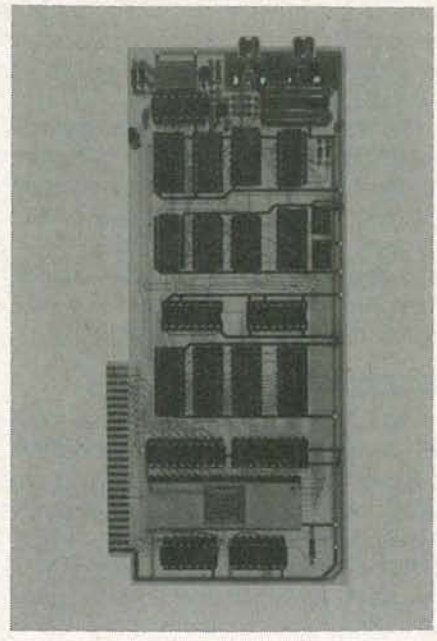
BUILD THE GRAFEX-32

PART 1



A 640 by 400 graphics adapter for the Apple II.

Ray Dahlby
Dep't. 255
Box C-34069
Seattle, WA 98124-1069

■The Apple II's graphics resolution of 280 by 192 pixels can no longer be considered the "state of the art." The newer Apple *Macintosh* and *Lisa* computers display their bit-mapped images from up to 32K bytes of memory, over four times that used for display on the Apple II.

Recently, the availability of 64K dynamic RAM chips and VLSI graphics controllers at reasonable prices has made it feasible to expand the Apple II's graphics resolution using a simple plug-in circuit board. The Grafex-32 circuit board uses four 16K by 4 bit RAM chips along with a 7220 graphics controller to display 32K bytes of memory as a bit-map of 640 by 400 pixels. The board can be expanded to 128K bytes by plugging in 64K by 4 bit RAM chips and three such boards can be installed in a system to provide the red, green, and blue signals for color graphics displays.

Our design tradeoffs favor circuit simplicity with the option of expansion later, over a complex design incorporating features which might not have been used by everyone. Although the basic circuit consists of just 20 chips, it displays a monochrome resolution of 256,000 pixels on a standard Apple Monitor. Since this baseline performance exceeds the graphics resolution of the Macintosh computer, I decided not to increase the complexity, and thus the cost, of the basic circuit board and thereby provide hardware "hooks" by which the design can be upgraded by those users with higher resolution or color requirements. This design philosophy seems consistent with that of the Apple II itself.

The Apple II

The Apple II/IIe line owes much of its success to the expandability afforded by its expansion slots. In the six or so years that this computer has been on the market, hardware manufacturers have responded to the challenge of filling those slots by developing products to expand the machine and now, with a multitude of add-on cards available, it is surprising to find an area which has not been adequately served.

One such area seems to be the expansion of the Apple II's graphics resolution. Although a few cards were produced to add sprite-oriented graphics and luminance attributes to the Apple II, these did not significantly increase the number of pixels on the screen. Part of this lapse can probably be attributed to the introduction of the I.B.M *PC* which has distracted the focus of third-party hardware manufacturers at a time when VLSI graphics chips and cheap memory make Apple II graphics expansion an easy matter.

The Apple II line can benefit from new graphics technology in other ways than increased screen resolution. The video design used in the Apple II maps the display memory into the 6502 micro-processor's address space in a technique called "memory-mapped video." This was a good choice at the time because it allowed the 6502 full access to the screen memory using all memory reference instructions and addressing modes just as if it were addressing normal system RAM. In fact, if the high resolution screens are not used, the memory space allocated to them can be used for program and data storage. The design of the Apple II limits the graphics resolution for a least two reasons, one being the restricted bandwidth of the home color televisions it was assumed that Apple owners would be using as their display devices. A second reason is limited amount of memory space which can be addressed by the Apple's 6502 microprocessor. This chip has 16 address lines which allows it to address only 65,536 bytes so a memory-mapped video design using half of this precious space just would not have been practical.

Memory-mapped video design is not limited to low resolution graphics as evidenced by the newer Apple *Macintosh* and *Lisa* machines. The 68000 microprocessor used in these computers has a 24-bit address bus and can directly address more than 16M

bytes of system memory. A large graphics RAM mapped into this address space does not represent a significant fraction of the total available for program and data storage.

The answer to improving the Apple II's graphics resolution without using up all of its memory is to keep the graphics RAM separate from the system RAM. The 6502 won't be able to directly access to display RAM but there are new chips optimized for managing large bit-mapped memories. The 6502 actually benefits from the increased program available by not having part of its system RAM allocated to graphics.

The 7220 GDC

The 7220 GDC (*G*raphics *D*isplay *C*ontroller) from NEC is designed to handle the repetitive tasks required in figure, line, and character drawing on a raster scan CRT. Unlike previous CRT controller chips such as the Motorola 6845 whose tasks were limited to display refresh and video synchronization. The 7220 has an instruction set which enables it to read, modify, and write data in the display memory. Positioned between the system microprocessor bus and the display memory, it responds to instructions passed to it and draws figures without processor intervention. Since the GDC can handle much of the repetitive pixel drawing and modification tasks, the bandwidth requirement of the microprocessor/display memory path is greatly reduced. Most of the data sent from the microprocessor during vector and geometric shape drawing will be in the form of commands and parameters sent directly to the GDC which then interprets them into pixel-level operations to be carried out over the high-bandwidth GDC/display memory path. Its pipelined architecture is optimized for such graphics manipulation and it handles these tasks with great speed. For example, a 7220 running at a clock frequency of 5 Mhz can draw a figure at the rate of 800 ns per pixel. This speed is independent of the type of figure being drawn and is much faster than a general purpose microprocessor, such as the 6502, handling the same task.

The 7220 was chosen for this design because its 8 bit uP data bus interfaces nicely with the Apple's 8 bit 6502 and its 16 bit video data bus and 18 bit video address bus allows it to accomodate large bit-mapped display memories without impinging on the limited 16 bit addressing of the 6502. It also handles dynamic RAM refresh and video sync generation. This part is housed in a 40-pin ceramic package and is fabricated in 3um NMOS. It encompasses the equivalent of over 13,000 transistors.

