Build an Oscilloscope

Ever wonder how to make a computer draw pictures for output? One way is to use an oscilloscope – which many readers have on general principles for debugging the logic circuitry. Jim Hogenson provides a practical circuit for accomplishing that end in his "Oscilloscope Graphics Interface" design. This graphics device was conceived by Jim as a neat idea to add to the 8008-oriented computer system he was building for a high school science fair. He first mentioned it to me in a letter late last year. I suggested to him (or was it the other way around?) that it might be appropriate to turn it into an article for the *ECS Magazine* I was publishing at the time. After a fair amount of time spent researching the various options – plus one lengthy phone conversation with me – Jim settled on the design shown in this article, which is reprinted here from its original publication in the last issue of *ECS Magazine*. The interface is very simple, and can be adapted to virtually any computer with a minimum of 8 parallel TTL output lines and a clock pulse line which is active when output data is stable. Arrangements have been made for a PC version of this design (see the parts list, Fig. 6) so you won't have to wire wrap the thing like Jim did in his first version.

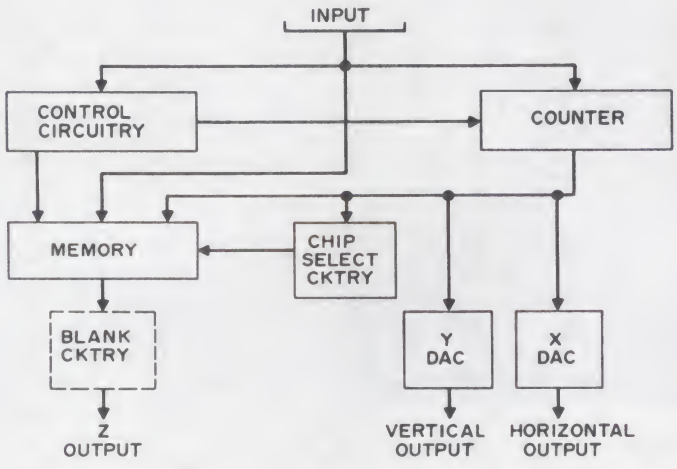
. . . CARL

by
James Hogenson
Box 295
Halstad MN 56548

Many members of the large family of alphanumeric computer output devices may be readily used in the home computer system. But there are as yet few devices of a graphic orientation which are economically acceptable in the home computer system. The oscilloscope graphic interface project presented here provides one unique, inexpensive and uncomplicated solution to the graphic output problem in small scale systems. It turns an essential test instrument – the oscilloscope – into a versatile output device.

The oscilloscope graphic interface is programmed and operated through a parallel 8-bit TTL compatible input. An image is represented by a pattern of dots which is organized according to the computer's instructions. During the scan cycle, the digital dot pattern is converted to analog waveforms which reproduce the image on an oscilloscope screen. The graphic interface stores the dot pattern within its own internal refresh memory. Therefore, once the pattern has been generated and loaded into the graphic interface memory, the computer is left free to execute other programs.

Fig. 1. Oscilloscope graphics display block diagram.



Principle of Operation

The raster begins its scan in the upper left-hand corner, scanning left to right and down. The full raster contains 4096 dots, 64 rows of 64 dots each. The horizontal scan is produced by a

Graphics Interface

Fig. 2. Oscilloscope graphics interface instruction codes.

Op Code Binary	Octal	Mnemonic	Explanation
00dddddd	Odd	STX	Set X
01dddddd	1dd	STY	Set Y
10xxx000	2x0	DCY	Control – Decrement Y
10xxx001	2x1	TSF	Control – Turn off scan
10xxx010	2x2	ZON	Control – Set Z on
10xxx011	2x3	ZOF	Control – Set Z off
10xxx100	2x4	ZNI	Control – Set Z on with increment
10xxx101	2x5	ZFI	Control – Set Z off with increment
10xxx110	2x6	TSN	Control – Turn on scan
10xxx111	2x7	DCX	Control – Decrement X
11xxxxxx	3xx	CNO	No Op

d = data x = null

stepping analog ramp wave. Each of the 64 steps in the ramp produces one dot. The vertical scan is similar. It is a stepping ramp wave consisting of 64 steps. However, there is only one step in the vertical wave for each complete horizontal ramp wave. The result is 64 vertical steps with 64 horizontal steps per vertical step, or 64 rows of 64 dots each.

The timing of horizontal and vertical sweep waveforms originates in a 12-bit binary counter, the operational center of the entire circuit. The six least significant bits of the counter are connected to a digital-to-analog converter (DAC) which converts the digital binary input to a voltage level output. The output of the least significant DAC is the horizontal ramp wave. The six most significant bits are connected to a second DAC. This DAC produces the vertical ramp wave. Incrementing the 12-bit counter at a frequency of around 100 kHz results in a raster on the screen of the oscilloscope.

The contrast in the pattern of dots needed to represent a picture is dependent upon the intensity of each dot. From this point, it is assumed that a dot can be either on or off. An "on" dot will show up on the screen as a bright dot of light. An "off" dot will be a dim dot of light.

When a particular dot is addressed by the counters, it may be set to either the "on" or the "off" state. The on-off control is represented by a single bit. It is this bit which is stored in the internal memory of the oscilloscope graphic interface. There is one bit in the memory for each of the 4096 dots in the raster. When displaying the image, the 12-bit counter which produces the raster addresses the appropriate dot status bit in the memory as that dot is produced on the screen. The on-off dot status bit taken from the memory is converted to a Z-axis signal which controls the intensity of the dot on the screen.

The major portion of the circuitry is taken up in the 12-bit counter, the DACs, and the memory. Fig. 1 shows a block diagram of the oscilloscope graphic interface. The remaining circuitry is the control circuitry which decodes the 8-bit input word and allows for completely programmed operation.

