CORONA SPECIFICATIONS

Display resolution	256 × 256
Display size	256 horizontal × 208 vertical
Displayed pixels	53,248
Memory size	8192 8-bit bytes low version; 24,576 8-bit bytes full range
Memory access modes	bit/byte
Memory organization:	
Bit mode	53,248 3-bit pixels
Byte mode	Three 8K byte planes (can also be used for normal data storage)
S-100 bus memory space	8K maximum
S-100 bus I/O ports	Two input, two output
Possible colors	256 (or 32 levels of grey))
Displayed colors	Sixteen maximum.
Display options	Alphanumerics can be mixed and overlaid with graphics and/or external video input on color or monochrome displays
External video input	Software-selectable, accepts standard RS-170 video; external source must be synced to computer
One frame	Two identical fields (noninterlaced)
One field	16.667 ms (1/60 s)
Horizontal sweep cycle	64.1 μs
Horizontal blanking	From 6 to 14 μs
Active display	40.3 μs
Horizontal scanning frequency	15,600 Hz
Vertical scanning frequency	60 Hz
Color subcarrier	3.579 MHz
Scan lines per field (noninterlaced)	260
Vertical blanking field	833 to 1300 μs
Power requirements	+8 to +10 V dc, 2 A maximum +15 to +20 V dc, 0.2 A maximum −15 to −20 V dc, 0.2 A maximum

The color-map RAM is a high-speed 16-word by 8-bit RAM array. This organization permits the choice of 256 colors (more precisely, 64 colors, each of which has four intensities). Colors for each point on the screen are determined by the three-bit code stored in the Graphic Display Memory that addresses the color-map RAM. The color-map RAM is loaded under program control via the A register.

The function of the color-video encoder is to transform the red, green, blue, and luminance data from the color map into NTSC color signals. This encoder is comprised of timing, four two-bit D/A converter, and the actual color-encoder sections.

The timing section generates the color-burst flag and composite sync (from the SOL-20 or VDM-1 composite sync). The red/green/blue/luminance D/A converters accept digital data from their respective sections of the color map and convert it to analog color-difference signals (R − Y and B − Y). The encoder section is designed around an IC that modulates the color subcarrier with the color-difference signals and outputs the composite video that consists of the video, blanking, and sync signals.

Physical Details. The Corona's circuitry mounts on two large printed circuit boards. The larger board plugs directly into the S-100 bus of the computer and contains 73 ICs, four voltage regulators, and miscellaneous discrete components. The smaller board, which contains 32 ICs that include the RAM memory system and the data multiplexers, is then connected to a jack on the larger board. ◇

A CASSETTE CONTROL SYSTEM for Computers

If you store many computer programs on tape, use the Aux Box to retrieve them quickly and easily

BY A. A. MANGIERI

AUDIO CASSETTE tape recorders make excellent low-cost mass-storage devices for home and small-business computers. They provide an efficient approach for storing a single long program on a cassette tape. For short programs, however, it is usual to store a string of different programs, separated by guard bands, on a common tape track, which leads to retrieval problems. The "Aux Box" cassette deck controller presented here simplifies the process of saving and loading multiple programs that, of necessity, are stored serially on a tape. Although designed to interface directly between Radio Shack's Model TRS-80 microcomputer and its mating cassette recorder without alterations, the Aux Box can be adapted to other systems.

There are two basic ways to locate a particular program on a cassette tape—strictly manually, using the tape deck's index counter, or with a computer-generated header that "tells" the computer when the named program is passing the playback head.

Working manually, you must "tease" the various tape-speed controls to get to the starting point of a desired program. When using the header approach, you may have to wait quite a while for the tape to get to the desired program. In many cases, including the header approach, you still must make the recorder move rapidly to the desired starting point, as specified by the tape counter. Also, with some computer/tape-deck systems, including the Model TRS-80, you must disconnect cables to regain manual control for rewinding, spotting tape, and monitoring. The Aux Box solves these problems.

With the Aux Box, you can transport tape at fast, medium, and slow speeds in either direction. This makes it very easy to accurately position the tape. Moreover, without removing any cables, control can be transferred between the computer and the recorder at the flip of a switch, while retaining monitoring capability at all times. Another circuit in the Aux Box permits audio recording without the need to dismantle the setup. Finally, a separate circuit in the Aux Box provides backup protection for the computer's relay.

About the Circuit. The complete circuit for the Aux Box is shown in Fig. 1, along with the computer/cassette-recorder interface. With the plug removed, the recorder's REM (remote) jack is normally shorted to ground, which puts the recorder's negative bus at ground potential. With the plug inserted in the REM jack, the computer starts and stops the recorder via the computer's normally open, fully isolated relay contacts.

Switch *S2* transfers control between the computer and the cassette recorder. With *S2* set to its COMPUTER position, the computer controls the recorder via the solid-state switch made up of *Q1* and *Q2*. This transistor switch eliminates the current surge through and provides full protection for the relay's contacts, which now carry only the very small base current for *Q1*.

Pushbutton switch *S7* permits tape advance following a data dump without returning control to the recorder.

With *S2* set to its RECORDER position and *S3* set to its off (REM MIC/TAPE SPEED) position, pushbutton switches *S4, S5*, and *S6* provide FAST, MEDIUM, and SLOW speed control, respectively, for the recorder's fast-forward and rewind modes. As selected by the pushbutton switches, diodes *D1* through *D5* vary the speed of the tape by controlling the voltage to the tape recorder's motor. The recorder's amplifier voltage is also varied, but monitoring is still available at a slightly reduced volume.

Switch *S1* alternately breaks one of two circuit grounds between the computer and the recorder. This ground-loop break reduces or eliminates any slight ac hum that can otherwise be heard during monitoring.

Jack *J3* accepts an earphone for audibly monitoring the signal from the tape recorder.

With *S2* set to its RECORDER position, *S3* is used to select either MIC or REM MIC for audio recording with *J4* and *J6*.

Construction. The circuit for the Aux Box can be assembled in a small plastic box, using the bottom of the box as the