The display memory

The 7220 requires its display memory to be organized in 16-bit words. The most common 64K dynamic RAM chips, such as the 4164, are organized as 65,536 locations of 1 bit each so 16 of these parts are

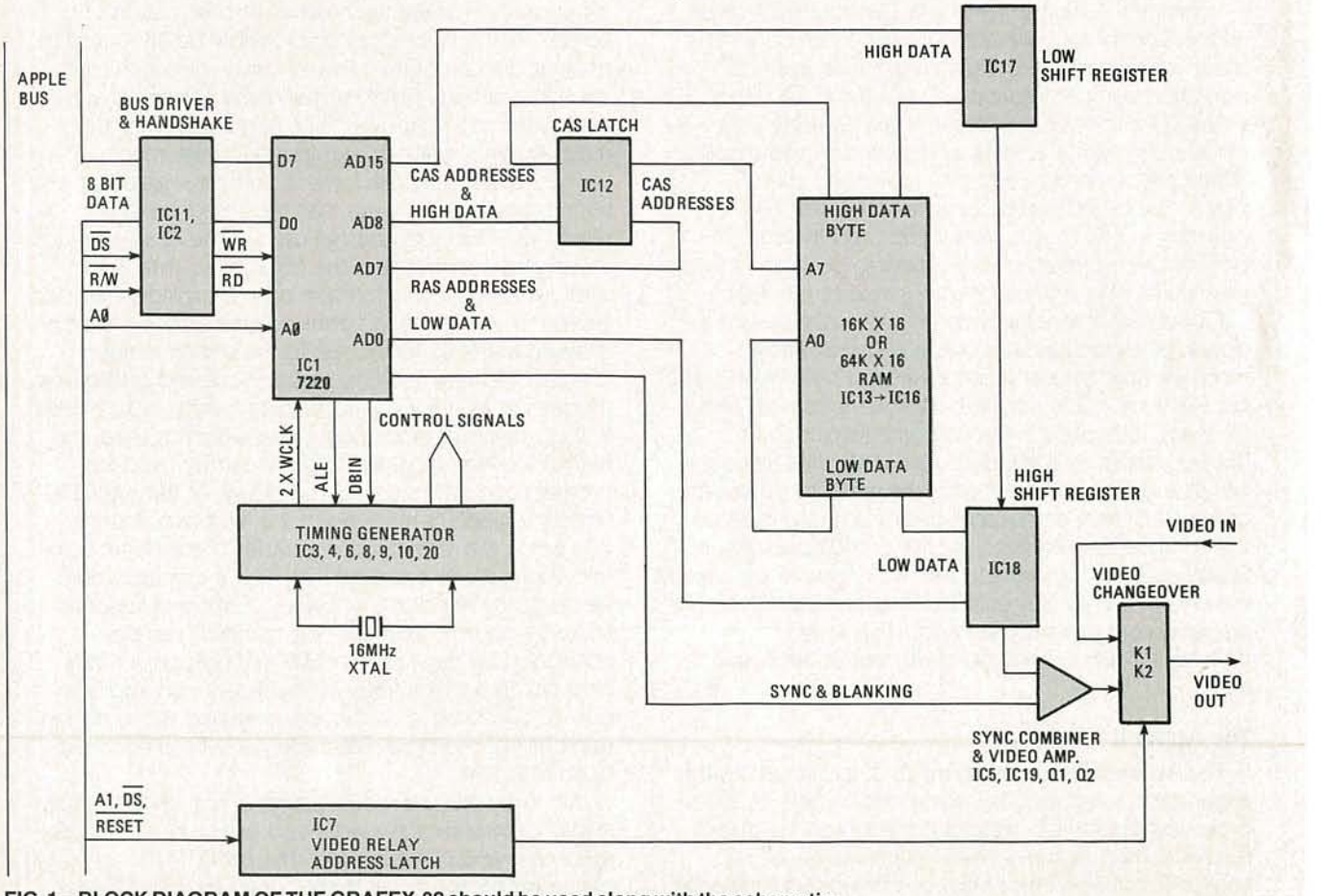


FIG. 1—BLOCK DIAGRAM OF THE GRAFEX-32 should be used along with the schematic diagram to help understand the theory of operation.

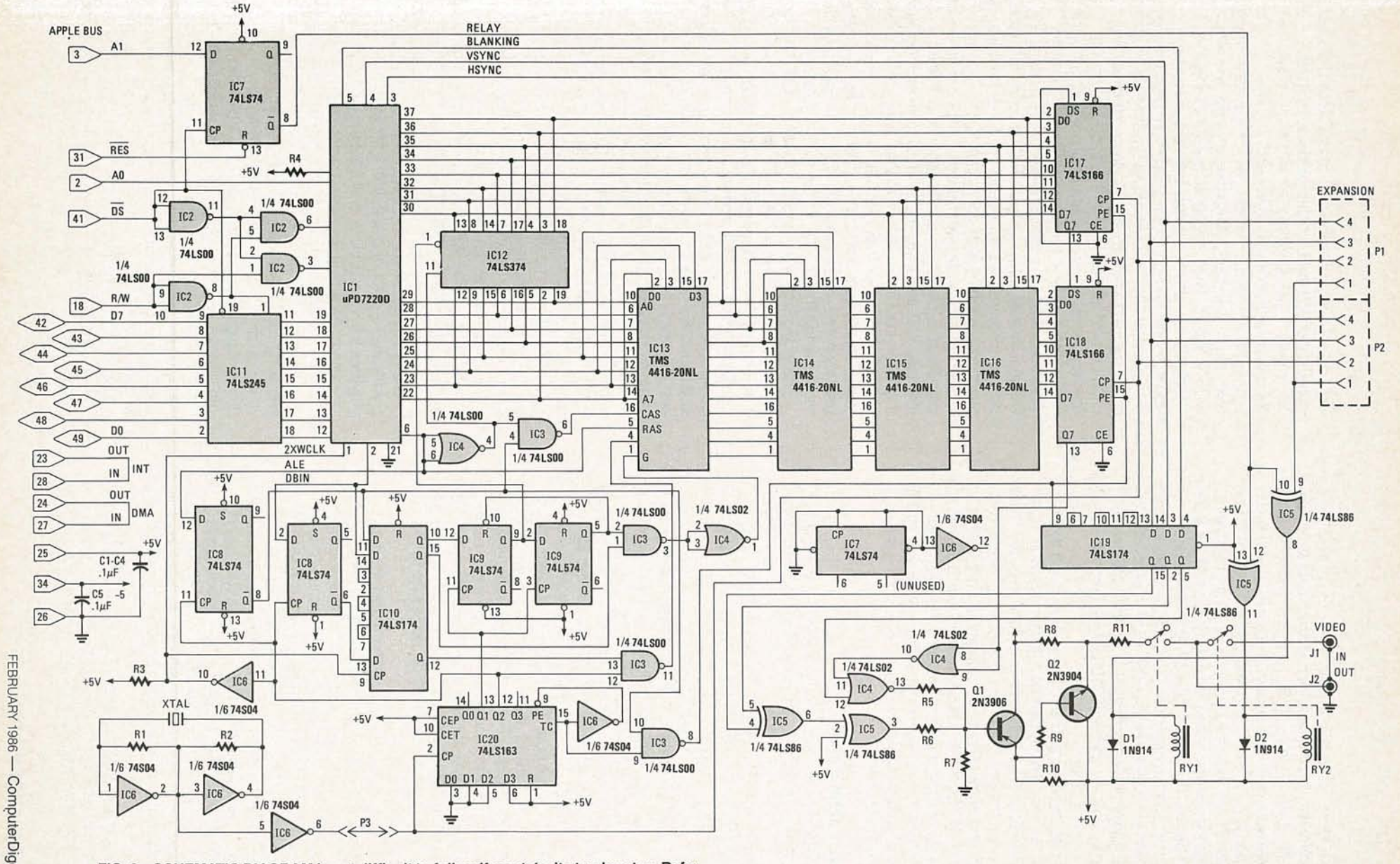


FIG. 2—SCHEMATIC DIAGRAM is not difficult to follow if you take it step-by-step. Refer to this while reading the text and you'll have an almost-immediate grasp of what is going on from input through output!

