A Color Television Interface

Easy — Versatile — Inexpensive

By William Rogers

The contents of this application note will cover many topics concerning the Video Display Generator; generalizing on some, baiting with others and specifying one complete project. First, I'll talk about why a versatile system is easy to build inexpensively. Then I'll turn to the performance abilities of the VDG and then mention two systems on either extreme. Fourth I'll enter into a software section including a demonstrating program, an expandable TV output display program (for an existing terminal) and a cursor program, which is the main software in this article, and is also expandable. Fifth comes the hardware section complete with an operational schematic for an Exorcisor compatible board. Other systems may function with the hardware as long as the proper signals are used.

Two new products built by Motorola help comprise a display interface circuit for the 525 line black and white televisions or the NTSC (National Television Standards Committee) standard color television sets. The Video Display Generator (MC6847), the Color TV Modulator (MC1372), some memory chips and approximately twelve passive discrete components coupled with an MC6800 microprocessor, or any other MPU (Microprocessor Unit) convert the display system into an active and intelligent terminal.

The ease of interconnection becomes apparent when constructing a system. Most pins have definite connections such as the data bus, the address bus, the analog outputs, the power pins, and the clock input. When using an MPU, the data and address buses need three-state buffers between the VDG's buses. The control pins may be hardwired or logically connected in some fashion making the degree of construction difficulty user definable. A pin similar to a memory chip's select allows three stating of the VDG's address bus and therefore accessibility to the display RAM by an MPU. The other three pins would probably not be used by hobbyist or consumer products houses unless an external character generator was required for a more sophisticated system. An example of a higher level system, which will not be discussed in this article, is the display of apparently 6K of RAM when only 1K of RAM exists in the system. The number of chips involved is decreased significantly using the VDG, therefore making a system easier to build.

Chip count also makes a system less expensive. One VDG costs about $19.95, one TV modulator costs about $4.42, eight 2102 1K x 1 RAMs cost approximately $8.00, two QUAD three-state bus transceivers cost about $5.40, and three HEX three-state buffers cost $5.88 which add up to $43.65. Add a few more dollars to that cost for discretes plus miscellaneous TTL for decoding and a complete display interface with alphanumeric, dense graphics and eight-color capability is achieved for less than $50 on a single unit basis. Compared to $250 up to a $580 cost for boards and compared to the functionality of each board this is a substantial savings in a system investment.

**Prices given are approximations only.*

Versatility? The VDG has it! Depending on how a person views the concept of modes, the VDG has eleven major modes with a total of 27 distinguishable modes including all the variations. If three state is considered as a viable mode then add one more to the total count.

An explanation of some performance abilities will also back up the variability of the VDG. The circuit operates on +5 volts only, therefore keeping system cost down if no other parts require other power supplies. An on-board character generator has 64 ASCII characters and is user definable with a mask change. An External/Internal Horizontal Synchronization and Row Preset signals are provided for the timing of an external character generator. Eight colors: magenta, blue, orange, green, cyan (a light blue color), yellow, red and buff (an off-white looking color) plus black make up the color selection. The color information feeds into the modulator from two chrominance pins R-Y (ϕA) and B-Y (ϕB). The complete video information (synchronization pulses and data) for a black and white television set comes out on the luminance pin (Y). Eight control pins allow hardware or logic selectable modes.

The first major eleven modes is an Alphanumerics mode which can use the internal or external ROM (character generator) in either green or orange color and can use inverse or noninverse video. Inverse and noninverse simply refer to the characters being black on a colored background or colored on a black background. The screen is sectioned off into 32 characters by 16 character rows.

The second mode is Semigraphic-4. This mode has a choice of eight colors or black and is alphanumeric compatible. The compatibility in this case means the SG-4 mode requires the same amount of display RAM (512 bytes) and each byte or character fills up the same amount of display area on the television screen. In other words, an alphanumeric "A" could have that same area cut up into four blocks with any combination of those blocks lit up. The color choice is one color per character or memory location or byte depending on how you care to define the information.

Semigraphics-6 is the third mode and is basically the same as SG-4 except the blocks are cut up into six pieces and a choice of two four color sets must be made with the Color Set Select pin on the VDG.

The next eight modes are referred to as full graphics modes and have increasing density and memory requirements. The memory locations relate to an area on the screen as in all other modes. The next four modes mentioned will allow a choice of two four color sets. A 64 x 64 graphics mode is three horizontal lines by four pictels or dots that the VDG lights up. This mode requires one kilobyte of memory. A 128 x 64 mode uses two dot clocks by three horizontal lines and two kilobytes of memory. A 128 x 96 graphics mode is two dot clocks wide by two horizontal lines high and uses three kilobytes of memory space. A 128 x 192 graphics mode requires six kilobytes for a two dot clock wide by one horizontal line high display.