Programming

The programming instruction format is shown in Fig. 2. Bits 7 and 6 of the input word are the high-order instruction code. It is assumed that the addressing of dots is done on the basis of X and Y coordinates. The X coordinate is the 6 bits in the least significant or horizontal section of the 12-bit counter. The Y coordinate is the 6 bits in the most significant or vertical section of the counter. In programming from an 8-bit microcomputer source, all 12 bits of the counter cannot be set at once. The counter is set one half or 6 bits at a time. It is for this reason X and Y coordinates are assumed in programming.

When the instruction code (bits 7 and 6) is set at 00, the data on bits 0 through 5 of the input word is loaded into the least significant counter section as the X coordinate. When the instruction code is set at 01, the data on bits 0 through 5 is loaded into the most significant counter section as the Y coordinate. In effect, the Y coordinate will select a row of dots, while the X coordinate will select one dot in the selected row. The coordinates loaded into the counter will address

the memory and select the desired dot status bit for programming.

After loading the coordinates of the dot selected for programming, the status of the dot (on or off) is set using the ZON, ZOF, ZFI or ZNI control codes. Setting the instruction code at 10 directs the control circuitry to decode the three least significant bits of the input word for further instruction. The three least significant bits are called the "control code."

Since the 12-bit counter must store selected coordinates during programming, the raster scan must be disabled before programming. Control code "1" will stop the scan. Control code "6" will restart the scan. When the scan is on, the 12-bit counter will be incremented at a high frequency and the programmed image is displayed on the scope screen.

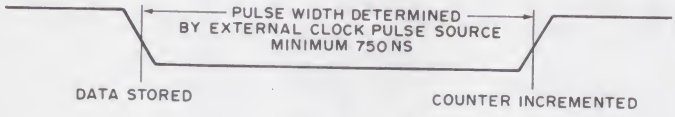
Control code "2", "set Z on", will program a bright dot to appear at the dot location presently stored in the counter. Control code "3", "set Z off", will program a dim dot or blank to appear at the dot location presently stored in the 12-bit counter.

Control codes "4" and "5" set Z in the same manner as control codes two and three. However, after setting Z, these instructions will increment the counter by one thus advancing to the next dot location in the raster scan pattern. This will allow programming of the entire raster using only a repeated "set Z" instruction.

Control code "0" will not set Z, but will decrement the most significant or vertical section of the counter only. In effect, control code "0" will decrement the stored Y coordinate. Control code "7" will not set Z, but will decrement the entire 12-bit counter by one. This, in effect, will decrement the stored X coordinate. Since the X and Y counter sections are cascaded, Y will automatically be incremented or decremented once for every 64 executions of an increment or decrement X control code.

The increment and decrement control codes are very useful in constructing lines in an image since lines require repeated "set Z" instructions, often on the same axis. An effective method of clearing an image from the screen is repeating a "set Z with increment" control code in a programmed loop. This method allows the option of using either a light or dark image background.

Fig. 3. Timing pulse input to the interface. The 8 data lines must be stable during this pulse.



Circuit Operation

Once the data word on the microcomputer parallel output interface is stable, one clock pulse is used to execute the instruction. This clock pulse is taken from the microcomputer output interface. The instruction code is decoded by the 7410 triple three-input NAND gate and two inverters. The clock pulse is enabled by the NAND gate to the appropriate counter section, or to the strobe input of the control code decoder.

The 12-bit counter consists of two 6-bit counting sections. Each section consists of two cascaded TTL 74193 presettable binary counters. Bits 0 through 5 of the data input are common to both sections of the counter. The set X instruction will pulse the load input of the least significant or horizontal section, while the set Y instruction will pulse the load input of the most significant or vertical section of the counter. A pulse on the load input will cause the data on bits 0 through 5 to be loaded into the proper counter section.

Four TTL counters must be used to provide independent loading capabilities for each 6-bit section. The counters within each section are cascaded in the normal fashion. The two sections are cascaded by connecting the upper data B output of the X counter section (IC 8, pin 2) through inverter "a" of IC 2 to the count up input (IC 9, pin 5) of the Y counter section. The inverter is needed to provide proper synchronization in high frequency counting.

The control code is decoded by a 74155 decoder connected for 3 to 8 line decoding. Bits 0 through 2 are decoded by the 74155. The control code is enabled by the pulse coming from the 7410 instruction decoder only when the instruction code is set at 10 on bits 7 and 6.

Decoder lines 1 and 6 are connected to an R/S flip flop

Fig. 4. PC artwork of the graphic interface, by Andrew Hay.
(a) Component side.