required to assemble a 16-bit memory system. There is another 64K RAM chip which is organized as 16,384 locations of 4 bits each. This part, designated the 4416, is available from Texas Instruments, Inmos, and Fujitsu, among others. The 16 bit data bus requirement of a 7220 can be met using just four of these chips, thus reducing power dissipation and printed circuit board complexity. The use of the 4416 actually helps to reduce chip count further as it has a separate output enable pin which can be used to tri-state its data lines. This feature allows very tight coupling between the display memory and the 7220 which helps to simplify P.C. layout further.

Texas Instruments has recently announced the 4464 and NEC, the 41254, dynamic RAM chips which are organized as 65,536 locations of four bits each. These parts are housed in the same 18-pin package as the 4416 and allow easy system upgrades. The Grafex circuit, for example, has been designed to address the additional memory provided by these parts and expansion of the Grafex circuit from 32 Kbytes to 128 Kbytes is accomplished simply by replacing the 4 display RAM chips.

Circuit description

Referring to the block diagram of Figure 1 and the schematic of Figure 2, it can be seen that the Grafex circuit can be broken down into five sections. These sections are a) the uP bus interface, b) the 7220, c) the display memory, d) the video output circuitry and e) the timing generator.

The 7220, IC1, communicates with the host microprocessor over its eight bidirectional data lines. IC11, A 74LS245 is used to provide buffering and IC2, a 74LS00 is used to convert the Apple's R/W and DS signals into the 8080-type RD and WR signals used by the 7220. The Grafex circuit uses four of the 16 addresses assigned to the peripheral slot in which it is installed in the Apple II. These addresses are used to select the command and parameter registers of the 7220 and to control the video changeover relay latch, IC7. The addresses and their functions are listed in Table 1 which will appear in a subsequent issue. The reset line on the Apple bus is connected to this video relay latch to force a default on power up to the Apple video signal.

The 7220 communicates with the display memory over its 16 multiplexed address/data lines, labelled AD0-AD15 on the block diagram. The CAS latch, IC12, a 74LS374, is used to latch the high order 8 bits of addresses and then to sequence them onto the 8 RAS/CAS address lines of the display memory. This is a tight-coupled design from the standpoint of data and address multiplexing. The low address/data path from the 7220 carries, in sequence, the row addresses, column addresses, and finally, the low byte of display memory data. The high byte of display memory data is carried over a separate path directly to the 7220 AD8-AD15 pins.

The display memory, IC13-IC16, has a multiplexed 8-bit address bus and a 16-bit data I/O with separate output enable. Like other dynamic RAM arrays, the eight row address bits are first strobed into the on-chip latches, then the eight column address bits are presented on the address lines and finally, these are strobed into the memory. After access time specifications have been met, the data is read during a read cycle or written during a write cycle. The 4416 and 4464 parts have a separate output enable pin which allows data to be read from the selected address location but not presented on the output pins until needed. This "G" pin allows a fast (40 ns) turn-on of the output buffers to supply data to the external circuit when needed later in the read cycle. In this manner, the eight row/column address lines can serve also as the low 8-bit data corridor to and from the 7220 and video shift register without the need for an external tri-state buffer. This kind of coupling facilitates printed circuit board layout and improves reliability by reducing the circuit inter-connections.

The 16 bits of data read from the display memory are presented to two 8-bit parallel in/serial out, shift registers, IC17 and IC18. These 74LS166 shifters serialize the 16-bit data into a video bit stream which is clocked at the 16 MHz dot-clock rate. The blanking, Hsync, and Vsync signals from the 7220 are brought into line with data by means of the 74LS174, IC19. It is loaded by the same load/shift signal as are the shift registers.

The video-bit stream is gated with the 7220 blanking signal by IC4 and then mixed with the composite sync provided by exclusive-or gate IC5. The video amplifier, consisting of Q1 and Q2 provides a standard 1 volt P-P composite video signal into 75 ohms. This composite video signal is routed to the video changeover relay, K1 and K2, which selects either the Grafex video or an external input as the source for the video monitor. When the Grafex board is installed in an Apple II, the external input is normally connected to the Apple's video output connector and the changeover relays output connected to the system video monitor. Alternatively, two monitors could be used to simultaneously display Grafex and Apple video. The video changeover relay is software actuated and defaults on reset to the external input, allowing the Apple system to be operated normally after power up. In this way, unless the Grafex board is specifically addressed, a user need never be aware of its existence.

The timing generator, comprised of the 16 MHz crystal and IC3, IC4, IC6, IC8, IC9, IC10, and IC20 provides the various clock and control signals used in the system. All timing is derived synchronously from the 16 MHz clock. The timing generator has two modes of operation depending on whether the 7220 is executing a display cycle or a RMW, (read, modify, write), cycle. These two types of cycles are differentiated by the DBIN pin of the 7220. The 7220, in turn, uses as its master clock, a 2 MHz signal labelled 2Xwc1k. This clock, as its name implies, runs at twice the display word rate and all internal timing of the 7220 is derived from this signal since 16 pixels comprise one 16 bit word of display memory, the 2Xwc1k used in this design is 2 x 16 x 65 ns = 500 ns or 2 MHz.

That's all the space for now. We'll continue this article next month.

BUILD THE GRAFEX-32 PART 2

RAY DAHLBY

This article, begun in February is continued here.

A 7220 display cycle requires 2-2Xwclk cycles and a RMW cycle requires 4-2Xwc1k cycles. The sequence of events which takes place during a RMW cycle is outlined below. The four 2Xwc1k intervals are referred to as e1, e2, e3, and e4 as per the timing diagram, Figure 3.