The following four modes only turn a pictel on or off and the on portion can be either green or white depending on the voltage applied to the Color Set Select pin. The element sizes have already been given for three of these modes but the memory requirements are different.

The 128 x 64 mode requires one kilobyte of memory. The 128 x 96 mode requires 1.5 kilobytes of memory. The 128 x 192 mode requires three kilobytes of RAM. The final and most dense mode is the 256 x 192 graphics mode and requires six kilobytes of memory. This mode maps the memory one bit for one pictel on the television screen for a total of 49,152 bits. The density of this mode will allow development of your own alphanumerics of special characters and shapes.

For alphanumeric characters of the same size as those in the alphanumeric mode, a 5 x 7 character font, with one blank line horizontally and one blank line vertically between each character, a total of 44 characters per character row can be achieved with a total of 21 character rows. This would give an overall character font of 6 x 8. Refer to Table 1 for a breakdown of the VDG modes.

For further versatility the VDG may be purchased with the non-interlace or interlace mask option, the interlace version costing slightly more. For those unfamiliar with the term interlace, a television has a frame composed of two fields, each 262½ horizontal lines for a total of 525 lines displayed on the screen.

The first field scans from the upper left hand corner to the middle of the bottom line skipping every other line as the electron beam travels downward. The second field scans from the middle of the top line to the end of the bottom line filing in the lines the first field skipped over. The interlace version scans both fields while the non-interlace only scans every other line (basically field one).

A few reasons for the availability of the two versions to customers are: 1) the non-interlace is a steady display which has neither dot crawl nor zipper effect and does not flicker at a 30 Hz rate, but scans at a 60 Hz rate allowing for an almost unperceivable screen refresh; 2) the interlace version fills in between the lines resulting in a "fuller" or more complete looking picture; 3) by separation and synchronization of odd and even fields through some external circuitry, it is possible to overlay two entirely different pictures on the TV screen. An example of this would be to overlay alphanumeric characters at the bottom of the screen on newscasts or any other broadcasts in order for the deaf or hard of hearing to enjoy television programs and announcements.

The other part of the basic display circuitry is the MC1372 Color Television Video Modulator. The chip generates a composite modulated RF video signal for the television set. The modulation of channel 3 or 4 carrier waves is possible as well as the ability to accept a sound carrier.

There are a minimum of parts required to operate this device. It requires only a single 5-volt power supply and has a TTL compatible clock output (one LS-TTL load) which can have an adjustable duty cycle with the addition of a 10K ohm potentiometer on pin 3 between supply and ground. A 50% duty cycle is achieved with no connection on pin 3.

The output pulse is basically a square wave with a frequency of 3.58 MHz which is the same frequency as the chrominance subcarrier oscillator. The output clock pulse is phase shifted for feedback to the chip. The modulator output amplitude and polarity correspond to the voltage difference between the chroma bias or Color Reference pin (pin 6) and the two color pins ϕA and ϕB (pins 7 and 5). The Chroma Modulator Output (pin 8) provides the vectorial sum of ϕA and ϕB which is fed back into the Chrominance Input (pin 10) which then RF modulates the signal. The RF tank which determines the channel or RF oscillator frequency is between pins 13 and 14. The final modulated output is pin 12 and can then be interconnected to a television set.

Table 1. Detailed Description of VDG Modes.