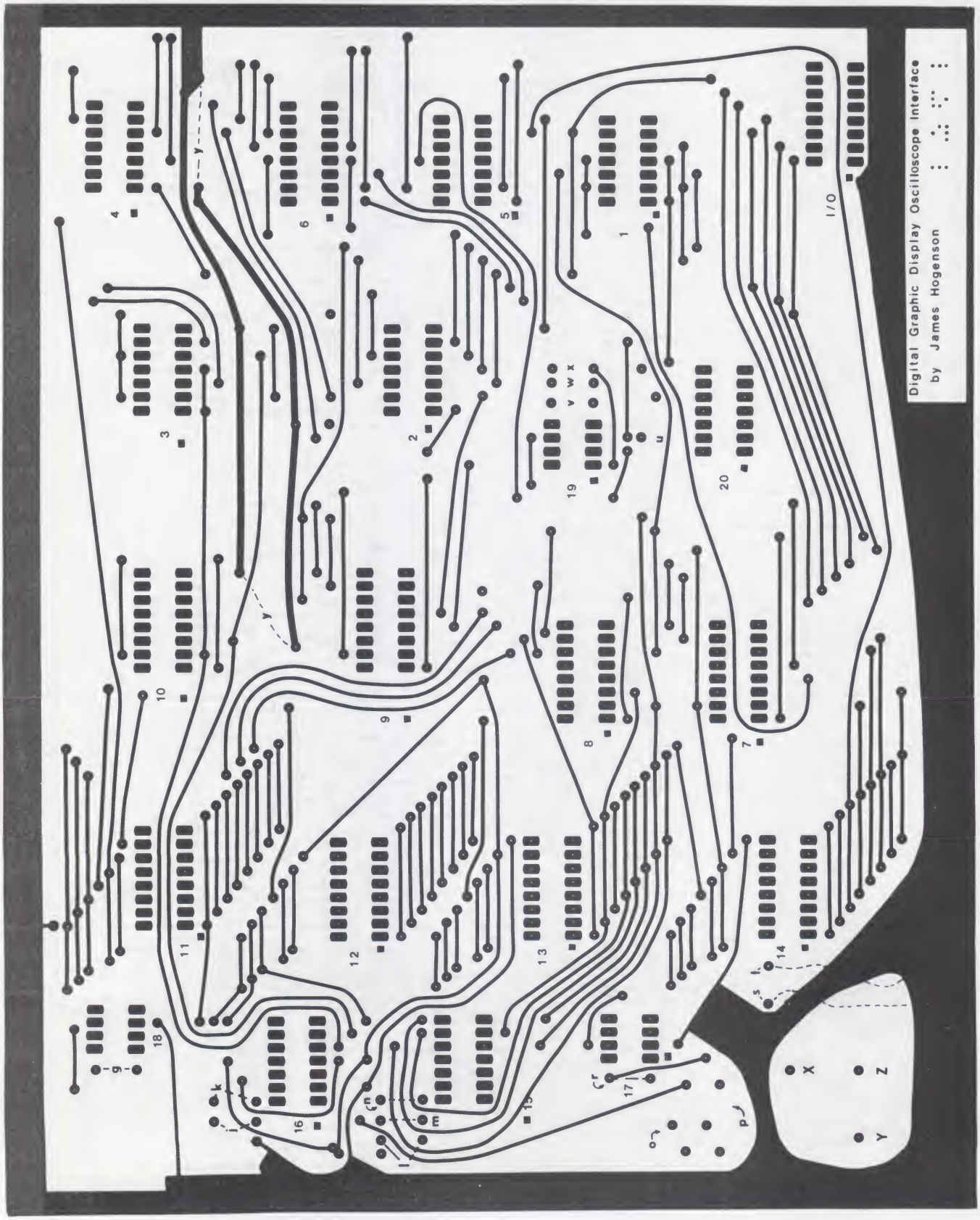
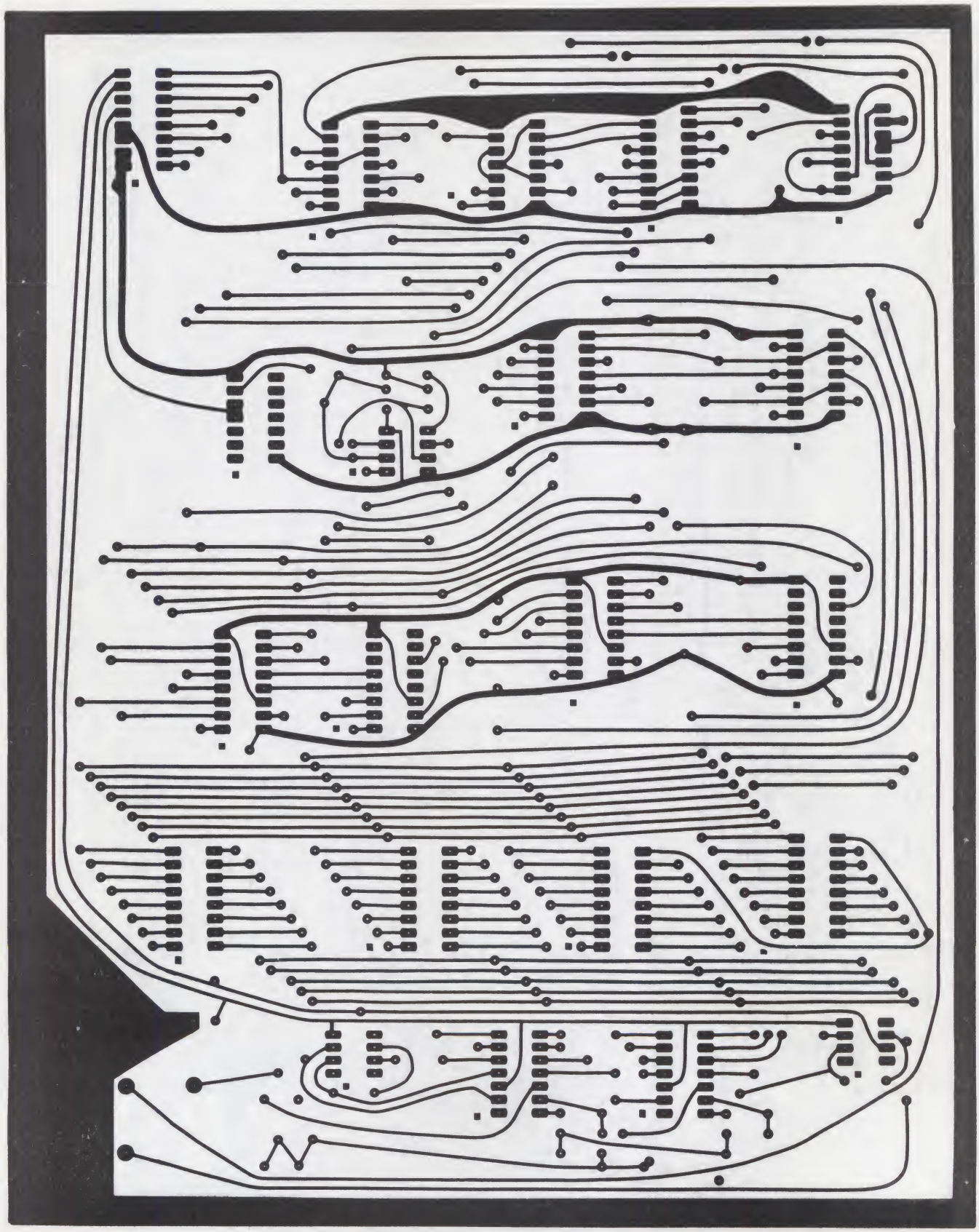


Fig. 4. PC artwork of the graphic interface, by Andrew Hay.
(b) Solder side.