The intervals description follows:

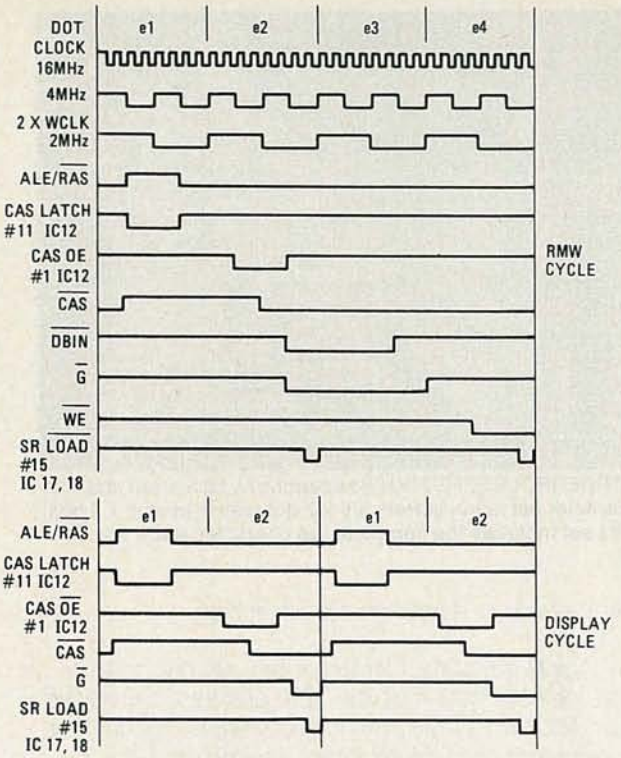


FIG. 3—TIMING DIAGRAM illustrates timing steps and sequences. See text for further information.

e1—The 7220 begins to output the display memory address on the 16 AD lines; ALE goes low to indicate this address is valid. This signal (also called RAS) strobes the low 8 address bits into the display memory and latches the high 8 address bits into the CAS latch, IC12.

e2—The 7220 tri-states the AD lines and DBIN goes low to indicate that a RMW cycle is in progress. The CAS latch OE pin is brought low to present the 8 column address bits to the display memory and then, the CAS line is brought low to strobe this address into the display RAM chips. The G pin of the memory is brought low to allow the read data to be presented to the 7220. The read data is loaded into the video shift registers.

e3—The 7220 reads the 16 bit data from the display

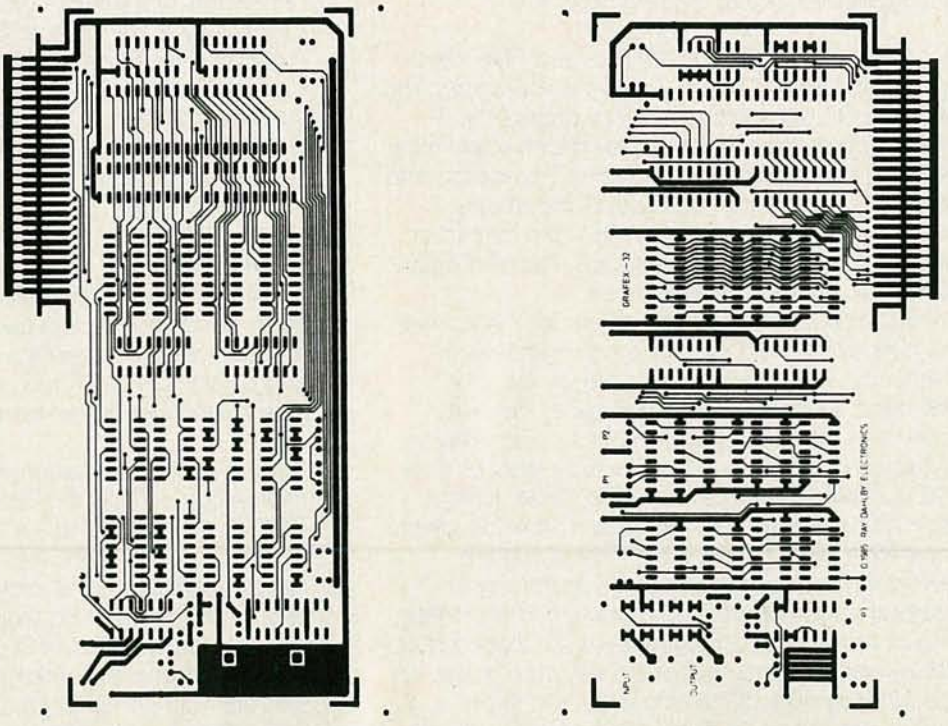
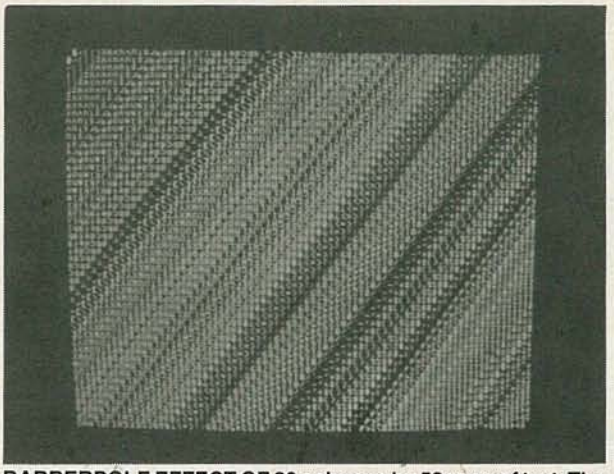
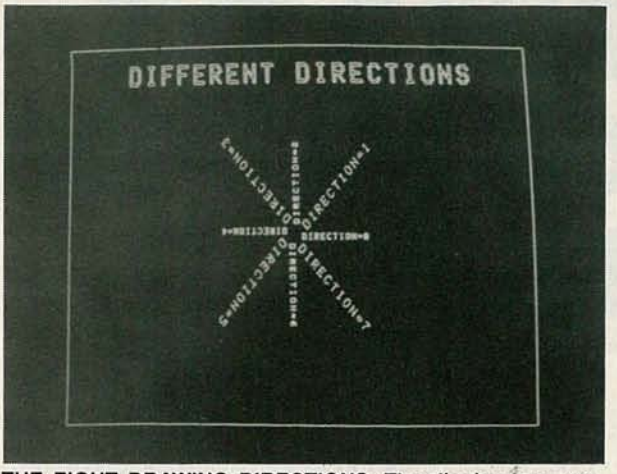


FIG. 4—PRINTED CIRCUIT BOARD is shown half size for those who wish to make their own. Due to space restrictions, board had to be reduced 50%. Be sure to have these drawings photographically enlarged 200% before making board. Component side is shown at left.



BARBERPOLE EFFECT OF 80 columns by 50 rows of text. The character set is made from a 5 x 7 dot matrix in an 8 x 8 box. This set includes the complete 128 character ASCII set.



THE EIGHT DRAWING DIRECTIONS. The display was produced by writing the string "DIFFERENT DIRECTIONS" at a magnification factor of 2, then writing it again with the x and y starting position shifted by 1 pixel and the mode set to complement.

memory and performs the modifications.

e4—The G pin of the memory is brought high to tri-state the RAM data lines, the 7220 presents its modified data onto the AD lines, the memory WE is brought low to write the modified data back into the display memory. The modified data is loaded into the video shift registers.