VDG PINS									COLOR			TV SCREEN		VDG DATA BUS	COMMENTS
$\overline{MS}$	$\overline{A}/G$	$\overline{A}/S$	$\overline{INT}/EXT$	GM2	GM1	GM0	CSS	INV	Character Color	Background	Border	Display Mode	Detail		
1	0	0	0	X	X	X	0 1	0 / 1 0 / 1	Green Black Orange Black	Black Green Black Orange	Black Black	32 Characters in columns 16 Characters in rows	8 dots 12 dots 7 5	extra / ASCII code	The ALPHANUMERIC INTERNAL mode uses an internal character generator which contains the following five dot by seven dot characters: @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\] ↑ ← SP !"#$%&'()*+,-./0123456789:;<=>?. The six bit ASCII code leaves two bits free and these may be externally connected to the mode pins ($\overline{A}$/G, $\overline{A}$/S, $\overline{INT}$/EXT, GM2, GM1, GM0, CSS or INV).
1	0	0	1	X	X	X	0 1	0 / 1 0 / 1	Green Black Orange Black	Black Green Black Orange	Black Black	32 Characters in columns 16 Characters in rows		one row of custom characters	The ALPHANUMERIC EXTERNAL mode uses an external character generator as well as a row counter. Thus, custom character fonts are graphic symbol sets with up to 256 different eight dot X 12 dot "characters" may be displayed.
1	0	1	0	X	X	X	X	X	Lx C2 C1 C0 0 X X X 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1	Color Black Green Yellow Blue Red Buff Cyan Magenta Orange	Black	64 Display elements in columns 32 Display elements in rows	L_3 L_2 L_1 L_0 one element	C_2 C_1 C_0 L_3 L_2 L_1 L_0	The SEMIGRAPHICS FOUR mode uses an internal "course graphics" generator in which a rectangle (eight dots by twelve dots) is divided into four equal parts. The luminance of each part is determined by a corresponding bit on the VDG data bus. The color of illuminated parts is determined by three bits.
1	0	1	1	X	X	X	0 1	X	Lx C1 C0 0 X X 1 0 0 1 0 1 1 1 0 1 1 1 0 X X 1 0 0 1 0 1 1 1 0 1 1 1	Color Black Green Yellow Blue Red Black Buff Cyan Magenta Orange	Black	64 Display elements in columns 48 Display elements in rows	L_5 L_4 L_3 L_2 L_1 L_0	C_1 C_0 L_5 L_4 L_3 L_2 L_1 L_0	The SEMIGRAPHIC SIX mode is similar to the SEMIGRAPHIC FOUR mode with the following differences: The eight dot by twelve dot rectangle is divided into six equal parts. Color is determined by the two remaining bits.
1	1	X	X	0	0	0	0 1	X	C1 C0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1	Color Green Yellow Blue Red Buff Cyan Magenta Orange	Green Buff	64 Display elements in columns 64 Display elements in rows	E_3 E_2 E_1 E_0	C_1 C_0 C_1 C_0 C_1 C_0 C_1 C_0	The GRAPHICS ONE C mode uses a maximum of 1024 bytes of display RAM in which one pair of bits specifies one picture element.
1	1	X	X	0	0	1	0 1	X	Lx 0 1 0 1	Color Black Green Black Buff	Green Buff	128 Display elements in columns 64 Display elements in rows	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	The GRAPHICS ONE R mode uses a maximum of 1024 bytes of display RAM in which one bit specifies one picture element.
1	1	X	X	0	1	0	0 1	X	Same color as Graphics one C		Green Buff	128 Display elements in columns 64 Display elements in rows	E_3 E_2 E_1 E_0	C_1 C_0 C_1 C_0 C_1 C_0 C_1 C_0	The GRAPHICS TWO C mode uses a maximum of 2048 bytes of display RAM in which one pair of bits specifies one picture element.
1	1	X	X	0	1	1	0 1	X	Same color as Graphics one R		Green Buff	128 Display elements in columns 96 Display elements in rows	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	The GRAPHICS TWO R mode uses a maximum of 1536 bytes of display RAM in which one bit specifies one picture element.
1	1	X	X	1	0	0	0 1	X	Same color as Graphics one C		Green Buff	128 Display elements in columns 96 Display elements in rows	E_3 E_2 E_1 E_0	C_1 C_0 C_1 C_0 C_1 C_0 C_1 C_0	The GRAPHICS THREE C mode uses a maximum of 3072 bytes of display RAM in which one pair of bytes specifies one picture element.
1	1	X	X	1	0	1	0 1	X	Same color as Graphics one R		Green Buff	128 Display elements in columns 192 Display elements in rows	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	The GRAPHICS THREE R mode uses a maximum of 3072 bytes of display RAM in which one bit specifies one picture element.
1	1	X	X	1	1	0	0 1	X	Same color as Graphics one C		Green Buff	128 Display elements in columns 192 Display elements in rows	E_3 E_2 E_1 E_0	C_1 C_0 C_1 C_0 C_1 C_0 C_1 C_0	The GRAPHICS SIX C mode uses a maximum of 6144 bytes of display RAM in which one pair of bits specifies one picture element.
1	1	X	X	1	1	1	0 1	X	Same color as Graphics one R		Green Buff	256 Display elements in columns 192 Display elements in rows	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0	The GRAPHICS SIX R mode uses a maximum of 6144 bytes of display RAM in which one bit specifies one picture element.