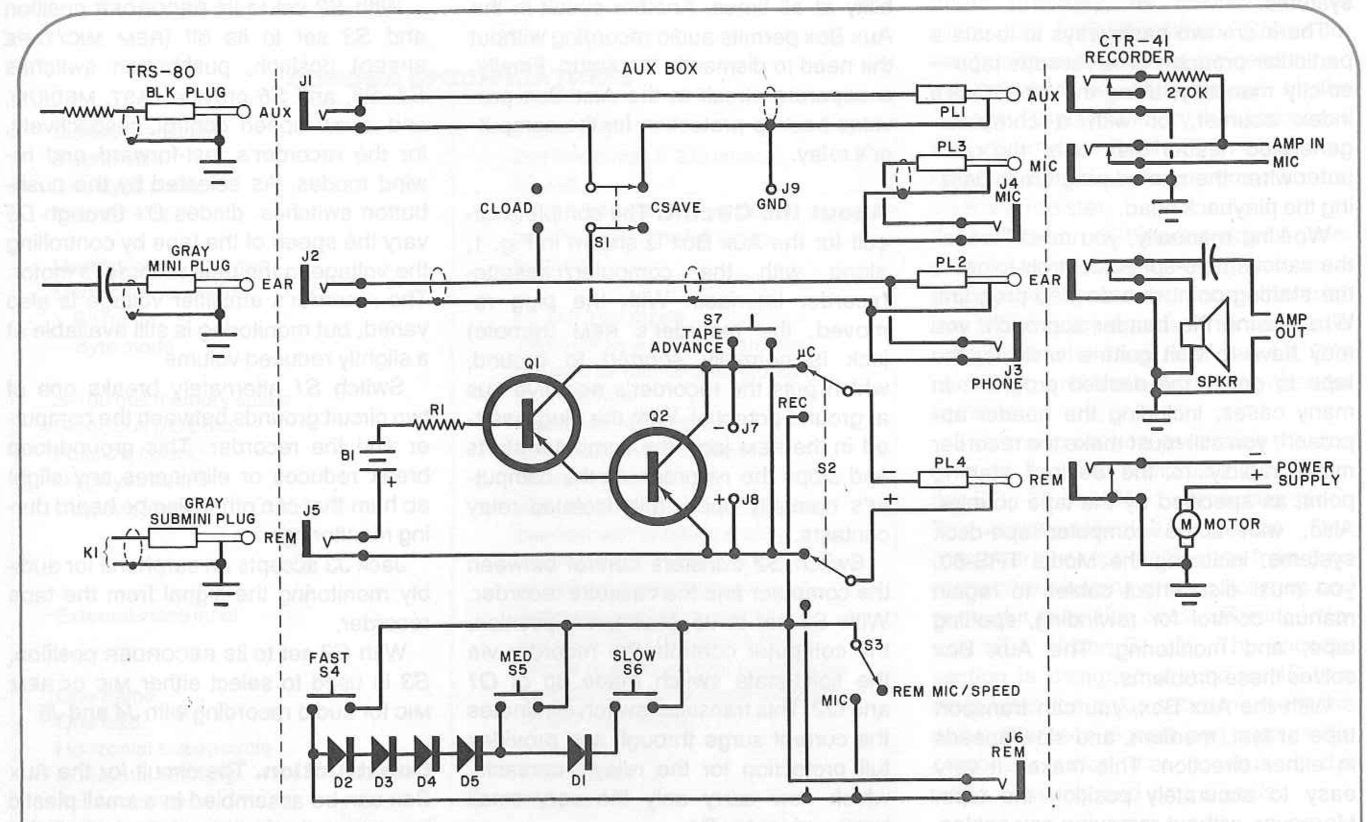


Fig. 1. Schematic of the Aux Box along with the computer/cassette-recorder interface.

PARTS LIST

B1—1.5-volt "C" cell
D1 through D5—1N4001 rectifier diode
J1 through J4—Miniature phone jack
J5,J6—Subminiature phone jack
J7,J8,J9—Phone tip jack
PL1,PL2,PL3—Miniature phone plug
PL4—Subminiature phone plug
Q1—HEP 251 (Motorola) or similar transistor
Q2—HEP 232 (Motorola) or similar transistor
R1—2200-ohm, ½-watt, 5% tolerance resistor
S1,S2—Dpdt toggle switch
S3—Spst toggle switch
S4 through S7—Spst normally open momentary-action pushbutton switch.
Misc.—Suitable plastic case; terminal strip; TO-3 transistor socket; battery holder; earphone; shielded cable; hookup wire; solder; machine hardware; etc.

control panel. Point-to-point wiring techniques are perfectly adequate for assembling the Aux Box. You can use a terminal strip for mounting *Q1* and to provide a convenient means for connecting it into the circuit. Also, use a TO-3 socket, spacers, and machine hardware to mount *Q2*. Diodes *D1* through *D5* are best mounted on a tag or terminal strip to permit easy removal should you decide to alter the speed of the cassette transport.

The walls of the box in which the Aux Box is mounted are fairly thick. This means that you must use jacks with fairly long bushings. Alternatively, you can counterbore the mounting holes for the jacks and use standard-length jacks.

The AUX, EAR, and REM jacks (*J1, J2*, and *J5*) should be mounted on the rear wall of the box. Jacks *J3, J4*, and *J6* (PHONE, MIC, and REMOTE jacks, respectively) go on the front of the box (see Fig. 2). CLOAD/CSAVE, COMPUTER/RECORDER, MIC/REM MIC, TAPE SPEED (FAST, MEDIUM, and SLOW), and TAPE ADV switches *S1* through *S7*, respectively, all go on the bottom of the box, which becomes the control panel. After drilling the holes for but before mounting the jacks and switches in their respective locations, it is a good idea to label each with a white dry-transfer lettering kit as shown in Fig. 2.

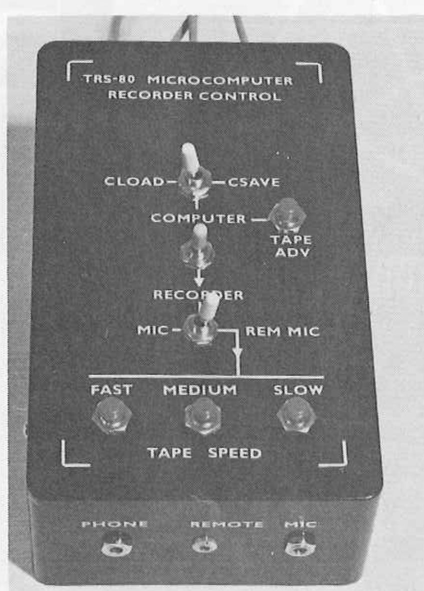


Fig. 2. Photo shows controls and identification on front panel.

Use insulated hookup wire for the REM circuit and shielded cable elsewhere. Check *PL4* with an ohmmeter and wire the cable to *S2* accordingly. Jacks *J7* and *J8* can be installed in any convenient location. The cables to which AUX plug *PL1*, MIC plug *PL2*, and EAR plug *PL3* are attached should exit the box near the respective jacks on the rear of the box.

Inclusion of ground jack *J9* is optional. Use a battery holder for *B1*. Also, use either a 4-ohm or, preferably a 400-ohm, impedance earphone for low loading.

Test and Adjustment. Connect a dc milliammeter in series with a 10,000-ohm potentiometer across *J5* to measure the base current of *Q1* and a dc voltmeter between *J7* and *J8* to measure $V_{CE(sat)}$ of *Q2*. Set *S2* to COMPUTER and insert *PL4* into the recorder's REM jack.

Now, set the pot to minimum resistance and note the base current of *Q1*. With the recorder's PLAY lever engaged, $V_{CE(sat)}$ should be 0.2 volt or less. Advance the pot until $V_{CE(sat)}$ just begins to increase and note the base current of *Q1*. This is the minimum required base current for *Q1*. Note this current, multiply it by 2 or 3, and adjust the pot until the milliammeter indicates the result of your calculation.

Without disturbing the setting of the pot, remove it from the circuit and use an ohmmeter to measure its resistance. Then wire into the *R1* location in the circuit a fixed resistor whose value is as near as possible to your measurement. The reason for choosing a resistor whose value yields twice or three times the minimum base current for *Q1* is to allow for battery ageing.