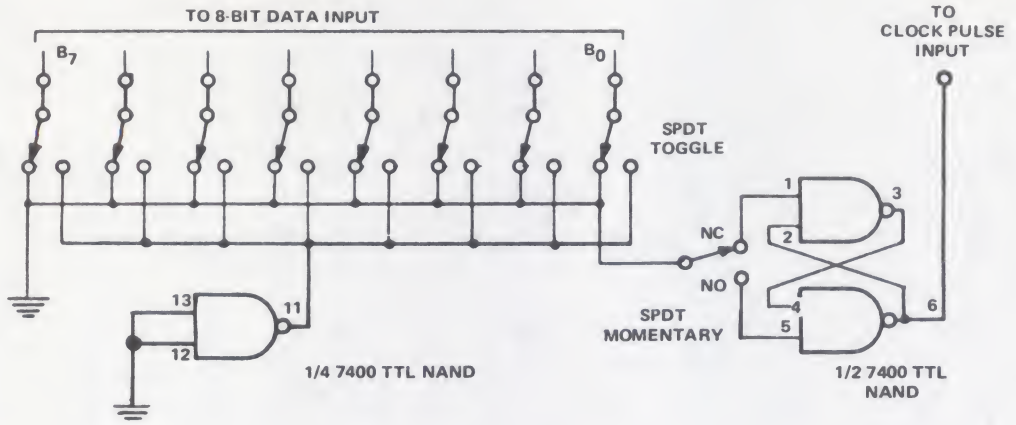


Fig. 5. A test circuit for manual operation. The set-reset flip flop of the 7400 circuit generates a debounced clock pulse which will perform the operation set into the toggle switches. If you haven't got a computer up and running yet, the manual interface can be used in order to test out the display.

which provides the scan on/off control. The flip flop enables the system clock to provide the high frequency square wave which increments the 12-bit counter.

Control codes 2 through 5 define the "set Z" instructions which perform a data write operation. Decoder lines 2, 3, 4 and 5 are connected to a group of AND gates (IC 5a, b, c) functioning as a negative logic OR gate. The output of this gate is the Read/Write control line for the memory. When this line is in the low state, the data present on the data input line of the memory will be written into the memory location presently stored in the 12-bit counter.

The data input of the memory is connected directly to bit 0 of the 8-bit input word. This bit is stored in the memory only when a set Z command is executed. The Z-axis circuit configuration will require a high state pulse for a blank or dim dot. As shown in the binary instruction format, Fig. 2, bit zero will be binary zero for "set Z on" instructions and binary one for "set Z off" instructions. The backward appearance of this binary format will be overlooked when programming in octal notation.

The high frequency system clock controlled by the R/S flip flop and decoder lines 4 and 5 are negative logic ORed. The resulting pulse increments the counter according to control commands.

The same clock pulse taken from the computer output interface is used to write data into the memory and increment the counter in control commands 4 and 5. The data is written into the memory on the leading edge of the pulse. The counter is incremented on the trailing edge. Fig. 3 shows the waveform timing.

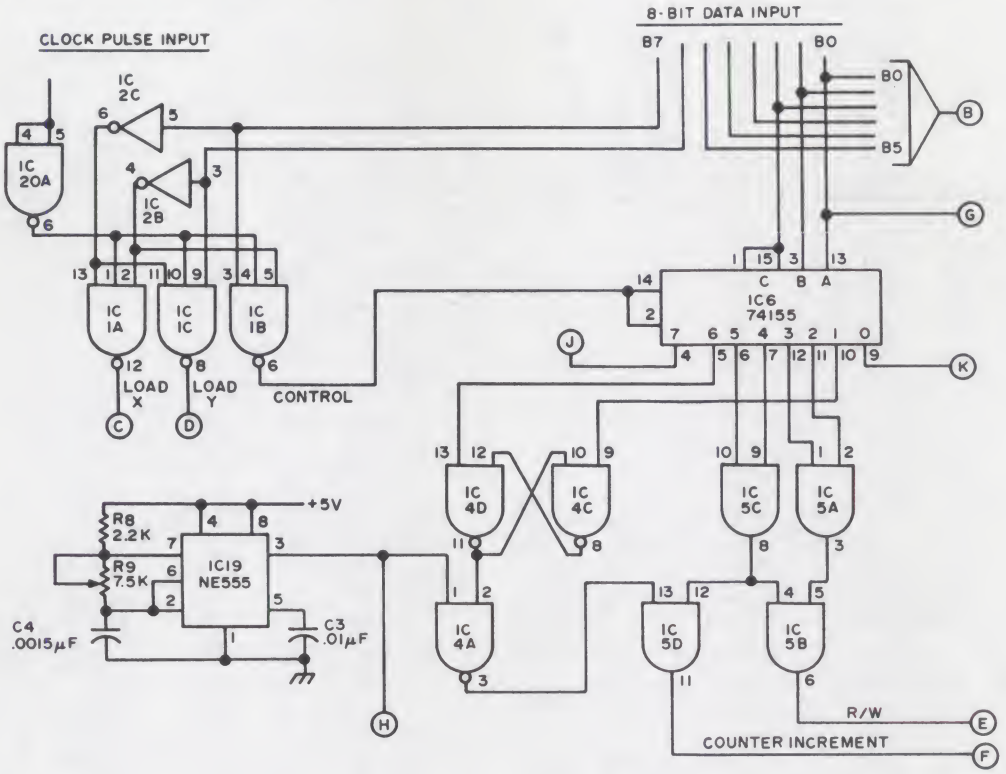
Output bits 0 through 9 of the 12-bit counter are connected to the address inputs of the memory. The memory uses four MM2102 type 1k x 1 bit MOS RAMs (Random Access Memories). Bits 10 and 11 of the counter output are connected to the chip select circuitry which

Fig. 6. Parts list.