TYPICAL MINIMUM AND MAXIMUM SYSTEMS

The VDG, a RAM or ROM (a ROM would be preferable since no MPU is around to store display data), and the linear modulator make a complete display system. Refer to Figure 1 for a basic display block diagram. The VDG is controlled by eight lines which may be hardwired, logically controlled through the use of TTL (Transistor Transistor Logic) and/or a PIA (Peripheral Interface Adaptor), or tied to the data lines of another block of RAM.

Before continuing, a brief explanation about the PIA is due. The MC6820 is a universal device for interfacing the MPU to peripheral instruments and equipment such as terminals, printers, cassette decks, keyboards, etc. with no or minimal external logic through two 8-bit bidirectional peripheral data buses and four handshake control lines.

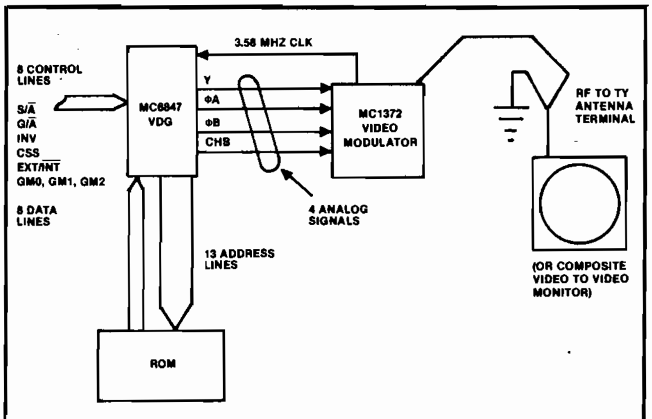


Figure 1. Block diagram of a simple display system.

During system initialization each of the sixteen data lines may be individually programmed as an input or output with a number of variations available for the type of handshake, control, or interrupt needed. A brief discussion of the above VDG control methods will be discussed shortly. The VDG increments through the address bus to the display RAM or ROM. The memory in turn outputs data to the VDG which interprets each byte according to the input on the control lines. The VDG outputs the video information on one pin and the chrominance information on two other pins.

Sync 1.0V

Blank 0.75V

Black 0.7V

White Low 0.62V

White Medium 0.5V

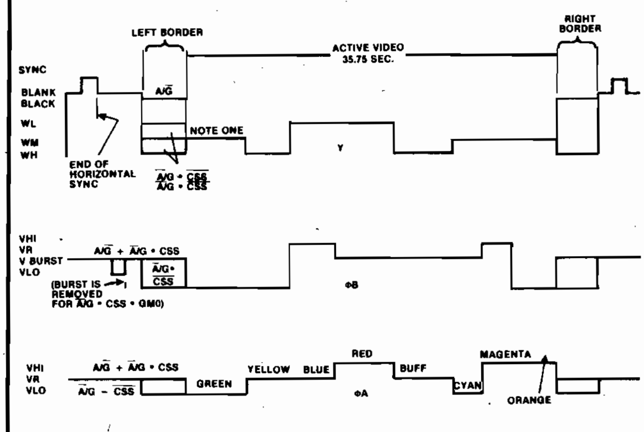
White High 0.38V

Figure 2. Nominal Luminance Levels.

(See Figure 2 for nominal luminance levels; see Figures 3 and 4 for horizontal and vertical output waveforms from the VDG.) The MC1372 modulator puts out the needed RF to the antenna terminals of a color or black and white television set. The outputs of the VDG feed into the RF oscillator modulator, which not only develops the RF carrier and final composite video signal complete with color burst, but also generates a 3.58 MHz crystal controlled clock for the VDG.

Table 2. Recommended Chroma-Luma Signals

	Pin #9 Luminance Input (Vdc)	Pin #7 Color A (Vdc)	Pin #6 Color Ref. (Vdc)	Pin #5 Color B (Vdc)
Sync	1.0	1.5	1.5	1.5
Blanking	0.75	1.5	1.5	1.5
Burst	0.75	1.5	1.5	1.25
Black	0.70	1.5	1.5	1.5
Green	0.50	1.0	1.5	1.0
Yellow	0.38	1.5	1.5	1.0
Blue	0.62	1.5	1.5	2.0
Red	0.62	2.0	1.5	1.5
Cyan	0.50	1.0	1.5	1.5
Magenta	0.50	2.0	1.5	2.0
Orange	0.50	2.0	1.5	1.0
Buff	0.38	1.5	1.5	1.5



NOTE:
1. The 3.58MHz Video must be in phase for every horizontal line, every field and every frame.
2. Horizontal timing non-interlaced option.

Figure 3. Video and Chrominance Output Waveform Relationships.

Refer to Table 2 for nominal chroma and luma input signals to the MC1372.