Set *S2* to RECORDER and observe the speed of the cassette tape in both the fast forward and rewind modes with the recorder's PLAY lever engaged and disengaged. The speed can be altered by diode selection. If desired, a separate string of diodes can be used for *S5* and *S6*. On SLOW, different speeds can be observed on fast forward and rewind with the PLAY lever engaged and disengaged. This is of no consequence. A good choice for a MEDIUM tape speed is half the normal fast-forward speed.

Saving and Loading Programs. Program locations on the tape are referenced to the recorder's tape-counter readings. Rewind the tape and reset the counter to 000. If the tape has a leader, advance the tape to just beyond the leader. Keep a record of the counter readings at the start and end of each program. Set the recorder up for recording and set *S1* and *S2* on the Aux Box to CSAVE and COMPUTER. Type CSAVE and hit ENTER on the computer to start recording data. Use the earphone at *J3* (PHONE jack) to verify that data is being transferred.

There will be very short blank spaces on a tape if you allow the computer to space the programs. Therefore, it is a good idea to put the tape recorder in the play mode and press *S7* to advance the tape an additional three to five digits on the index counter after each program. (It has proven handy to assign program starting locations at numbers on the counter ending in 0 or 5. Then the actual recording can be started at, say, 23, 38, etc., to allow a short duration guard band. Also, allow two or three digits of blank tape on the other side of the assigned starting location.)

As an example of the foregoing, let us assume that a data dump ends at 144. Here, you might advance the tape to 153 and record the starting location of the program as being at location 150 on the index counter. When saving programs, remove the microphone from *J4* and *J6*.

The Aux Box also permits various procedures for loading programs. (You will eventually adopt your own.) First, the earphone monitor is active when the recorder is in fast forward, play, and rewind only when the PLAY lever is engaged. The monitor's volume drops slightly when the SLOW button on the Aux Box is pressed, but it will be adequate.

There are three general methods for positioning the tape, plus many variations. The tape can be positioned with the PLAY lever engaged at all times, with continual monitoring of the tape. As usual, the recorder's FAST-FORWARD and REWIND levers do not latch in the engage positions with the recorder in the play mode and, therefore, must be firmly held down to move the tape. Second, with the PLAY lever disengaged (monitor inactive), the tape can be run just short of the target location and then switched over to recorder play to access the monitor while setting the tape between data jumps. Finally, the tape can be positioned with the PLAY lever disengaged and use of the monitor deferred until a load is required. The last approach works well if you save programs as described above.

GRAPHICS STAR PROGRAM

The eye-catching star on the monitor in the lead photo can be duplicated with a TRS-80 computer. The program given here causes a star to be drawn, pauses, and then proceeds to draw the pattern in reverse. Conserving memory or execution time were secondary in this program; several improvements are possible. Look for them. Lines 40 through 90 draw three nested inverted V's. Lines 210 through 250 draw a left-to-right descending line. Lines 310 through 340 draw a right-to-left descending line. Lines 240 and 330 place lower limits on the screen.

The diagonal line routines merit close study for use in other graphic programs. In line 210, try other integer and noninteger numbers for B along with several less than one. Some unexpected surprises are in store; try it! As you ponder the results, consider that screen X-Y coordinates are integers and that computation X/B is often a noninteger.

Lines 410 through 490 introduce a pause followed by reversal of screen. For some artistic effects, add STEP 5 to line 420. Add line 425 IF Y = 20 THEN NEXT Y and run. Isn't that something? Try other step increments in lines 420 and 430 alone and in combination. With a few more statements and functions, you can come up with some astonishing abstract stars.

```
10 REM * TRS-80 GRAPHICS STAR PROGRAM *
20 REM * DRAW INVERTED V *
30 CLS
40 FOR Y = 0 TO 2
50 FOR A = 0 TO 47 - Y
60 SET (64 + A, Y + A)
70 SET (63 - A, Y + A)
80 NEXT A
90 NEXT Y
100 REM * DRAW HORIZONTAL LINE *
110 FOR X = 0 to 127
120 SET (X, 20)
130 NEXT X
200 REM * DRAW LEFT DESCENDING DIAGONAL *
210 B = 4
220 FOR X = 0 TO 127
230 SET (X, 20 + X/B)
240 IF 20 + X/B  > 47 THEN 310
250 NEXT X
300 REM * DRAW RIGHT DESCENDING DIAGONAL *
310 FOR X = 127 TO 0 STEP - 1
320 SET (X, 20 + (127 - X)/B)
330 IF 20 + (127 - X)/B)  > 47 THEN 410
340 NEXT X
400 REM * PAUSE AND REVERSE SCREEN *
410 FOR N = 1 TO 2000: NEXT N
420 FOR Y = 0 TO 47
430 FOR X = 0 TO 127
440 IF POINT (X,Y) = 0 THEN 470
450 RESET (X,Y)
460 GOTO 480
470 SET (X,Y)
480 NEXT X
490 NEXT Y
999 GO TO 999
```

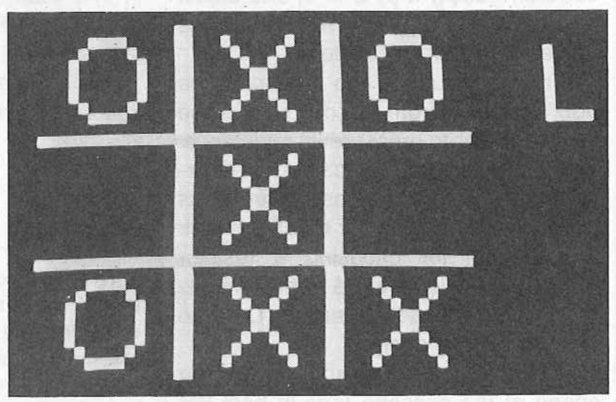
Here's an example of how the Aux Box can be used in conjunction with a cassette tape system. Assume that the assigned location of a program on tape is at 150 on the index counter. Rewind the tape by setting *S2* to RECORDER and *S3* to TAPE SPEED. Latch the recorder's REWIND lever and press *S4* (FAST switch) to rewind a short tape. With long tapes, use *S3* in the MIC position for rewinding. Reset the index counter and then operate the recorder's STOP lever.

To position the tape, press the recorder's FAST-F lever and latch it. Press *S4* (FAST) and observe the counter's 10's and 100's digits. When 14X appears on the counter, release *S4*. The tape will probably coast to about 146. At this point, you must select between immediate use of the monitor (PLAY lever engaged) or defer use of the monitor (PLAY lever disengaged) and move the tape to 150 on the index counter, using *S5* (MEDIUM) or *S6* (SLOW).

The tape can be run back and forth to access other programs without rewinding or resetting the index counter. Going backward (rewinding) from 175 to 20, use the REWIND lever on the recorder and *S4* (FAST switch) on the Aux Box, letting go of *S4* when 03X appears in the index counter. Use SLOW reverse to jog the tape to its final position. On the MEDIUM speed, the counter's 1's decade is at least readable and tape coasting is reduced by one-half. This can be used to great advantage.

Summing Up. If you store many programs on one cassette tape, the Aux Box lets you see just how easy it can be to retrieve programs quickly and precisely. Without having to pull cables, you can transfer control between the computer and tape deck at the flip of a switch and monitor the tape at all times. And you can transport tape at any of three speeds in either direction. In short, the Aux Box takes much of the hassle out of loading programs. ◇

A TIC-TAC-TOE GAME for your Elf Computer



Use a simple light pen as an input selector and the programs given here.