The two 2Xwc1k intervals which make up a display cycle are outlined below. Again, e1 and e2 refer to the 2Xwc1k intervals labelled on the timing diagram.

e1—The 7220 begins to output the display memory address on the 16 AD lines; ALE goes low to indicate this address is valid. This signal strobes the low 8 address bits into the display memory and latches the high 8 address bits into the CAS latch, IC12.

e2—The 7220 tri-states the AD lines and DBIN stays high to indicate that a display cycle is in progress. The CAS latch OE pin is brought low to present the 8 column address bits to the display memory and then, the CAS line is brought low to strobe this address into the display RAM chips. The G pin of the display memory is brought low to allow the read data to be presented to the video shift registers. The read data is loaded into the video shift registers.

Notice that the video shift registers are loaded with new data every two 2Xwc1k interval regardless of whether the 7220 is executing a display cycle or a RMW cycle. In the case of a RMW cycle, the data loaded into the shift registers will not correspond to valid screen data and will cause visible glitches if the 7220 is allowed to access the screen RAM during active video intervals. The 7220 has a provision which allows RMW cycle to take place only during the blanked Vsync and Hsync intervals preventing a disturbed screen display while drawing is in progress. Although the video shift registers are still loaded during RMW operations, these cycles are restricted to the approximately 30% of the time when the screen is blanked, preventing the false data from being seen.

When two or more boards are installed in a system, they are daisy-chained together by means of the expansion connectors, P1 and P2. These connectors carry the 16 MHZ dot clock and the 7220 video sync signals from the board designated to be the master, to the slave boards. The jumper on P3 is installed only on the board acting as the master. When initiallized by software, the slave 7220's synchronize their timing with the master so that all of the 7220's run in phase with each other. In this way, three Grafex boards installed in a system can each drive one gun of an RGB color monitor. The composite sync is carried on the video for those monitors capable of accepting sync on the green input. RGB monitors having external synchronization inputs can be driven from one of the expansion connectors of a Grafex board. Specific information explaining the use of three Grafex boards with RGB monitors is included with the color software package available from my company. Please check with the source mentioned at the end of this article for price and delivery information. Figures 4 and 5 provide necessary information should you prefer to fabricate your own boards.

Programming the 7220

After power up, the 7220 must be initiallized by a series of commands and parameters to configure it for the type of display desired. Usually, these commands and parameters are stored in a table which the initialization routine can refer to and then pass to the 7220.

The path for information flow between the host microprocessor and the 7220 is the first-in first-out (FIFO) buffer internal to the 7220. Commands and parameters and loaded into this buffer by the host and removed at the other end by the GDC's command processor. Care must be taken by the programmer to avoid overflowing the FIFO buffer with data faster than the GDC empties it. For this purpose, the GDC has a status register containing bits which indicate when the FIFO buffer is full or empty and also when data is ready

TABLE 1

Address	Read	Write	Video Relay
CONO	Status Register	Parameter	Apple Video Displayed
CON1	FIFO	Command	
CON2	Status Register	Parameter	Grafex Video Displayed
CON3	FIFO	Command	

to be read by the host microprocessor. Referring to Table 1, the status register can be seen to be mapped into the Apple's expansion slot area and can be read at any time. Other bits in this register indicate the state of the Vsync and Hsync video timing counters to allow smooth scrolling and other effects needing software synchronization to the video field rate.

The first command issued after power up is the Reset command. This command is interpreted by special hardware ahead of the FIFO to ensure that the internal registers, FIFO buffer, and command processor of the GDC are reset to their idle state prior to the initialization commands and parameters which follow. Normally, it is a good idea to check the status register for a FIFO FULL condition before each byte is output. On power up, the flags in the status register are not meaningful, so the RESET command must be issued before attempting to read the status register or load other commands and parameters into the GDC.

A typical initialization program is shown in Listing 1.

:L

Listing I

```
 1 IO        EQU  $FE
 2           ORG  $0300
 3           LDA  ##$F2      ;SLOT7
 4           STA  IO
 5           LDA  ##$C0
 6           STA  IO+1
 7 INIT      LDX  #00
 8 LOOP      LDA  TABLE1,X
 9           LDY  TABLE2,X
10           STA  (IO),Y
11           INX
12           CPX  #30
13           BNE  LOOP
14           RTS
15 TABLE1    DFB  $00,$1F,$26,$04,$1A
   ,$0B,$1E,$C8,$40,$6F,$47,$28,$70,
   $00,$00,$00,$19,$4B,$00,$C0,$00,$
   46,$00,$78,$FF,$FF,$23,$4C,$10,$6
   B
16 TABLE2    DFB  1,0,0,0,0,0,0,0,0,1
   ,1,0,1,0,0,0,0,1,0,0,0,1,0,1,0,0,
   1,1,0,1
```

GRAFEX-32 Power and Ground Pin-outs

	+5volts	ground
IC1	40	20
IC2	14	7
IC3	14	7
IC4	14	7
IC5	14	7
IC6	14	7
IC7	14	7
IC8	14	7
IC9	14	7
IC10	16	8
IC11	20	10
IC12	20	10
IC13	9	18
IC14	9	18
IC15	9	18
IC16	9	18
IC17	16	8
IC18	16	8
IC19	16	8
IC20	16	8