C1, C2	20 pF	disc capacitor
C3, C5, C6-C11	.01 mF	disc capacitor
C4	.0015 mF	disc capacitor
C12	25 mF	electrolytic capacitor
IC 1	7410	TTL triple 3-input NAND gate
IC 2	7404	TTL hex inverter
IC 3, IC 4, IC 20	7400	TTL quad 2-input NAND gate
IC 5	7408	TTL quad 2-input AND gate
IC 6	74155	TTL dual 2-to-4-line decoder
IC 7 – IC 10	74193	TTL presettable 4-bit binary counter
IC 11 – IC 14	2102	NMOS 1024-bit static RAM
IC 15, IC 16	MC1406	Motorola 6-bit DAC
IC 17, IC 18	741	Op amp
IC 19	NE555	Oscillator (timer IC)
R1, R2	3.3k Ohm	resistor
R3, R4	5.6k Ohm	resistor
R5, R6	10k Ohm	miniature potentiometer
R7	1k Ohm	resistor (all resistors ¼ Watt, 10%)
R8	2.2k Ohm	resistor
R9	7.5k Ohm	miniature potentiometer

A printed circuit board using the masks of Fig. 4 is available for $29.95. Write to M. F. Bancroft, CELDAT Design Associates, Box 752, Amherst NH 03031.

Fig. 7. Oscilloscope graphics interface circuit diagram. (a)



enables one memory chip at a time for addressing and data input/output operations. The chip select circuitry uses 2 inverters and a TTL 7400 Quad two-input NAND gate.

The data outputs of the RAMs are OR-tied and connected to an AND gate. The data output is synchronized with the high frequency clock for better blanking performance. The output of this gate is connected to the Z-axis blanking circuitry. The blanking circuitry converts the TTL level signal to a scope compatible signal which may be varied over a wide range of output voltages to best match the scope being used.

Bits 0 through 5 of the 12-bit counter are connected to the X coordinate DAC. Bits 6 through 11 are connected to the Y coordinate DAC. The DACs are Motorola MC1406 ICs. The DACs operate on voltages of +5 and -5 to -15. A current output is produced by the DACs. The current output is converted to a voltage output and amplified by the 741 op amps. The output from the X coordinate amp is connected to the horizontal input of the scope. (The scope should be set for external horizontal sweep.) The output from the Y amp is connected to the vertical scope input.

Although the scope used does not need dc-coupled inputs, triggered sweep, or high frequency response for this project, a Z axis or intensity input is required. The Z axis output provided on the interface PC pattern is TTL compatible only. Most scopes will need some type of blanking circuitry to amplify the TTL level pulses. The design of the blanking circuitry will be of the builder's choice, allowing the builder to best suit his scope. A suggested method which is simple and effective is the use of the circuit shown in Fig. 13.

Construction

This project may be wire wrapped, the PC artwork in Fig. 4 may be used to fabricate a double-sided printed circuit board, or the printed circuit board product mentioned in the parts list may be employed. The PC pattern is designed for easy soldering. The components need be soldered on the bottom side only.

Remember that the memory ICs are MOS devices and should be handled as such. Static electricity will easily puncture the thin MOS transistor junctions.

Bypass capacitors should be connected between supply voltages and ground. A minimum of a 10 mF electrolytic or tantalum capacitor should be used for all supply voltages. For the +5 logic supply, one .01 mF disc capacitor should be used for each 2 to 5 integrated circuits. The large

electrolytics will filter out low frequency noise and voltage transients while the small disc capacitors will filter out high frequency noise which could falsely trigger flip flop and counter circuits.

Set-up, Testing and Operation

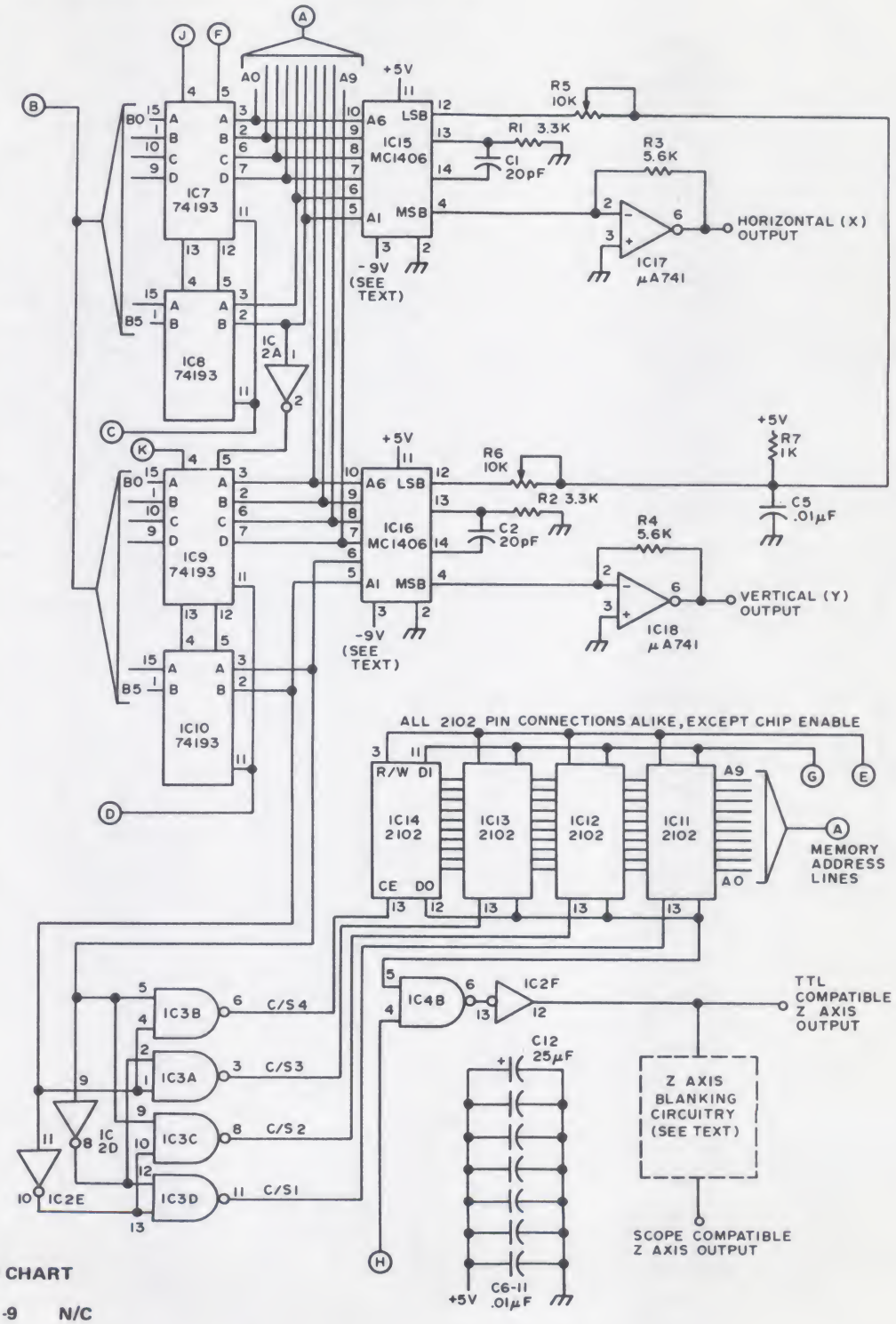
The system requires a +5 volt, 400 mA power supply and a dual polarity supply of from ±9 to ±15 volts at 10 mA. The wide range of analog supply voltages allows use of existing power supplies for the graphic interface.