BY EDWARD M. McCORMICK

AN ELF computer that contains 1K of RAM can be programmed to play Tic-Tac-Toe if it is equipped with a video (1861) display and a light pen, the latter to be described here. The computer, using an O, plays against the human (X), and either the computer or the human can go first.

Unless forced to make some other play, the computer will randomly select any open position as its response. This results in a wide variety of games, and although the computer can be beaten, it will also win more than one might expect.

Playing the Game. The playing sequence is straightforward. The IN switch is operated to clear the screen. If any toggle switch is on, the computer will indicate its O after a pause of two seconds. If the input toggle switches are all off, the computer will wait for you to make the first move.

Whenever you are to play, a P will be displayed at the upper-right corner of the screen. You place the light pen in front of the position you wish to play, then depress the IN switch. An X will appear at the selected position. The computer will then indicate its response and you play again. This cycle continues until the game ends.

A D for draw, W for win, or L for lose will be shown at the upper-right corner at the end of a game. To clear the screen and set up for another game, simply operate the IN switch.

The Light Pen. The light pen consists of a cadmium-sulphide cell (Radio Shack 276-116) connected between EF-3 and ground in the Elf. Make sure that a 47,000-ohm resistor is connected between EF3 and +5 volts.

The main program, subroutines, and data sets are given in Tables I through V. Program execution starts at 0400. Table VI indicates the initial contents of page 6, the display area. Note that this forms the familiar Tic-Tac-Toe grid.

The program requires about 525 bytes, including main program, subroutines, and data storage. In addition, 256 bytes are used for the display area. The remainder of the RAM space can be used for embellishments to the program.

Initially, the program sets up the registers used. After the IN switch has been operated, the nine playing positions and the status position are cleared and N (the number of plays) is set to zero. If the input toggle switches are not set to zero, the program goes to the "any place" position (Table I). The computer randomly selects one of the unplayed positions and places an O in that location. The program then adds a 1 to N and checks to see if all nine positions have been played. If not, it allows the opponent to play an X. Note that when the input toggle switches are at zero, the program goes directly to X.

TABLE I—MAIN PROGRAM

Loc.	Program	Comments
04 00	C0 05 20	Initialize Registers
04 03	E9 69	Start video display
04 05	3F 05 37 07	Operate IN Switch
04 09	F8 E0 A3	Clear All Positions
04 0C	E3 F8 14 A5	including 9 playing
04 10	F0 32 18	and the status
04 13	A6 D4 13	position
04 16	30 0C	
04 18	F8 00 A8	N = O
04 1B	E9 6C 32 4D	Toggle .Sw On?
04 1F	8B AC 8C	Any Place
04 22	FF 09 32 2A	Determine random
04 26	33 22 FC 09	position to play,
04 2A	FC E0 A3	if occupied go
04 2D	E3 F0 A6 E6	back for another
04 31	F0 3A 1F	random position
04 34	F8 80 59 DD	'O'
04 38	F8 24 A5 D4	Write 'O'
04 3C	88 FC 01 A8	N = N + 1
04 40	FF 09 3A 4D	N = 9?
04 44	F8 44 A5	'D'
04 47	F8 0F A6 D4	If N=9, write 'D',
04 4B	30 05	return to restart
04 4D	F8 4C A5	Operate IN Switch
04 50	F8 0F A6 D4	Write 'P' then
04 54	3F 54 37 56	wait for IN switch
04 58	F8 01 59 DD	'X'
04 5C	F8 00 A3	If area bright
04 5F	36 6B	then skip to turn on
04 61	83 FC 01 A3	'?' and retry after
04 65	FF 20 3A 5F	short wait
04 69	30 78	
04 6B	F8 2C A5	Put '?' in status
04 6E	F8 0F A6 D4	
04 72	F8 20 59 DD	
04 76	30 5C	
04 78	F8 14 A5	If area dark, start
04 7B	F8 0F A6 D4	scanning
04 7F	F8 00 A7	Get playing position
04 82	FC E0 A3 E3	from table
04 86	F0 A6 BC	If it is occupied
04 89	E6 F0 32 A1	get next position
04 8D	87 FF 08	If all unoccupied
04 90	32 98	positions scanned but
04 92	87 FC 01 A7	not recognized, turn
04 96	30 82	on
04 98	F8 2C A5	'?' as status but
04 9B	F8 0F A6 D4	continue
04 9F	30 7F	scan
04 A1	F8 1C A5 D4	Write 'X' in
04 A5	F8 01 59 DD	unoccupied position,
04 A9	F8 00 A3	make tries to see
04 AC	36 BE	if it is pen
04 AE	83 FC 01 A3	light position
04 B2	FF 20 3A AC	
04 B6	9C A6	If not recognized
04 B8	F8 14 A5 D4	erase 'X' and go to
04 BC	30 8D	next position.
04 BE	F8 14 A5	When 'X' recognized
04 C1	F8 0F A6 D4	clear status.
04 C5	F8 1B A5 DA	Lost?
04 C9	86 32 D5	If so, write
04 CC	F8 3C	'L'
04 CE	A5 F8 0F A6	and return
04 D2	D4 30 05	to restart
04 D5	88 FC 01 A8	N = N + 1
04 D9	FF 09 32 44	N = 9?
04 DD	F8 06 A5 DA	Win?
04 E1	86 32 F0	If so, write
04 E4	F8 80 59 DD	'O'
04 E8	F8 24 A5 D4	to win then write
04 EC	F8 34 30 CE	'W'
04 F0	F8 12 A5 DA	Def?
04 F4	86 32 1F	If so, play that 'O',
04 F7	30 34	else go to Any Place

TABLE II—CHARACTER PRINT SUBROUTINE WITH DATA FOR CHARACTERS USED.

Loc.	Program	Comments
07 00	DF E5 F8 08 AC F0 56	Write reg 5 8 byte
07 07	8C FF 01 AC 32 00	* character into reg 6
07 0D	15 86 FC 08 A6 30 05	* position on screen
07 14	00 00 00 00 00 00 00 00	Blank
07 1C	81 42 24 18 18 24 42 81	X
07 24	3C 42 81 81 81 81 42 3C	O
07 2C	00 0E 11 02 04 04 00 04	?
07 34	00 11 11 11 15 15 1B 11	W
07 3C	00 10 10 10 10 10 10 1F	L
07 44	00 1E 11 11 11 11 11 1E	D
07 4C	00 1E 11 11 1E 10 10 10	P

TABLE III—DELAY SUBROUTINE

Loc.	Program
05 F0	DF E9 8B F4 AC 59
05 F6	8B F7 3A F6 30 F0

Immediately after the opponent has played, the computer checks to see if it has lost. If it has lost, the computer displays an L and waits for the IN switch to be operated to start the next game. If the computer has not lost, it adds 1 to N to see if all nine positions have been played and, if so, displays a D and awaits a new game.

If the computer has not lost and the game is not a draw, the computer checks to determine if it can win. All eight sets of positions are checked to find one that has two O's and the third position blank. If it finds such a combination, it will display an O in that third position, display a W and return to restart.