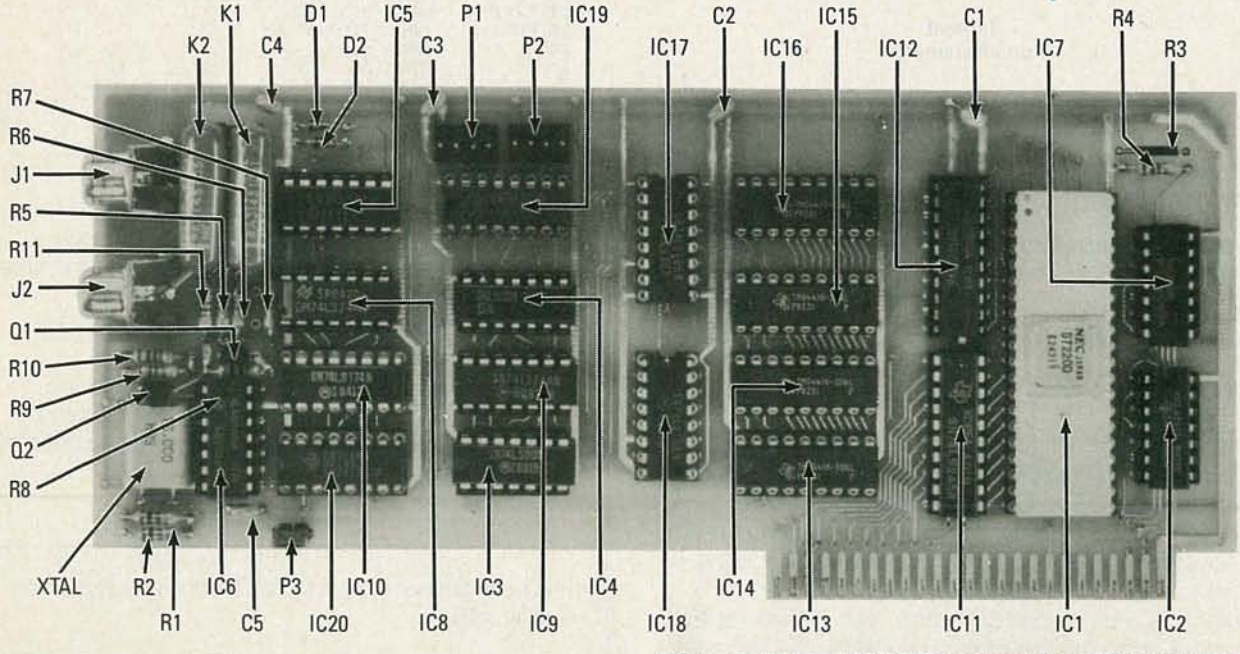


FIG. 5—PARTS PLACEMENT is shown above. No components are mounted on reverse side of board.

Listing II

```
:L
  1 IO      EQU  $FE
  2         ORG  $0300
  3         LDA  #$F2       ;SLOT7
  4         STA  IO
  5         LDA  #$C0
  6         STA  IO+1
  7         JSR  FIFO       ;MAKE SURE FIFO IS EMPTY /
  8         LDY  #01
  9         LDA  #$49       ;CURSOR COMMAND
 10         STA  (IO),Y
 11         DEY
 12         LDA  #00
 13         STA  (IO),Y
 14         STA  (IO),Y
 15         INY
 16         LDA  #$4A       ;MASK COMMAND
 17         STA  (IO),Y
 18         DEY
 19         LDA  #$FF
 20         STA  (IO),Y
 21         STA  (IO),Y
 22         INY
 23         LDA  #$0F       ;SYNC COMMAND
 24         STA  (IO),Y
 25         DEY
 26         LDA  #$0F       ;FAST MODE
 27         STA  (IO),Y
 28         LDX  #$00
 29 LOOP    LDY  #01
 30         LDA  #$4C       ;FIGS COMMAND
 31         STA  (IO),Y
 32         DEY
 33         LDA  #$02
 34         STA  (IO),Y
 35         LDA  #$FF
 36         STA  (IO),Y
 37         LDA  #$3F
 38         STA  (IO),Y
 39         INY
 40         LDA  #$22       ;WDAT MODE
 41         STA  (IO),Y
 42         DEY
 43         LDA  #$FF
 44         STA  (IO),Y
 45         STA  (IO),Y
 46         JSR  FIFO       ;MAKE SURE FIFO IS EMPTY
 47         INX
 48         CPX  #$01       ;CHANGE TO #$04 FOR 128K
 49         BNE  LOOP
 50         RTS
 51 FIFO    LDY  #00
 52 WAIT    LDA  (IO),Y
 53         AND  #$04
 54         BEQ  WAIT
 55         RTS
```

Table II
Video Parameters

Active Line =	40μs
HFP =	7μs
HBP =	12μs
HSYNC =	6μs
Total Line Time	64μs
Active lines per video field =	200 lines
VFP =	30 lines
VBP =	16 lines
Vsync =	16 lines
Total lines per field =	262 lines

Video field rate = 1/64μs × 262 lines) = 59.637 Hz

This sequence configures the GDC to generate 640 by 400 interlaced video with the video timing parameters given in Table II. The GDC is designated as master, dynamic RAM refresh is enabled, and transparent mode is selected (the GDC is allowed to draw only during blanked screen intervals). Also the entire screen is defined as a bit-mapped graphics area with the screen window set to the top of memory. When this program is run, the 7220 begins outputting video to the system

Listing III

```
:L
  1 IO      EQU  $FE
  2         ORG  $300
  3         LDA  #$F2       ;SLOT7
  4         STA  IO
  5         LDA  #$C0
  6         STA  IO+1
  7         JSR  FIFO       ;MAKE SURE FIFO IS EMPTY '
  8         LDY  #01
  9         LDA  #$23       ;WDAT MODE
 10         STA  (IO),Y
 11         LDA  #$78       ;PATTERN RAM
 12         STA  (IO),Y
 13         DEY
 14         LDA  #$FF
 15         STA  (IO),Y
 16         STA  (IO),Y
 17         INY
 18         LDA  #$49       ;CURSOR COMMAND
 19         STA  (IO),Y
 20         DEY
 21         LDA  #$81
 22         STA  (IO),Y
 23         LDA  #$17
 24         STA  (IO),Y
 25         LDA  #$00
 26         STA  (IO),Y
 27         INY
 28         LDA  #$4C       ;FIGS COMMAND
 29         STA  (IO),Y
 30         DEY
 31         LDA  #$40
 32         STA  (IO),Y
 33         LDA  #$03
 34         STA  (IO),Y
 35         LDA  #$00
 36         STA  (IO),Y
 37         LDA  #$63
 38         STA  (IO),Y
 39         LDA  #$00
 40         STA  (IO),Y
 41         LDA  #$63
 42         STA  (IO),Y
 43         LDA  #$00
 44         STA  (IO),Y
 45         LDA  #$FF
 46         STA  (IO),Y
 47         LDA  #$3F
 48         STA  (IO),Y
 49         LDA  #$63
 50         STA  (IO),Y
 51         LDA  #$00
 52         STA  (IO),Y
 53         INY
 54         LDA  #$6C       ;FIGD COMMAND
 55         STA  (IO),Y
 56         RTS
 57 FIFO    LDY  #00
 58 WAIT    LDA  (IO),Y
 59         AND  #$04
 60         BEQ  WAIT
 61         RTS
```

CRT monitor from the random power-up data in the display memory.

A screen clear sequence is shown in Listing 2. This sequence clears the entire 32,768 bytes of display memory to zeroes in under 16 ms, or less than the time required for one video field. This routine can be modified to clear 128K bytes of display memory by changing the indicated line to: CPX #$04.

A final sequence of commands and parameters, shown in Listing 3, draws a rectangle in the center of the screen with the dimensions of 100 vertical pixels by 100 horizontal pixels.

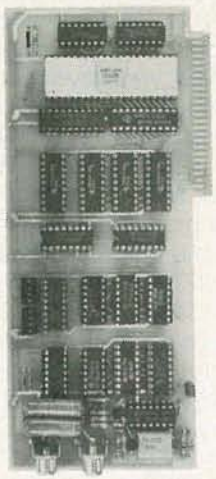
That's all the space for now. We'll conclude this article next month.

BUILD THE GRAFEX-32

PART 3

High-Resolution Graphics for your Apple II.