Now for a brief discussion on control methods. Hardwiring the control lines or using switches will allow manual operational control. The use of TTL or a PIA enables the user to switch modes on the fly under software control. This method must be under constant supervision of the MPU. The third method involves using twice as much RAM. Control RAM uses 8 bits and display data uses 8 bits (Figure 5). The MPU accesses two blocks of RAM, each 6K by 8 of bits making the available RAM look like 21K by 8 bits.

The software will initially have to know where mode information goes with respect to the display data. When the MPU is through reading and/or writing to the RAM, the VDG takes over and the blocks of RAM are simultaneously selectable by the VDG. This gives the MC6847 a memory block of 6K by 16 bits. This is allowing for a maximum system with a maximum amount of RAM. A reduction to 13 bits of RAM may be achieved by connecting some don't care data lines to

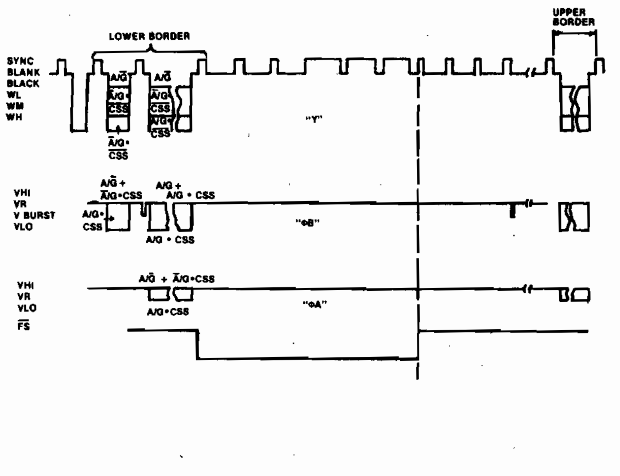


Figure 4. Vertical Output Non-Interlaced Option.

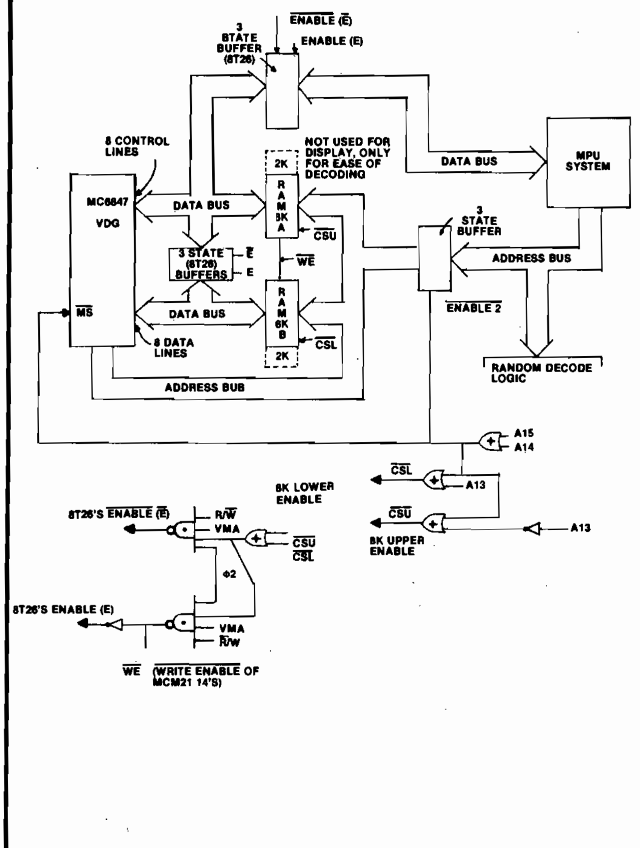


Figure 5. Maximum System.

each other as in Figure 6, and if the Color Set Select pin is hardwired or connected otherwise (say to a PIA) then 6K by 12 bits would allow for a maximum system.

Table 3 decodes the pin connections for the various modes. Speaking of decoding, all the necessary signals for the maximum system are developed in Figure 5. The Read/Write and Enable signals are clocked with ϕ2 and Valid Memory Address inputs. Selection of the upper and lower 8K memory blocks is available, but the user will need to decode the individual RAMs in each block if 8K RAMs are not available. Similar yet "smaller" systems could use 1K by 13 or 12 bits using this idea. All modes could still be used. The software program in ROM would keep the display system under con-