If the computer has neither lost nor won, the program then takes defensive action. The eight sets of positions are examined to see if the opponent has two X's and the third position blank. If it finds such a situation, the computer inserts an O at that position, and continues. If not compelled to make a defensive play, the

TABLE IV—LOSE-WIN-DEFENSIVE (L-W-D) SUBROUTINE

Loc.	Program	Comments
05 60	DF F8 C0 A7	Set addr init position
05 64	F8 00 A3 AC	Clear counters
05 68	E7 F0 AE EE F0 32 7F	Examine location
05 6F	FF 81 32 79	* addressed,
05 73	8C FC 03 AC 30 81	* if 'O' add 3,
05 79	8C FC 09 AC 30 81	* if 'X' add 9, else
05 7F	8E A6	* store blank position
05 81	17 83 FC 01 A3	Repeat for all 3 positions
05 86	FF 03 3A 68	* of combination
05 8A	85 59 E9 8C F7 32 9B	If L-W-D match not found,
05 91	87 FF D8 3A 64	* go to next combination
05 96	F8 00 A6 30 60	Exit if all 8 combs fail
05 9B	86 3A 60	Set register 6 and exit
05 9E	F8 01 A6 30 60	* if L-W-D match found
05 C0	09 0B 0D 09 63 BD	The eight combinations
05 C6	09 61 B9 0B 63 BB	* of three positions
05 CC	0D 63 B9 0D 65 BD	* (rows, columns
05 D2	61 63 65 B9 BB BD	* and diagonals)
05 E0	63 09 0D B9 BD	The 9 playing and the
05 E5	0B 61 65 BB 0F 00	* status positions

TABLE VI—DISPLAY AREA IN PAGE 6 WITH TTT GRID

Loc.	Data							
06 00	00	00	18	00	18	00	00	00
06 08	00	00	18	00	18	00	00	00
06 10	00	00	18	00	18	00	00	00
06 18	00	00	18	00	18	00	00	00
06 20	00	00	18	00	18	00	00	00
06 28	00	00	18	00	18	00	00	00
06 30	00	00	18	00	18	00	00	00
06 38	00	00	18	00	18	00	00	00
06 40	00	00	18	00	18	00	00	00
06 48	00	00	18	00	18	00	00	00
06 50	07	FF	FF	FF	FF	FF	E0	00
06 58	00	00	18	00	18	00	00	00
06 60	00	00	18	00	18	00	00	00
06 68	00	00	18	00	18	00	00	00
06 70	00	00	18	00	18	00	00	00
06 78	00	00	18	00	18	00	00	00
06 80	00	00	18	00	18	00	00	00
06 88	00	00	18	00	18	00	00	00
06 90	00	00	18	00	18	00	00	00
06 98	00	00	18	00	18	00	00	00
06 A0	00	00	18	00	18	00	00	00
06 A8	07	FF	FF	FF	FF	FF	E0	00
06 B0	00	00	18	00	18	00	00	00
06 B8	00	00	18	00	18	00	00	00
06 C0	00	00	18	00	18	00	00	00
06 C8	00	00	18	00	18	00	00	00
06 D0	00	00	18	00	18	00	00	00
06 D8	00	00	18	00	18	00	00	00
06 E0	00	00	18	00	18	00	00	00
06 E8	00	00	18	00	18	00	00	00
06 F0	00	00	18	00	18	00	00	00
06 F8	00	00	18	00	18	00	00	00

TABLE V—INTERRUPT ROUTINE FOR THE 1861 CHIP

Loc.	Program
05 00	72 70 22 78 22 52
05 06	C4 C4 C4
05 09	F8 06 B0 F8 00 A0
05 0F	80 E2
05 11	E2 20 A0 E2 20 A0
05 17	E2 20 A0
05 1A	3C 0F 1B 30 00

TABLE VII—REGISTER USE IN TTT PROGRAM

Register	Use
0	Interrupt DMA pointer
1	Interrupt routine
2	Stack pointer
3	Page 5 pointer
4	Character print subr
5	'From' char pointer
6	'To' char pointer
7	Page 5 pointer
8	N, number of plays
9	Temp storage pointer
A	L-W-D subr
B	Refresh count
C	Temp storage
D	Delay subr
E	Win, Lose, Draw
F	Main program

TABLE VIII—INITIALIZATION OF THE REGISTERS

Loc.	Program						
05 20	F8	05	BF	F8	27	AF	DF
05 27	F8	05	B1	B2	B3		
05 2C	B7	B9	BA	BD	F8	06	
05 32	B6	BE	F8	07	B4	B5	
05 38	F8	02	A1	F8	BF	A2	
05 3E	F8	E0	A3	F8	01	A4	
05 44	F8	FF	A9	F8	61	AA	
05 4A	F8	F1	AD	C0	04	03	

program returns to "any place" and randomly selects its next response.

If the light pen is placed in front of an already lit position when it is the opponent's turn to play, a ? will be displayed. As soon as the pen is moved to a dark position, the program will start scanning all unplayed positions to display the X. If the light pen is away from all positions, a ? will be displayed. The ? is erased when a valid X is written.

Program. The program consists of the Main Program shown in Table I and four subroutines. These are Character Print in Table II, Delay in Table III, L-W-D (Lose-Win-Defensive) in Table IV, and Video Chip Interrupt in Table V.

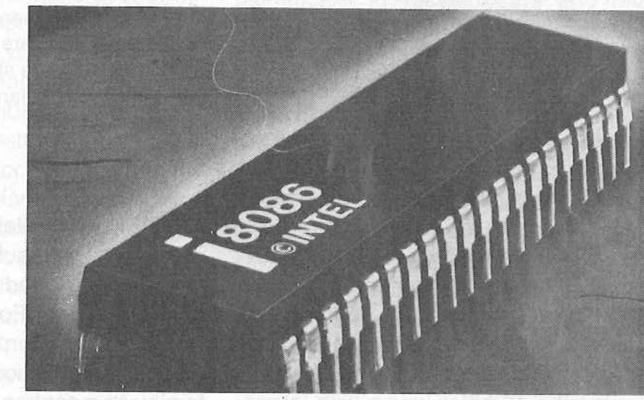
The character-print subroutine can print a blank or any of the seven characters X, O, ?, W, L, D, or P in any of the 10 positions on the screen. The delay subroutine is used for various delays including the 2-second delay before displaying an O. The L-W-D subroutine is used for the three tests to determine if the computer has lost, can win, or must play defensively. For each row, column, or diagonal, the computer examines each of the three positions. If it is an X, nine is added to a register; if a O, three is added; and if blank, nothing is added. Thus, a total of 27 indicates a loss, a total of 6 a potential win, and a total of 18 indicates that defensive action is required. The video interrupt instruction is conventional.

The random position for "any place" is determined by the B register, which is used to count the number of refreshes of the TV screen. Whenever "any place" is entered, the program takes the modulo-9 value of this count for the position to be played. If that position is already occupied, another number is generated and the process is repeated until an empty position is found.

The use of the various registers is shown in Table VII.

The first seven bytes of the register initialization section (Table VIII) makes the switch if the program register upon entry is not F. If it is F, this part of the initialization must be skipped. ◇

16-Bit VS 8-Bit Microprocessors



8086 CPU is part of Intel's new 16-bit family.

BY GORDON LETWIN & HAMPTON MILLER
Heath Co.