RAY DAHLBY



This article was begun two issues ago, continued in the last issue, and is concluded here. We'd recommend that you go back to the February, 1986 issue and re-read the entire story from beginning to end, and if you elect to build this unit, do so, knowing that you now have all of the required information.

A word about monitors

Before I set out to design this project, I went looking for a video monitor with two essential characteristics; a high video bandwidth and a long persistence phosphor screen. The Sanyo 9-inch B&W monitor I had been using with my Apple did a satisfactory job displaying the Apple's 40 column text and 280 by 192 graphics screens but blurred noticeably when displaying 80 column text, an indication of its rather poor video bandwidth of 5Mhz or so. In order to display 640 dots in a 40 μs line, the required video bandwidth should be about equal to the dot clock frequency of 16 Mhz. Another characteristic of the Sanyo, which is common to all video monitors using a standard short persistence phosphor screen, is a flicker effect when displaying interlaced computer graphics images. This flicker is the result of the non-spatially limited bandwidth nature of computer generated images as compared to images from a video camera. Since interlace is necessary to get a vertical resolution above about 200 lines and still stay within NTSC television timing, (and thus retain monitor-compatibility with the Apple II), I had to find a screen with a longer persistence. Fortuitously, Apple Computer Inc. came to my rescue when they introduced the Monitor III. This screen has the necessary combination of high bandwidth, long persistence, and modest price. I wondered if they anticipated my needs since the Apple II does not normally require these.

A peculiarity of the 7220 when generating interlaced video is a timing skew between sync pulses responsible for even and odd video fields. The effect on some monitors, is line-pairing which shows up as an uneven spacing between horizontal lines. This effect can be eliminated on the Monitor III as well as other monitors by a slight adjustment of the vertical hold control.

The Grafex sub-interpreter software includes a command to switch the display from 640 by 400 interlaced video to 640 by 200 non-interlaced. This command can be useful for those users who do not have a long persistence screen. The 400 line vertical resolution can still be used by programs to address to complete bit-map and the display scrolled vertically to view it. In effect, the screen shows a window of 640 by 200 on a 640 by 400 "world." If 64K by 4 bit RAM chips are substituted for the 16K by 4 bit 4416's, this "world" will be 640 by 1600. The sub-interpreter has a command called "SCROLL" which accepts a parameter in the range of 0 to 1599 for this purpose.

Conclusion

The 7220 is the first graphics co-processor chip to become widely available in the microcomputer marketplace. Now in volume production, it has dropped in price to about $40.00 in single units from over $100.00 a year ago. The Grafex board serves as an evaluation and development tool for this chip, in addition to its main use as a graphics expander for the Apple II. As more sophisticated graphics co-processors and larger memories become available, the open-architecture of the Apple II will likely play host to these products as well, allowing Apple users the opportunity to stay abreast of the latest technology while expanding and enhancing the utility of their machines.

It was not the intention of this article to provide a complete tutorial on programming the 7220. Such a tutorial alone would take several times the size of this article. A software package has been written to facilitate the use of the Grafex circuit board from BASIC programs. This package is in the form of a sub-interpreter to Applesoft and adds about thirty commands relating to drawing points, lines, rectangles, arcs and circles as well as text, from within Applesoft BASIC programs. Please check the end of this article for prices and ordering information. Those readers interested in writing their own drivers for the 7220 are urged to buy, borrow, or steal a copy of the "7220 DESIGN MANUAL." This 138 page book is an excellent source of information on both the hardware and software aspects of this chip. It is available for $10.00 from NEC Electronics U.S.A., One Natick Executive Park, Natick, Massachusetts 01760.

PARTS LIST
SEMICONDUCTORS
D1, D2—1N914 or 1N4148 diode
IC1—PD7220D graphic display controller
IC2, IC3—74LS00—Quad NAND gate
IC4—74LS02—Quad NOR gate
IC5—74LS86—Quad exclusive OR gate
IC6—74SO4—Hex inverting amplifier/buffer
IC7-IC9—74LS74—duel-D flip-flop
IC10, IC19—74LS174—Hex-D flip-flop IC11—74LS245—Octal transceiver
IC12—74LS374—Octal D-type flip-flop
IC13-IC16—TMS4416-20NL Dynamic RAM
IC17, IC18—74LS166 Paralle-to-serial shift register
IC20—74LS163PNP Binary Counter
Q1—2N3906 PNP Silicon Transistor
Q2—2N3904 NPN Silicon Transistor
XTAL—16MHz quartz crystal

CAPACITORS
C1-C5—0,1 + microF, 50-volt ceramic disc

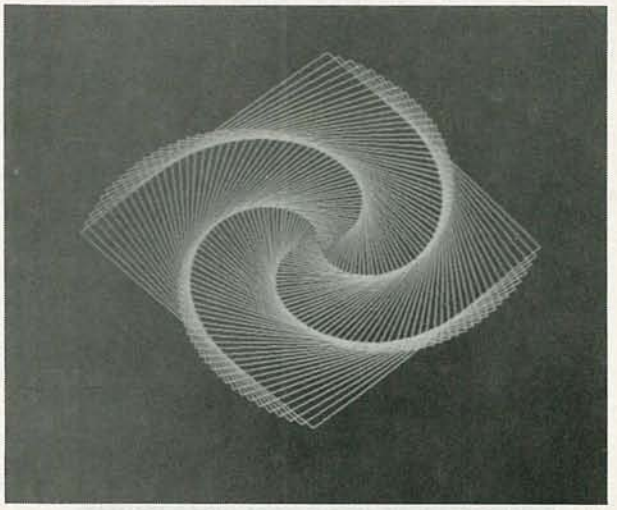
RESISTORS
(All rersistors ¼-watt 5%)
R1, R2, R9—100-ohm
R3, R10—1000-ohm
R4—1500-ohm
R5—3300-ohm
R6—2200-ohm
R7—5100-ohm
R8—200-ohm
R11—75-ohm

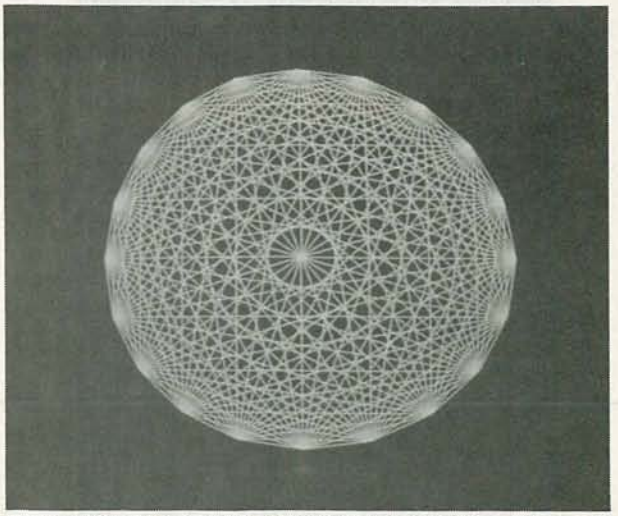
MISCELLANEOUS
J1, J2—PC-mount phono jac
RY1, RY2—SPST N.O. 5-volt reed relay