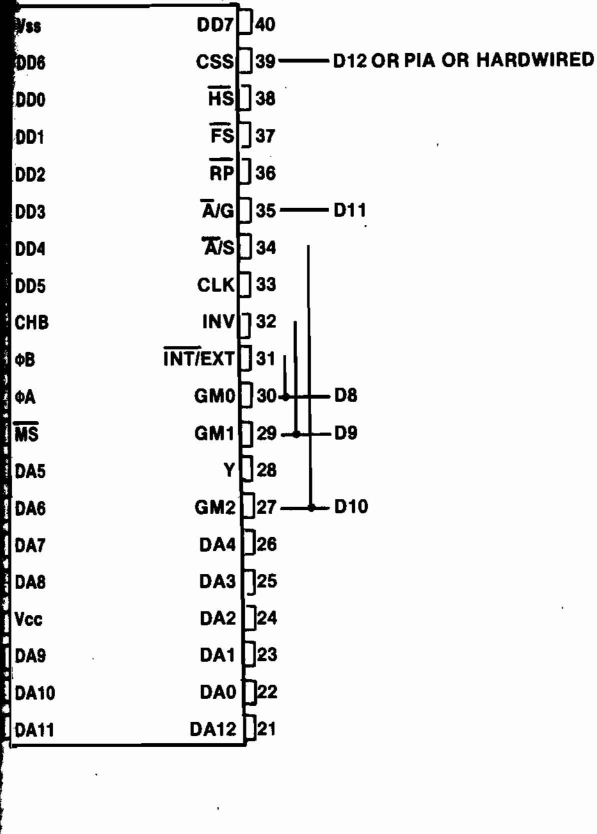


Figure 6. VDC Mode Control Connections.

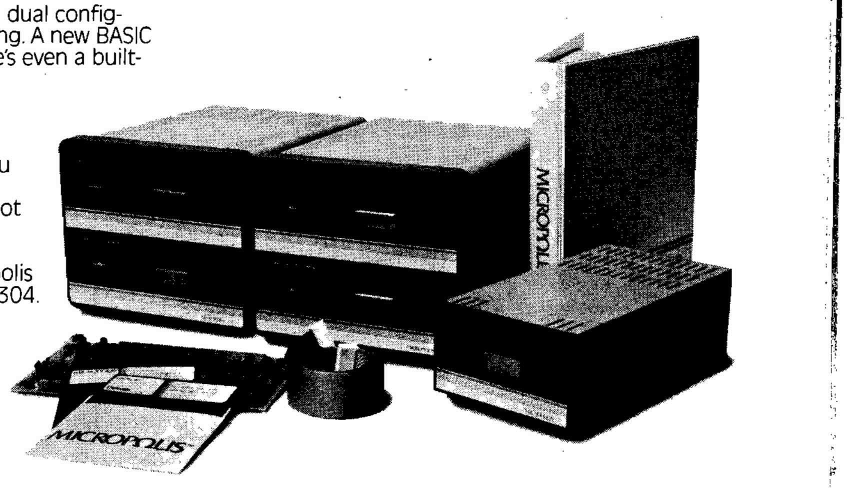
Table 3. VDG Mode Selection

D_{11}	D_{10}	D_9	D_8	MODE.
0	0	0	0	Internal Alpha Numerics
0	0	0	1	External Alpha Numerics
0	0	1	0	Internal Alpha Numerics Inverted
0	0	1	1	External Alpha Numerics Inverted
0	1	X	0	Semigraphics 4
0	1	X	1	Semigraphics 6
1	0	0	0	Graphics Mode 0
1	0	0	1	Graphics Mode 1
1	0	1	0	Graphics Mode 2
1	0	1	1	Graphics Mode 3
1	1	0	0	Graphics Mode 4
1	1	0	1	Graphics Mode 5
1	1	1	0	Graphics Mode 6
1	1	1	1	Graphics Mode 7

stant supervision. The program would need to determine the number of scan lines desired in any particular mode enabling ease of mode changing, where the memory address ought to be (as far as the VDG is concerned), and if object code is used, where it is and where it ought to go in memory.

These considerations as well as general housekeeping must be taken into account by the system's microprocessor. Another way of enhancing performance with fewer parts is to use a bi-phase method. If the VDG is used with a MC6800 family microprocessor then 6K of RAM could be displayed using only 1K of actual in-system RAM.

If the Interlaced VDG is used, a flip flop could choose between memory banks of 6K each (maximum type system — less memory could be used incorporating some of the other



ideas) allowing for smaller characters on the screen since it is now effectively twice as dense as before. The flip flop is toggled by the Field Synchronization pulse enabling different information to be displayed every field change instead of having the same dot for both fields, thus allowing mapping of 98,304 bits.