The clock pulse derived from the computer parallel I/O interface should be active in the low state. If a device operating with an active high pulse is used, one of the free gates of IC 20 may be used to invert the clock pulse or IC 20 may be omitted.

When ready for testing, be certain of voltage supply polarities, then apply power. If the scan does not come on at random, execute a "turn on scan" command. Using the 10k Ohm pots, R5 and R6, adjust the DAC voltage references to eliminate any distorted concentration of dots in the raster.

The system clock consists of a 555 timer IC connected as an astable multivibrator.

Fig. 7. Oscilloscope graphics interface circuit diagram.(b)



IC POWER AND PIN CONNECTION CHART

IC	+5	GND	+9	-9	N/C
1,2,3,4,5	14	7			
6	16	8			9,4
7, 9	16	8, 14			
8, 10	16	8, 14			6,7,9,10,12,13
11,12,13,14	10	9			
15,16	11	2		3	1
17, 18			7	4	1,5,8
19	4,8	1			
20	14	7			1,2,3,8,9,10,11,12,13

2102 MEMORY ADDRESS PIN CONNECTIONS

A-0 – pin 8 : A-1 – pin 4 : A-2 – pin 5 : A-3 – pin 6
A-4 – pin 7 : A-5 – pin 2 : A-6 – pin 1 : A-7 – pin 16
A-8 – pin 15 : A-9 – pin 14

Fig. 8. CLEAR Program flow chart.

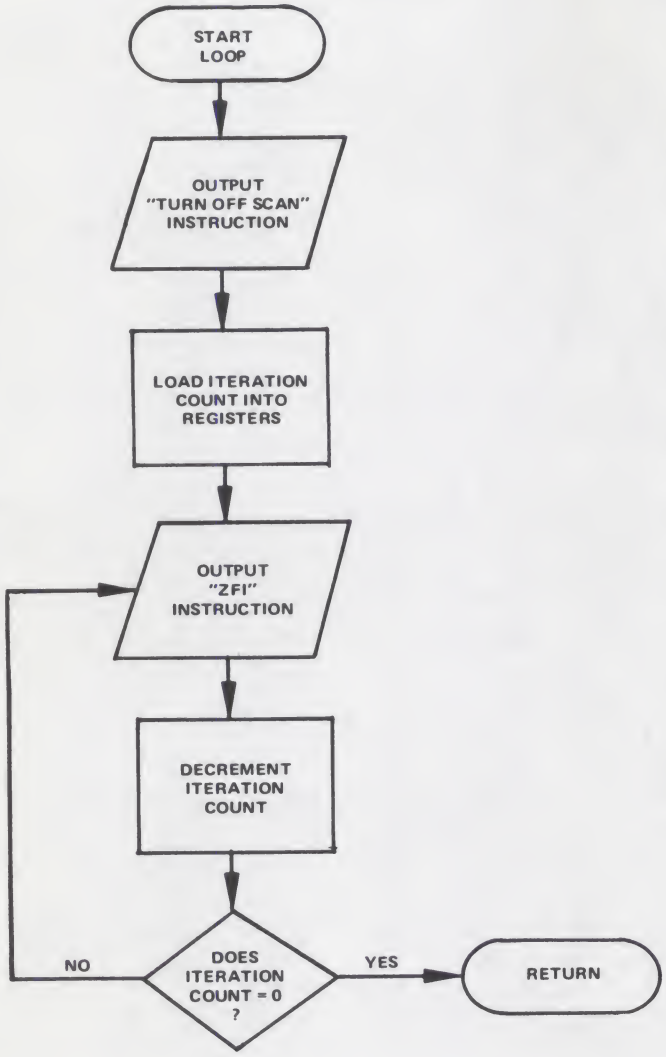


Fig. 9. Listing of 8008 code for the CLEAR program.

```
START
                00/344 = 006    LAI
                00/345 = 201    (TSF)
                00/346 = 121    OUT 10
                00/347 = 006    LAI
                00/350 = 205
                00/351 = 016    LBI
                00/352 = 377
                00/353 = 026    LCI
                00/354 = 021
                00/355 = 121    OUT 10
                00/356 = 011    DCB
                00/357 = 150    JTZ
                00/360 = 365
                00/361 = 000
                00/362 = 104    JMP
                00/363 = 355
                00/364 = 000
                00/365 = 021    DCC
                00/366 = 110    JFZ
                00/367 = 355
                00/370 = 000
                00/371 = 377    HLT
```