IT IS OUR purpose here to provide a brief comparison of the recently introduced 16-bit microprocessors with their predecessors, the 8-bit CPU's. While we cannot do full justice to a complete comparison of all aspects of the two types, some broad generalizations can be made that will give a good idea of what to expect from the new 16-bit machines.

The Differences. Microprocessors deal with elementary units of digital information called "bits." An 8-bit processor's registers and data paths are all eight bits wide. Therefore, it operates on data and instructions eight bits (one byte) at a time. This does not mean that an 8-bit processor cannot deal with data larger than eight bits wide. The data can be broken up into 8-bit bytes and processed one byte at a time.

Many processor instructions, such as branches and memory references, are 16 and 24 bits long and are known as 2- and 3-byte instructions. These processors may also contain a few instructions that deal directly with 16-bit values. However, these multibyte instructions and 16-bit operations are handled eight bits at a time and, therefore, take more time to execute than do single-byte instructions dealing with 8-bit values. Typical applications are process control and communication, tasks that require minimal computation.

In contrast to the 8-bit processor, the registers and data paths of a 16-bit processor are all 16-bits wide. Consequently, the 16-bit processor operates on data and instructions 16 bits at a time. Since the number of operations per second is roughly the same for both 8- and 16-bit processors, two times as much work gets done per instruction with a 16-bitter. Hence, a 16-bit processor can be much faster than an 8-bitter.

The 16-bit processor is more than a double-sized 8-bit device. Since the 16-bitter fetches instructions from memory 16 bits at a time, its instruction set usually includes 16-, 32-, and 48-bit instructions. The longer instructions allow a 16-bit machine to implement sophisticated instructions that tend to be more general-purpose to take better advantage of a given architecture than does the 8-bit device. For example, to interpret the BASIC statement A=B(4) with an 8-bit 8080A, the BASIC interpreter might execute the code as follows:

```
LXI   H,B
LXI   D,4
DAD   D    ;(HL)=address of value
MOV   E,M
INX   H
MOV   D,M  ;(DE)=value
XCHG
SHLD  A    ;store in A
```

On the other hand, a 16-bit computer such as an LSI-11 could use the following code:

```
MOV   #4,R1      ; (R1)=subscript
MOV   B(R1),A    ;look up and store
```

Note how the 16-bit machine avoids the clumsy and time-consuming memory-access gyrations required by the 8-bit device.

Although 8-bit devices deal with smaller values than do 16-bit devices, this does not mean that the former cannot perform all the computations that a 16-bitter can perform. It just requires more time because calculations must be processed in smaller pieces that must be "fetched" sequentially from memory. To perform the operation A=B-C with the 8080A requires the following code:

```
LHLD  C
XCHG
LHLD  B
MOV   A,L
SUB   E     ; subtract low 8 bits
MOV   L,A
MOV   A,H
SBB   D     ; subtract high 8 bits
MOV   H,A
SHLD  A     ; store result
```

PROCESSOR COMPARISON TABLE

Processor	Data width	Register count	Memory (bytes)	Task timing in microseconds Load	Subtract A=B-C	Code bytes
8080	8	7	65K	8	14	273	22
6800	8	3	65K	8	14	266	30
LSI-11	16	7	65K	7	3.5	42.7	20
PACE	16	3	65K	9	9	61.0	36
9900	16	16	65K	7.3	4.7	49.9	24
8086	16	7	1M	2.8	0.6	17.6	24
Z8000	16	14	8M	2.25	1.0	14.0	24

The timing examples shown are for the code as it would typically be executed in an assembly-language program, FORTRAN, or BASIC interpreter. These are not the fastest special-case times because the actual code is never optimized for adding two specific variables; instead, it is designed to add any two variables. As an example, the 8086 appears to be faster than the Z8000 because the 8086 can do register-to-register operations twice as fast. However, due to the architecture of the registers in the two processors, the 8086 must do many more operations than are required in the Z8000. To accurately compare the two processors, one should compare memory reference times of the 8086 to the register reference for the Z8000. The figures shown here represent the author's assessment of capabilities of processors listed rather than quotes from manufacturer specification sheets.

The following very short code is required by the LSI-11:

```
MOV  B,A  ;store B in A
SUB  C,A  ;subtract C from B
```

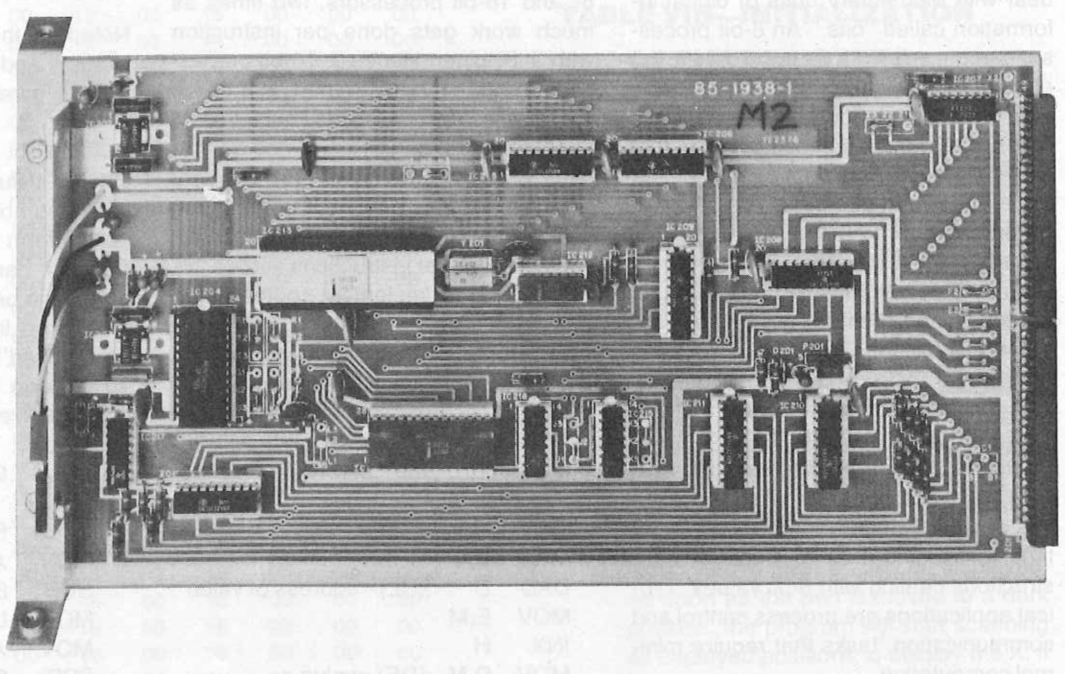
The table lists the memory and time required by several popular 8- and 16-bit processors to execute the statement C=B-A, where A, B, and C are 32-bit integers. Many BASIC interpreters make use of 32-bit or larger numbers to provide the necessary accuracy.

Advantages and Disadvantages. Since 16-bit devices operate on larger chunks of data and, thus, need perform fewer operations for a given task, their primary advantage is speed. This is usually several times faster than 8-bit devices. The larger the data items to be manipulated, the greater the advantage of the 16-bit design. This is true not only because the 16-bitter works with larger pieces, but also because the greater number of registers frequently reduce the number of memory references required.

