Given a reasonable period of time, plenty of hardware and software will probably be developed for the IBM Personal Computer.

One other item of interest is the announcement of a new magazine called *PC: The Independent Guide to the IBM Personal Computer*. It is

published by David Bunell, of Software Communications, Inc., 44 Montgomery St., San Francisco, CA 94104; subscriptions are \$12 for 12 issues. It should be of great interest to owners of the IBM Personal Computer.

Current Weaknesses

The IBM Personal Computer is a very good machine, but it does have some shortcomings. This is no reflection on IBM's ability to design a microcomputer; rather, it is a reflection of the trade-offs between capability and cost that had to be made to make the machine competitive in the existing market.

The most serious weakness of the IBM Personal Computer is the small number of expansion slots available for future use. Note that I say "for future use"; one slot is taken up by a video adapter card (or two if you want both kinds of display), and another is taken up for each of the following devices: the 5¼-inch Diskette Adapter card (if you want a floppy disk), the Asynchronous Communications Adapter card (if you want an RS-232C port), the Printer Adapter card (if you have the color/graphics video card and want a parallel printer), and the Game Control Adapter card (if you want joysticks or game paddles). Since you need an empty expansion slot for each 64 K bytes of memory above the first 64 K bytes, it is obvious that you cannot put everything into the IBM microcomputer that you might want to. The most frequently encountered limitation is the amount of memory you can have in the microcomputer; if you want a floppy disk and the RS-232C card, you can have only (!) 192 K bytes of memory—all five slots are filled. With a moment's reflection, you will see that the expansion slots in the IBM Personal Computer will fill rather quickly.

At the moment, the IBM Personal Computer system is weak with respect to word processing. First, IBM does not market a letter-quality printer. This means that, if you want to do word processing on the IBM microcomputer, you have to trust that your IBM dealer will also sell

DUAL/68000

... Hard Power from
the Soft Support of

UNIX®



A new and powerful computer has been born...
the System 83. The versatile UNIX* operating system pilots the System 83's raw power through a myriad of software such as "C", FORTRAN, PASCAL, BASIC, COBOL, and even Networking. Step into a bold new frontier with more power than you ever dreamed possible.



☐ UNIX V7 configured by UNISOFT**
☐ Full IEEE 696/S-100 Compatibility

☐ MC68000 8MHz Processor
☐ 32-Bit Data Operations with 32-Bit Internal Registers

☐ 16-Bit Data Transfer Operations
☐ Memory Management Allows Concurrent Use of Mapped and Non-mapped Address space

☐ 16 Mb of Main Memory Directly Addressable
☐ 7 Vectored Interrupt Levels

☐ 256 Kb of RAM with Parity Per Board Slot

* UNIX is a trademark of Bell Laboratories and is supported on the DUAL System 83 by UNISOFT

** UNISOFT is a trademark of UNISOFT Corporation of Berkeley, CA.

DUAL SYSTEMS
system reliability/system integrity

720 Channing Way • Berkeley • CA 94710 • (415) 549-3854 • Telex: 172029 SPX

you a letter-quality printer and cable that will work properly with your software. This problem of compatibility has been very common in the microcomputer industry to date. It is surprising that IBM, which has worked so hard in some other areas to avoid this problem, has literally left it to chance here.

Of course, the explanation of "limited resources" can be given here, too. That is, if IBM had waited until it had *everything* lined up, the product would not have been introduced as early as it has been. Still, the criticism stands that, by not providing a full product line, *the IBM Personal Computer system, through no fault of its own, may fall prey to hardware and software incompatibility*, thus creating still more disappointed microcomputer users.

Another weakness of the IBM Personal Computer as a word processor is the lack of versatile word-processing software to drive the machine. The only word processor available at the time of this writing is Information Unlimited's EasyWriter. I was given a chance to work with the EasyWriter word processor on the IBM microcomputer, and I found a few things I didn't like about it. In general, the software didn't seem to be of the same caliber as, say, VisiCalc or the Peachtree business packages. Specifically, at times the software left me not knowing exactly what to do next, and I found the scrolling—both up *and* down—to be slow. (Scrolling down is understandably slow because the entire screen has to be rewritten, but scrolling up is usually fast, whether it be on a memory-mapped video display or a terminal. On the IBM EasyWriter, the scrolling is as slow going up as it is going down.) I have used the Apple II version of EasyWriter extensively, and my opinion of it is the same as for the IBM version: it is a good piece of software for the money, but it isn't as versatile as some applications require.

I'm sure that Magic Wand, WordStar, or something similar will be available very soon for the IBM microcomputer, but EasyWriter is the only choice for the moment. My ad-

vice is: if you have an IBM Personal Computer, use the EasyWriter package a lot before you buy it. If you are looking for a system to be used primarily for word processing and you can't afford to wait for better software, I suggest that you look at other existing systems, such as the Radio Shack TRS-80 Model II or the Xerox 820. The IBM system, as it currently stands, does not compare favorably with these other systems.

Another limitation of the IBM Personal Computer is that, even though up to 256 K bytes of memory are available, the extended Microsoft BASIC cannot access more than a 64 K-byte workspace (I assume this includes both program and data), even though the IBM Pascal Compiler (also by Microsoft) and other proposed system software are said to be able to access all the workspace memory in the machine. Sixty-four K bytes seems to be so much memory, especially since we are used to program, data, *and* the BASIC interpreter fitting into 64 K bytes. Still, it's unfortunate that all that extra memory (which is one of the main reasons for buying the machine) can't be used by BASIC, the computer language that will probably most often be used on the machine.