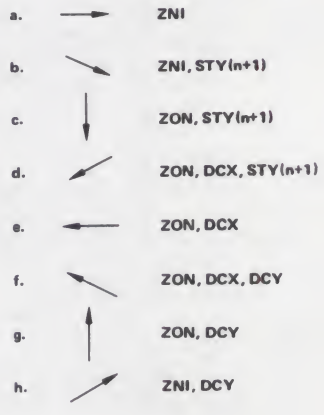
Adjusting the frequency may be necessary to obtain a stable raster. The frequency is adjusted using R9, the 7.5k pot. The frequency of the system clock should be approximately 100 kHz, but is not critical. The only requirement is appearance of the raster.

If the raster is evenly distributed over the screen, but is severely chopped up, check the digital inputs to the DACs. Use the scope to check the vertical and horizontal ramp waves individually. If the wave is not an even ramp, two or more of the DAC inputs may be reversed. Note that DAC input A1 is the most significant bit while input A6 is the least significant bit. Reversed inputs may also cause incomplete raster formations. Slight gaps or overlapping between some dots is caused by non-linearities in the manufacturing of the DACs.

If no raster at all appears, first check for a square wave output at pin 3 of the 555 timer IC. Then check for square wave outputs at each TTL 74193 counter. These square waves will be binary submultiples of the oscillator frequency. If the counter is operating, check all connections to the DACs and op amps.

Applying power will produce a random pattern of on and off dots. Adjust the amplitude of the Z axis signal for best contrast. Since most scopes will have an ac-coupled (or capacitor coupled) Z axis input, both amplitude and frequency of the signal will affect

Fig. 10. To construct a line segment in the direction shown by the arrow, alternately execute the commands shown.



performance. Charging the capacitor within the scope with too much voltage at a given frequency will cause the blank pulse to carry over into the next dot. This could cause more dots than desired to be blanked out or dimmed.

After a satisfactory raster is obtained, each instruction should be executed to verify its operation. First, clear the screen. The flowchart for a simple CLEAR program is shown in Fig. 8. The method outlined is to simply send out a "set Z off with increment" instruction 4096 times.

Fig. 9 shows the program listing for an 8008 system. This example used the B and C registers to keep track of the iteration count. The register contents are decremented once for each output ZFI instruction. The RETURN instruction may be substituted with a HALT if the CLEAR program is not to be used as a called subroutine. The CLEAR subroutine as listed in Fig. 9 begins by turning off the scan (which must be done before any programming, as stated), but does not turn the scan back on after the interface memory is cleared. The course of operation is left to the programmer once CLEAR has been called.

The chart in Fig. 10 may be used in testing the various control commands. The chart shows the commands to be used to construct a line segment in the direction shown by the arrow. Lines moving in a downward direction require that Y be reset with (n+1) for each dot programmed, "n" being the

Fig. 11. CHECKERBOARD Test Pattern Program flow chart.