A second major advantage of the 16-bit machine is the volume of *professional* software available for them, since many of these machines execute the instruction set used by well-established mini-computers. For example, the Digital Equipment Corp. (DEC) LSI-11 executes the instruction set for the PDP-11 family. As a consequence, with the proper peripherals, it can run the DEC PDP-11 operating systems, such as PTS, RT-11, RSX-11, etc.

Another advantage of the 16-bitter is that some 16-bit systems can accept peripherals designed for minicomputer members of the family. This provides the option of purchasing high-level printers, disks, etc., and obtaining service contracts on them as well.

A major disadvantage of the 16-bit system is its higher cost, which results from multichip configurations and relatively expensive support circuitry. Since memory is accessed in 16-bit words, a computer that uses a 16-bit processor requires 16-bit data paths. This means more bus interface, control circuitry, and connectors



This photo, and the one on the facing page, illustrate the difference between an 8-bit CPU board (above) as used in the Heath H8 and the 16-bit CPU.

Also, current 16-bit processors tend to use more memory for a given task than do 8-bit machines. A program being run in a 16-bit machine may require 10% to 20% more memory bytes than the same program being run in an 8-bit machine. In addition, there are not, at present, many peripheral devices available for 16-bit machines, so 8-bit computers have the advantage here.

Candidates for 16-Bitters. Users with large computational needs (so-called "number crunchers") should definitely consider a 16-bit machine. Avid game players will find that many advanced and sophisticated games require the extra speed that a 16-bit machine can provide.

Some 16-bit machines have available optional floating-point hardware that can speed up numerical calculations by a factor of 20 or more. Users who are or plan to be sophisticated programmers will want a 16-bitter because of its sophisticated instruction set. As shown in the foregoing examples, assembly-language programming for these machines is many times easier for 16-bit machines than it is for 8-bit machines. In addition, the more powerful addressing capabilities allow the programmer to take advantage of algorithms that might be difficult to implement on an 8-bit machine.

If you plan to use your computer system primarily for business applications, you should consider factors other than just raw CPU speed. A primary concern in business applications is the availability of peripherals and loads of business software that have been in use for many years and are, thus, completely debugged. If the system under consideration is software-compatible with a minicomputer, it can take advantage of the already existing software for that minicomputer.

Much of the existing software is inexpensive or in the public domain and, therefore, available for a copying charge. However, much of it may be available from only the manufacturer of the "parent" mini-computer line and can be quite expensive. If you are interested in transferring existing minicomputer software to a compatible microcomputer, you should investigate this area closely with the microcomputer manufacturer and the manufacturer of the parent minicomputer before making a purchase. Either or both companies may sell software and/or have user groups that distribute programs.

The Future. The future appears bright for the 16-bit computer. Most of the new generation of 16-bit microprocessors will be single-chip devices that, together with high-volume production, will bring chip prices to levels closer to current 8-bit processors. An important use of 16-bit μP will likely be in military, automobile and large-system applications.

Chip designers are also working on memory utilization, which is another important cost factor. Due to their advanced architecture, the new generation of 16-bit chips will generally require less memory for a given program than do present 8- and 16-bit devices.

The new 16-bit chips will also provide vastly improved performance. Many of them will be capable of running faster than many present-day minis that cost $60,000 or more.

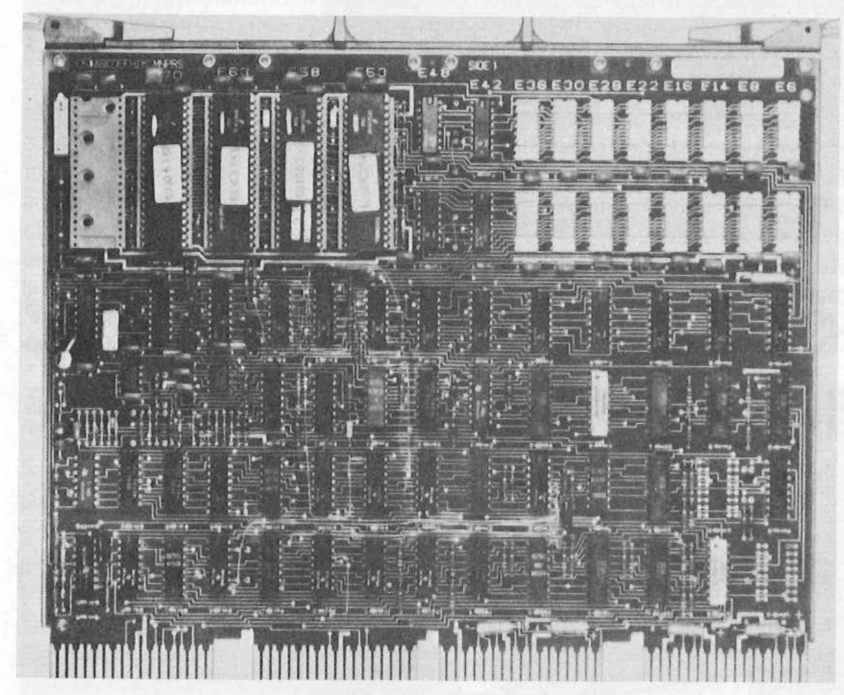
Semiconductor memory prices have plummeted in recent years and may drop even faster in the future as new memory devices (like bubble memories) come along. New 16-bit processor designs have anticipated this "problem" by providing the capability of addressing extended amounts of memory. Intel's 8086, for example, can handle up to 1 megabyte of memory, while Zilog's Z8000 can address up to 8 megabytes. The combination of processor speed and memory capacity makes these chips suitable for use in large multi-user computer systems as well as in smaller single-user systems.

Manufacturers also realize that customers will be reluctant to rewrite their expensive software (developed for their 8-bit machines) to use their new 16-bit processors. Most have, therefore, designed their new processors to be "upward-compatible" with their previous products. Hence, programs written for the 8080A can be reassembled to run with the 8086, and Z80 programs can be reassembled for the Z8000.

Conclusion. Although we've concentrated on the merits and features of 16-bit processors, the 8-bit processor is far from dead. It currently enjoys a wide customer base and an ever-increasing amount of software and peripherals, and will continue to do so for many years to come. Though they do not have the processing capability of the 16-bit systems, the 8-bitters are certainly fast enough for most home and small-business applications.

As 16-bitters come into their own, they will likely eclipse and supersede our present 8-bitters. This will happen when complex applications require more effective addressing and where memory requirements in larger systems become an important consideration. ◇

(Computer section continues on next page.)



The 16-bit CPU above, used in the Heath H11, is much denser than the 8-bit unit at left. It also carries 4K of RAM; but both have necessary support chips.

BY LESLIE SOLOMON
Technical Director

How to Care for Diskettes

Correct handling and storage of diskettes ensures good data retrieval.

FLOPPY-DISK systems are being used in growing numbers in personal computing systems. Since about 70K of data can be stored on a small 5¼" (13.3-cm) diskette, you should be aware that all of the work put into creating and storing long programs can be catastrophically destroyed by improper handling. This problem can be caused by a single act like smoking near a disk system, touching a diskette with a fingertip, bringing metal tools near a diskette, and even by having an audio system or radio receiver too close to the disk system.

Let us take a brief look at how a disk system works and how improper handling can cause a diskette to lose data.