Another weakness that must be mentioned is an extension of one previously discussed: the IBM dealer will have to supply certain useful or even vital components of a complete IBM microcomputer system. IBM says it has no interest in manufacturing color monitors, letter-quality printers, joysticks, or light pens, nor can IBM supply you with the cables that will have to be made to connect these devices to the IBM microcomputer. In addition, if you want to connect your IBM microcomputer to a standard color TV (which is what most people will do), you will have to rely again on the judgment of your IBM dealer for the correct cable and RF modulator. I'm sure that in most cases no serious problems will arise, but by not making the entire product line itself, IBM has lost its guarantee of total system compatibility.

As someone not unacquainted with the programming of games, I found a

few inadequacies with the graphics commands of the extended Microsoft BASIC. Although the graphics-definition language is excellent, shapes are allowed to rotate only in 90-degree increments. In addition, the only way I found to detect the collision of a drawn shape with the contents of the screen is through a POINT function that gives the color number of a given point on the screen. Although this *can*, with some effort, be used for that purpose, it falls far short of the methods of detecting collisions available on the Apple II and the Atari 400/800 computers. Perhaps some enthusiastic programmer will find a memory location that indicates whether or not a drawn shape has collided with another image on the screen. In any case, these are small criticisms of a machine that does so many things so well.

Speculations

One interesting thing about IBM is that it refuses to acknowledge the existence of any product that is not ready to be put on dealers' shelves tomorrow. Although this is frustrating at times, it is a refreshing change from some companies' practice of announcing a product even before its design is finished. Here are some speculations about future IBM Personal Computer products. The first two are almost assured, while the rest follow in increasing degree of uncertainty.

- *Two more disk drives.* Although, at the time of this writing, IBM maintains that only two disk drives are available for the IBM microcomputer, *Technical Reference* indicates in several places that provision is made for two external disk drives to be connected to the 5¼-inch Diskette Drive Adapter via the DB-37 connector on the back of the adapter card. (See the leftmost plug in the expansion slot area in photo 10.)

- *An 8086/8088 macro assembler.* The *Technical Reference* bibliography lists a manual for the *IBM Personal Computer Macro Assembler*. It may be available by the time you read this article.

Now we start with the speculations:

- *SofTech Microsystems' UCSD p-System.* IBM announced that this operating system would be available for the IBM Personal Computer; this would make UCSD Pascal, FORTRAN, and BASIC available, and it would allow the IBM microcomputer to run the same programs as other UCSD systems. However, IBM would not give me any availability dates.

- *A typing tutorial program.* This was mentioned once in the front of the *IBM Guide to Operations*—but then, so were joysticks and RF modulators. Microsoft may adapt its Typing Tutor for the IBM Personal Computer.

- *An official letter-quality printer and a major-league word processor.* IBM may have plans to do this, or it may be relying on manufacturers' eagerness to expand their potential market. Someone will probably do it, but it may not be IBM.

- *An "expansion box" to increase the number of peripheral cards that can be placed in the computer at one time.* This would resolve a design limitation of the IBM Personal Computer as it now exists.

- *A 128 K-byte (or more) memory board.* As the 64 K-bit memory ICs decrease in price and become more available, IBM may market expansion boards that hold more than their current 64 K-byte limit. This would free up one or two expansion slots, but it might also allow the IBM Personal Computer to hold more than 256 K bytes.

- *A database management system.* This, like many other business packages, is needed to strengthen the position of the IBM microcomputer in the business area.

- *An official RGB color monitor.* I don't think IBM is going to go for this one, but it should. I have seen three separate IBM Personal Computers with RGB monitors. In all three cases, the monitor used did not have an input for the intensity signal and so could display only eight of the sixteen

Text continued on page 68

possible colors. This is, again, a situation where IBM has surrendered the guarantee of complete compatibility by not manufacturing the product itself.

- *A Winchester hard disk* (the bigger, the better). Admittedly, this is a real guess, but it would make the IBM Personal Computer more attractive for certain business applications. The Apple III, a direct competitor to the IBM microcomputer, is now being offered with a Winchester disk. Is IBM going to ignore this?

- *Memory expansion past 256 K bytes.* It may be possible to replace the 16 K-bit 4116 dynamic memory integrated circuits with the new 64 K-bit devices, both on the main printed-circuit board and on the memory-expansion cards. If this can be done, the theoretical memory limit is the 20-bit, one-megabyte addressing limit of the 8088 microprocessor. The actual limit is somewhat less than that—a memory map in the *Technical Reference* manual (see table 6) allows room for “future expansion” of 576 K bytes, for a total of 832 K bytes.

Summary

When I look at the several inches of IBM Personal Computer manuals that fill my bookshelf, I am reminded that there is so much about this system that I have left out. Still, I have tried to talk about the most exciting and most important aspects of the system. The genius of the people who designed the IBM microcomputer is that they managed to do everything conventionally but well—the IBM Personal Computer doesn't have any startling innovations, but it also lacks the moderate-to-fatal design problems that have plagued other microcomputers.

The IBM Personal Computer isn't as well supported as the Radio Shack TRS-80 family or the Apple II, but then it hasn't been around very long. In two years or so, I think the IBM microcomputer will be one of the most popular and best-supported microcomputers around. This microcomputer is as close as I've ever seen to being all things to everybody. IBM should be proud of the people who designed it. ■