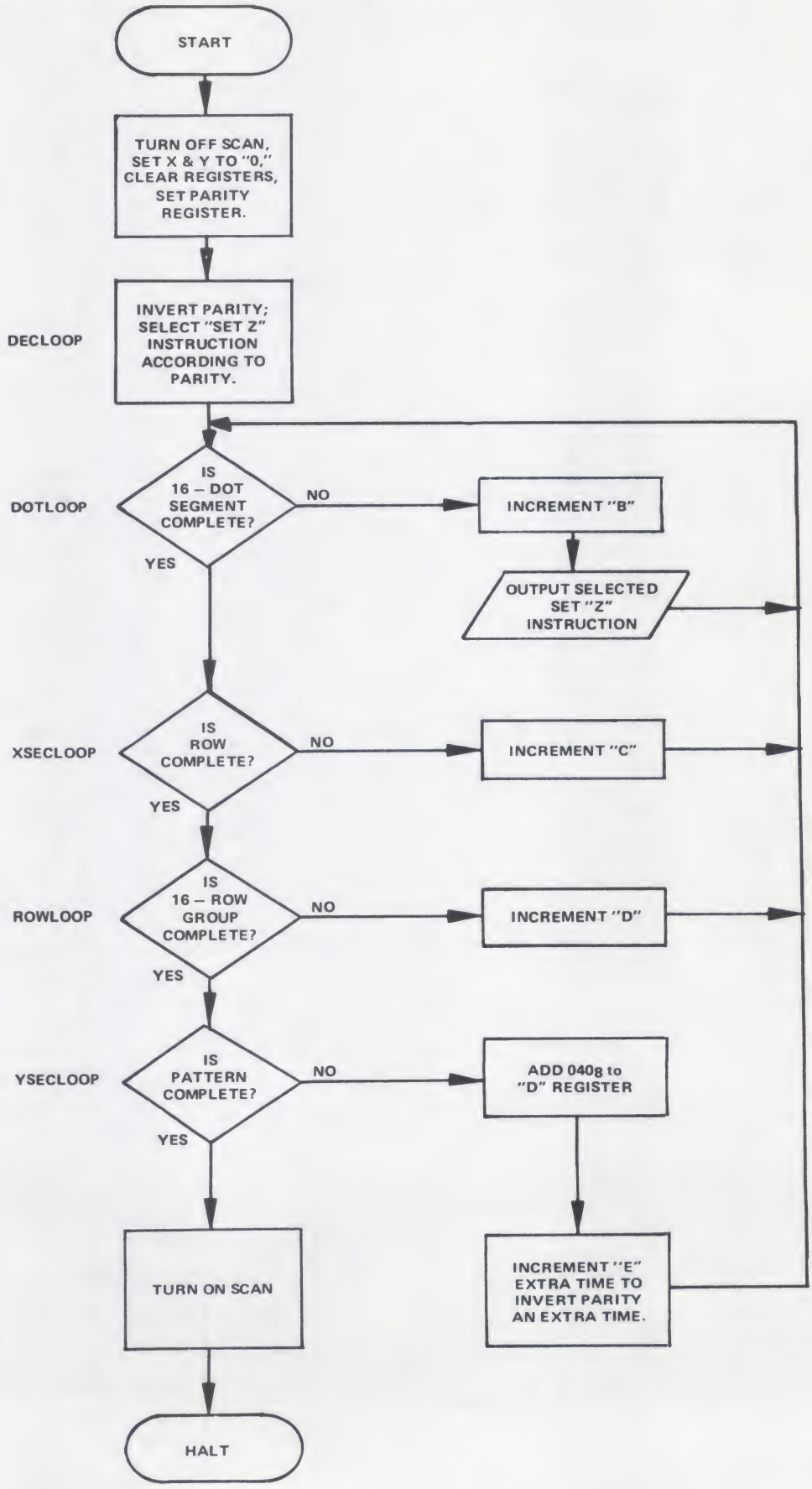


Fig. 12. Listing of 8008 code for the CHECKERBOARD program.

```
START       00/200 = 006    LAI
            00/201 = 201    (TSF)
            00/202 = 121    OUT 10
            00/203 = 006    LAI
            00/204 = 000    (STX)
            00/205 = 121    OUT 10
            00/206 = 006    LAI
            00/207 = 100    (STY)
            00/210 = 121    OUT 10
CLEAR       00/211 = 016    LBI
REGISTERS   00/212 = 000
            00/213 = 321    LCB
            00/214 = 331    LDB
            00/215 = 351    LHB
            00/216 = 361    LLB
            00/217 = 046    LEI
PARITY REG  00/220 = 000
DECLOOP     00/221 = 040    INE
            00/222 = 304    LAE
            00/223 = 044    NDI
            00/224 = 001
            00/225 = 150    JTZ
            00/226 = 246
            00/227 = 000
            00/230 = 066    LLI
            00/231 = 332
DOTLOOP     00/232 = 301    LAB
            00/233 = 024    SUI
            00/234 = 020
            00/235 = 150    JTZ
            00/236 = 253
            00/237 = 000
            00/240 = 010    INB
            00/241 = 307    LAM
            00/242 = 121    OUT 10
            00/243 = 104    JMP
            00/244 = 232
            00/245 = 000
DECLOOPJMP  00/246 = 066    LLI
            00/247 = 333
            00/250 = 104    JMP
            00/251 = 232
            00/252 = 000
XSECLOOP    00/253 = 016    LBI
            00/254 = 000
            00/255 = 302    LAC
            00/256 = 024    SUI
            00/257 = 003
            00/260 = 150    JTZ
            00/261 = 267
            00/262 = 000
            00/263 = 020    INC
            00/264 = 104    JMP
            00/265 = 221
            00/266 = 000
ROWLOOP     00/267 = 026    LCI
            00/270 = 000
            00/271 = 303    LAD
            00/272 = 044    NDI
            00/273 = 037
            00/274 = 024    SUI
            00/275 = 017
            00/276 = 150    JTZ
            00/277 = 305
            00/300 = 000
            00/301 = 030    IND
            00/302 = 104    JMP
            00/303 = 221
            00/304 = 000
YSECLOOP    00/305 = 303    LAD
            00/306 = 044    NDI
            00/307 = 340
            00/310 = 330    LDA
            00/311 = 024    SUI
            00/312 = 140
            00/313 = 150    JTZ
            00/314 = 326
            00/315 = 000
            00/316 = 303    LAD
            00/317 = 004    ADI
            00/320 = 040
            00/321 = 330    LDA
            00/322 = 040    INE
            00/323 = 104    JMP
            00/324 = 221
            00/325 = 000
END         00/326 = 006    LAI
            00/327 = 206    (TSN)
            00/330 = 121    OUT 10
            00/331 = 377    HLT
            00/332 = 204    (ZNI)
            00/333 = 205    (ZFI)
```

present Y coordinate. Use the STX and STY instructions to select a starting point. The dot whose coordinates are X=00, Y=00 will be in the upper left corner, the point where the scan begins its cycle.

The flow chart for a CHECKERBOARD TEST PATTERN program is shown in Fig. 11, with an 8008 listing in Fig. 12. The pattern produced will be 16 alternating light and dark squares. The 64 rows of dots are divided into 4 groups of 16 rows each. Each row is divided into 4 segments. The segments are alternately light and dark. The 4 groups also alternated to reverse the pattern between each group.

The set Z with increment instructions is used. The least significant bit of the E register is used in DECLOOP to alternate between "set Z on" and "set Z off." To obtain the complement of the entire pattern on the screen, place a 001 in location 00/220 instead of 000.

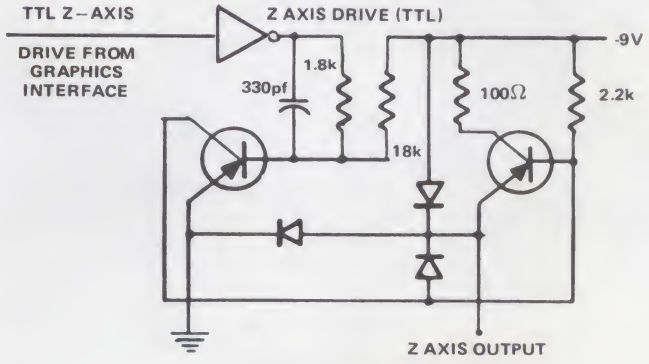


Fig. 13. A Z-axis drive circuit used to control blanking in the author's original version of the design. The transistors are 2N5139s and the diodes are silicon switching diodes such as the 1N914 part or its equivalent.