SOFTWARE

Three programs are incorporated within. The first program shown in Listing 1 is extremely short and uses two modes: Alphanumerics and Semigraphics-4 mode. The data in the display memory is incremented every location to show the color and character capabilities. The alphanumerics inverted and non-inverted characters will appear first. Then two columns of four rows each with blocks of color will appear from left to right like this: green, yellow, blue, red, buff, cyan, magenta and orange. They will also appear in all the possible combinations of illumination. The VDG TEST program only loads up the display memory with successively incremented numbers and then returns control to the monitor program.

The second program shown in Listing 2 is a short piece of coding to demonstrate terminal possibilities. The program was written to run with a MIKBUG 2.0 monitor, therefore, the input routine is at $F878. If an Exorcisor is being used, replace $F878 with $F015, the input without parity routine INCHNP. The program is extremely limited; it allows only alphanumeric characters (inverted and noninverted). This could use Semigraphics-4 if the user reconfigures the software for control characters putting the desired information into the display memory. In this instance the most significant bit, bit 7, indicates Alpha or Semigraphics-4.

The only special functions allowed are backspacing and escaping to the monitor. Other commands may easily be added by the user. Oddly enough the backspace key and the escape key were chosen for their respective functions. The program is not for a stand alone system; in other words the system used must already have a fully functional terminal. The program returns to the top left hand corner of the screen after the last character is input at the bottom right hand corner of the television screen.

This program may be added to quite easily. All the control characters should be checked over before any character is thrown into the display RAM and those comparisons and branches should be inserted at $0222 between the BEQ ESC and STA A 00, X instructions. The actual code for implementation of each additional control character is placed after the ESC SWI instruction (ESC is the label and SWI is the mnemonic for the instruction Software Interrupt).

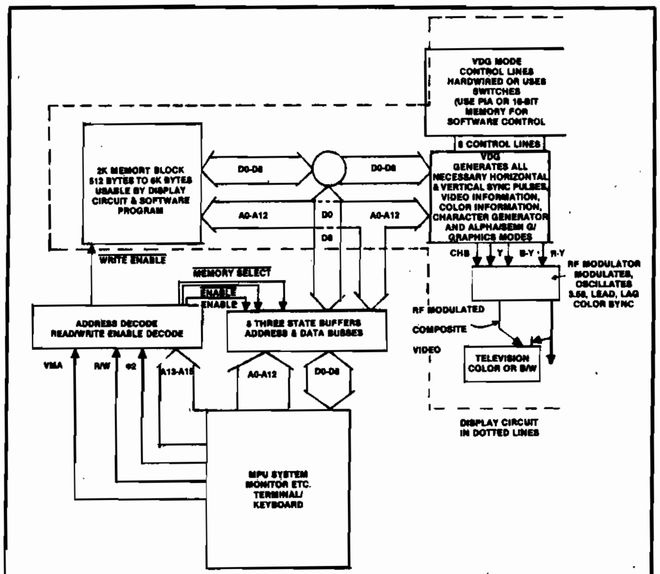


Figure 7. Block Diagram of VDG Board.

The third (and last) program is for cursor control. Again, if the Exorcisor is used change $F878 to $F015. The program is fairly well documented and explains itself. The cursor program assembly listing is shown in Listing 3.

The method for creating color graphics is as follows: First block off an existing picture and enlarge it on a similar screen-sized piece of graph paper. Second make a transparency of the enlarged picture and tape it over the television set. Finally use the cursor program to color in the appropriate colors behind the transparency.

HARDWARE

The system used is either an EXORCISOR or an EXORCISOR Compatible System such as a MIKBUG 2.0 controlled D2 Kit. The block diagram of the existing hardware is depicted in Figure 7. The display board interface to the Exorcisor Bus shown in Figure 8 consists of three quad three-state bus transceivers, three HEX buffers, four RAM chips (2K total), one VDG, one Linear Part and four SSI TTL logic parts for decoding. The total chip count is 16 to decode 2K bytes of static RAM and have alpha and full color graphics capability. Further decoding is possible when more RAM is desired in the system.