KITS AND SOFTWARE AVAILABILITY
The following kits and software packages are available for the Grafex-32: a) Printed circuit board (professionally made with plated-through holes, solder-mask and gold-plated edge connector) $45.00. b) Kit of all parts for above $125.00. c) Assembled and tested board $195.00. d) Sub-interpreter software and demos on 5¼" diskette, Apple DOS 3.3 $45.00. e) Color software package on 5¼" diskette. Apple DOS 3.3 (requires three Grafex-32 boards and RGB monitor) $95.00. Please add $4.00 per order for Air-mail postage and handling. Payment accepted by check or money order in U.S. funds to: Ray Dahlby Electronics, Dept. #255, Box C 34069, Seattle, Washington 98124-1069, or: Ray Dahlby Electronics, Dept. 255-810 West Broadway, Vancouver, B.C. V5Z 4C9, For technical inquiries, phone (604) 732-1080 (no phone orders please).

The screen photos

The screen photos reproduced here were taken from an Apple Monitor and show text and graphics at 640 by 400 resolution. The displays were generated with a Grafex board installed in an Apple IIe running under Applesoft Basic with the Grafex sub-interpreter software. The text displayed in these examples comes from a software character set loaded into system RAM by the sub-interpreter. All Grafex displays are bit-mapped; there is no hardware character generator on the board. The character font stored in system RAM can be read into the 7220 GDC which then writes it into the screen bit-map in any of eight directions and 16 sizes. In addition, derived character sets can be displayed by





THE AVAILABLE DETAIL at 640 by 400 resolution . These are common examples of computer graphics. Start with a figure, and repeat at different rotations and sizes.

shifting the position of characters and writing them on top of themselves using the set, clear, and complement modes of the GDC.

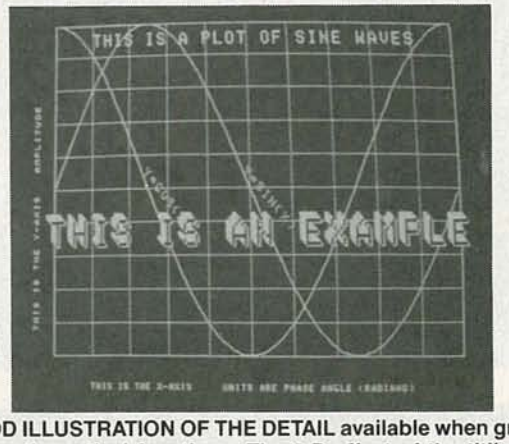
Photograph #2 shows a barberpole effect of 80 columns by 50 rows of text. The character set used in this and the other examples, is made from a 5 by 7 dot matrix in an 8 by 8 box.This character set includes the complete 128 character ASCII set. The Monitor /// screen displays highly readable text in this 80 by 50 format and, in fact, I find it to be preferable to the Apple 80 column display.

The 8 drawing directions are shown in photograph #3.The title for this display was produced by writing the string "DIFFERENT DIRECTIONS" at a magnification factor of 2, and then writing it again but with the x and y starting position shifted by 1 pixel and the mode set to complement.

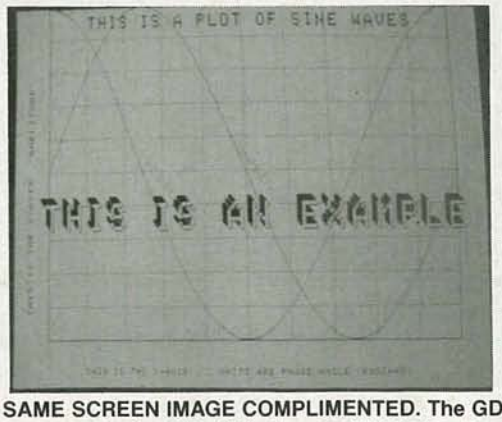
Photographs 4, 5 and 6 show off the detail available at 640 by 400 resolution and are quite common examples used in computer graphics. Photos 4 and 5, for example, are simply the result of repeatedly drawing a triangle and a rectangle, at different rotations and sizes.

Photograph #7 is a good illustration of the detail available when graphing mathematical functions. The 3-D effect of the titling is again produced by writing a shifted text string on top of itself. Photograph #8 is the same screen image complemented.The GDC can perform an exclusive-or of this type on 32K bytes of memory in under 16ms, obviating the need for hardware reverse video capability.

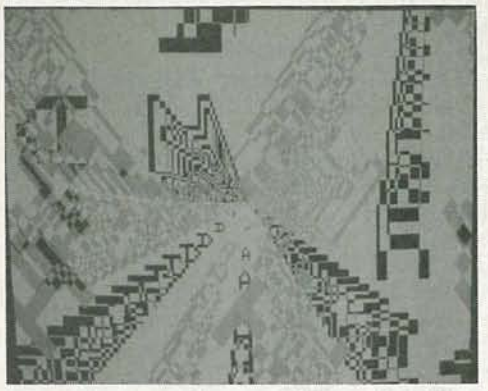
The last two pictures, #8 and #9, show a fancifull "video art" example I wrote by accident. This effect is produced by writing the letter "A" at different magnifications, rotations, and modes (set, clear, and complement). The program, written in Applesoft Basic, is just a few lines in length yet produces videotape-like animation.



GOOD ILLUSTRATION OF THE DETAIL available when graphing mathematical functions. The 3-D effect of the titling results from a text string written on top of itself.



THE SAME SCREEN IMAGE COMPLIMENTED. The GDC can perform an exclusive-OR of this type on 32K-bytes of memory in under 16 ms.



SOME VIDEO ART produced accidentaly by writing the letter "A" at different magnifications, rotations and modes. The program, written in Applesoft BASIC, is just a few lines in length yet produces a videotape-like animation.

References

1—"High Resolution Sprite-Oriented Color Graphics:" Steve Ciarcia: Byte Magazine, August 1982.
2—"Polish Your Apple With A Luminance Board:" Ray Dahlby: Computers and Electronics magazine, November 1982.
Further Reading
1—"Display Controller Simplifies Design of Sophisticated Graphics Terminals:" Jeffery L. Wise and Henryk Szejnwald: Electronics magazine April 7, 1981.
2—"Super Graphics Hardware from NEC :"Steve Levine: Byte magazine, April, 1983.
3—"MOS Memory Data Book": Texas Instruments:1984.
4—"Output-enable feature a must on fast common I/O memories": Al Reddy and Sherri Craig: Computer Systems Equipment Design magazine, January, 1985.