Physical Makeup. As shown in Fig. 1, a floppy diskette consists of a relatively heavy cardboard (or other nonmetallic) jacket whose dimensions vary with the size of the diskette. Inside the jacket is the actual diskette itself. The diskette is made of very thin, flexible Mylar on which is deposited a layer of magnetically active material similar to that used on audio magnetic recording tape.

When the diskette is inserted into the disk-drive mechanism, the large hole at its center is engaged by a spindle and the disk inside the jacket is made to revolve at high speed. This is similar to the playing of a conventional record on its player, except that the read/write head in the disk-drive mechanism does not contact the diskette. Also, there are no grooves on a diskette as there are on a record. Instead, the disk drive's mechanical system uses computer commands to position the read/write head at the correct track.

A large track-access slot extends part way along one radius of the protective jacket. This slot permits the read/write head to access the magnetic surface of the diskette. A small hole near the large disk-drive hole allows a sensor in the disk-drive system to detect the rotational position of the diskette.

Handling a Diskette. Before removing a diskette from its jacket, read the instructions printed on the back of the jacket. Listed here are usually a lot of no-no's such as: do not touch the diskette's magnetic surface with fingers; keep the diskette away from magnetic fields; do not bend or fold the diskette; keep the diskette at a reasonable temperature; and keep the diskette in its protective jacket when not in use. Needless to say, heed the printed advice.

Since a diskette works because a magnetic track is laid down by the read/write head, the diskette's surface is extremely sensitive to magnetic fields. This means that a diskette should never be placed in a magnetic field. It is obvious that one would not intentionally bring a powerful magnet near a diskette, but accidents can occur if you are not on guard. For example, audio equipment and even a small pocket radio near a diskette can wreak havoc to the data on a diskette. Bear in mind that loudspeakers operate by powerful magnetic action and that the speaker's magnetic field does not stop at the surface of its enclosure or the plastic case of the radio.

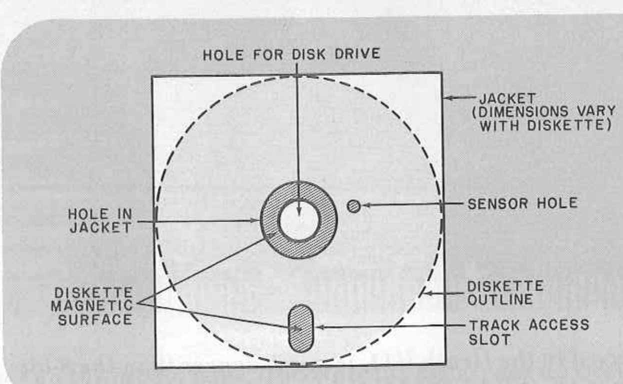
It is best to keep all metal tools away from diskettes because chances are that the tools have become magnetized during use. You can easily check this by seeing how many paper clips your favorite screwdriver picks up. But even if your tools do not pick up paper clips, this is no guarantee that they are not slightly magnetized. So, play it safe; keep all tools away from diskettes.

There is also the problem of the ac field that surrounds some transformers and turned-on soldering guns. This field is very similar to the ac field used in degaussing devices for audio tape heads. The degaussing effect of these fields can erase a diskette as fast as they degauss a tape head.

Foreign Matter. Although the illustration in Fig. 2 is not drawn to scale, it does serve to explain some important points. On a typical diskette, the magnetic coating is a mere 50 to 200 microinches thick and is deposited on a flexible 0.003" thick Mylar base. The read/write head, which has a curved surface close to the diskette, does not actually touch the magnetic surface but is maintained about 20 microinches away. As the diskette spins inside its jacket, the head-to-diskette speed can be as great as 140 mph!

A typical smoke particle can be up to 250 microinches thick, while a human hair is typically 0.003" thick. Hence, you can imagine the pounding that the very thin magnetic layer on the diskette gets if any of these particulants is between the head and the moving diskette.

Dust and lint particles can appear to be as big as boulders when compared to the gap between the read/write head and diskette. Since most of us have slight oil deposits on our fingertips, even touching an exposed diskette's surface can cause a loss of data. Bending or folding a diskette can produce serious "hills and valleys" that can disrupt data acceptance and delivery. Even a minute scratch can produce dropouts that will ruin the depositing and fetching of data.



Physical layout of a typical diskette package.

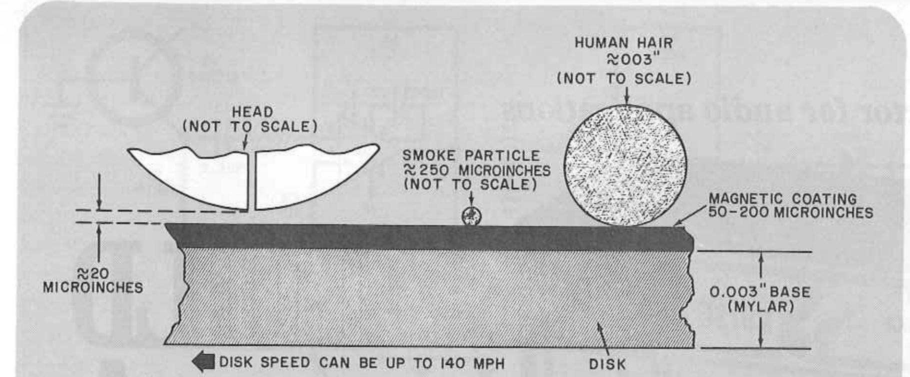


Fig. 2. Anything bigger than the 20-microinch head/diskette spacing can grind up the relatively soft magnetic coating and ruin a diskette. Head-to-diskette speed can reach 140 miles per hour.

Identification labelling of a diskette's jacket can cause problems if you are not careful. *Never* write with a ballpoint pen on a label that is already in place on a jacket. (The pressure required to write with a ballpoint pen can transmit enough energy through the label and jacket to damage the relatively soft diskette surface.) In fact, it is a good idea to avoid using anything but a felt-tipped pen to write lables because these pens write cleanly, dry quickly, and cannot contaminate the surface of the diskette. Always write on your labels before attaching them to the diskette's jacket.

Do not attempt to erase a label after it is fixed to a diskette's jacket. Eraser particles can cause a lot of damage in the area between the diskette and read/write head, no matter how much care you exercise. It is far better and safer to simply make up another label and place it over the old one.

If possible, try to keep the read/write head scrupulously clean. Consult the manual supplied with your disk system for the proper procedure to accomplish this. Be especially careful to avoid getting oil or grease on the head because they are contaminants themselves and they tend to attract dirt.

Storing Diskettes. Never stack diskettes on each other for storage. The proper way to store diskettes is to stand them vertically, preferably inside a metal cabinet to keep them safe from stray magnetic fields. The temperature in the room in which diskettes are stored should be between 60° and 90° F, with the humidity between 10% and 90%.

If you intend to store diskettes for long intervals, place them in their original heavy shipping boxes and store them just about anywhere the temperature is between −40° and +150° F. After prolonged storage, be sure to leave the diskettes in the room where they are to be used for at least 24 hours to allow them to stabilize.

On final word: store important diskettes inside antimagnetic, fire-resistant, and waterproof containers. For further protection, you might consider making duplicates of your important diskettes and storing them separately. ◇