SWITCH SELECT FOR VOG MODES

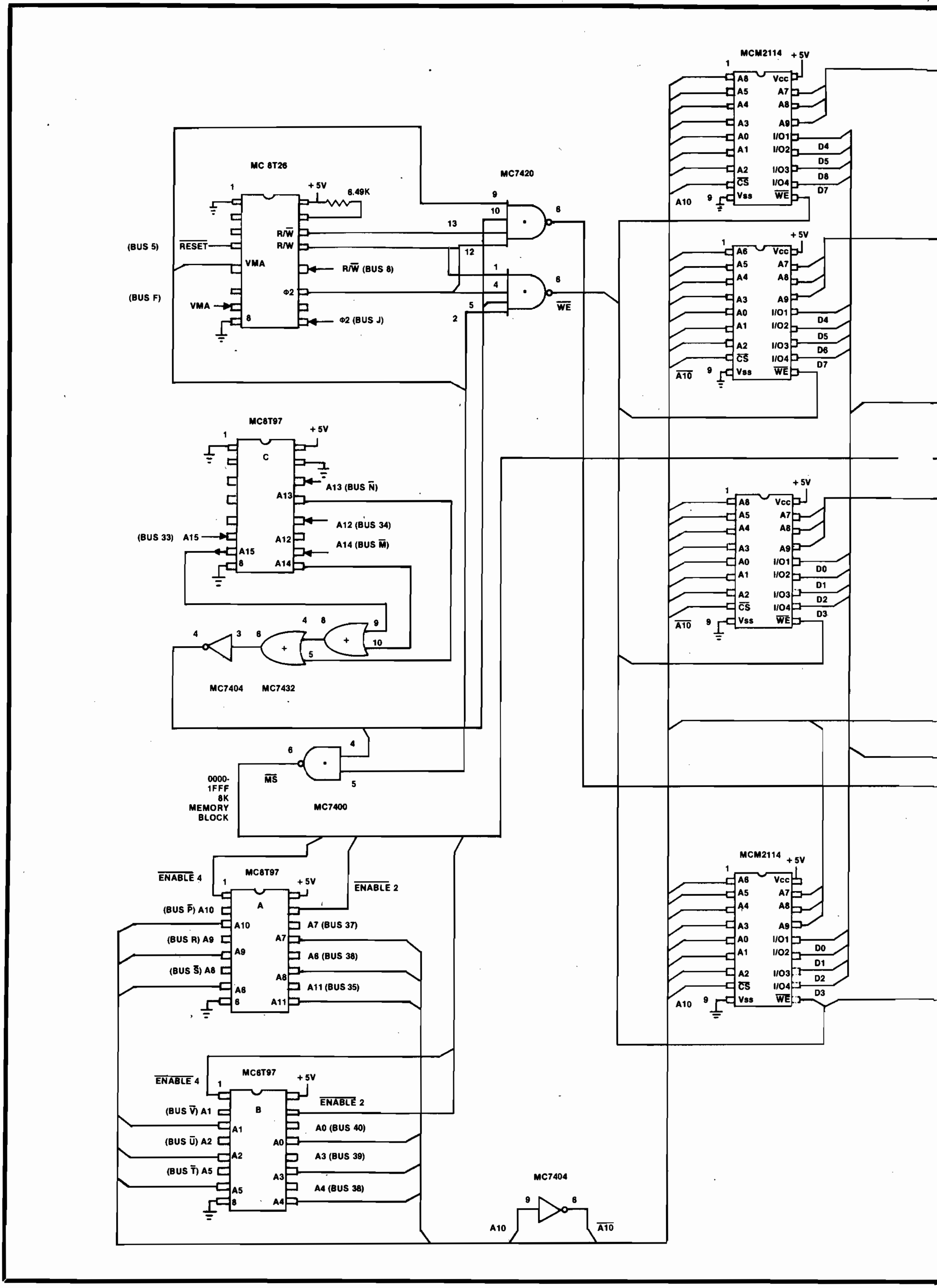
Switch #	VOG Pin	Function	256X192 GRAPHICS	128X192 COLOR GRAPHICS	128X96 GRAPHICS	128X96 COLOR GRAPHICS	128X96 GRAPHICS	128X64 COLOR GRAPHICS	128X64 GRAPHICS	64X64 COLOR GRAPHICS	SEMI GRAPHICS 6	SEMI GRAPHICS 4	EXT ALPHA INV	EXT ALPHA	INT ALPHA INV	INT ALPHA
		INV	X	X	X	X	X	X	X	X	X	X	1	0	1	0
1	30	GMØ	1	0	1	0	1	0	1	0	X	X	X	X	X	X
2	29	GM1	1	1	0	0	1	1	0	0	X	X	X	X	X	X
3	27	GM2	1	1	1	1	0	0	0	0	X	X	X	X	X	X
4	31	$\overline{INT}$/EXT	X	X	X	X	X	X	X	X	1	0	1	1	0	0
5	34	$\overline{ALPHA}$/ SEMI GRAPHICS	X	X	X	X	X	X	X	X	1	1	0	0	0	0
6	35	$\overline{ALPHA}$/ GRAPHICS	1	1	1	1	1	1	1	1	0	0	0	0	0	0
7	39	CSS														

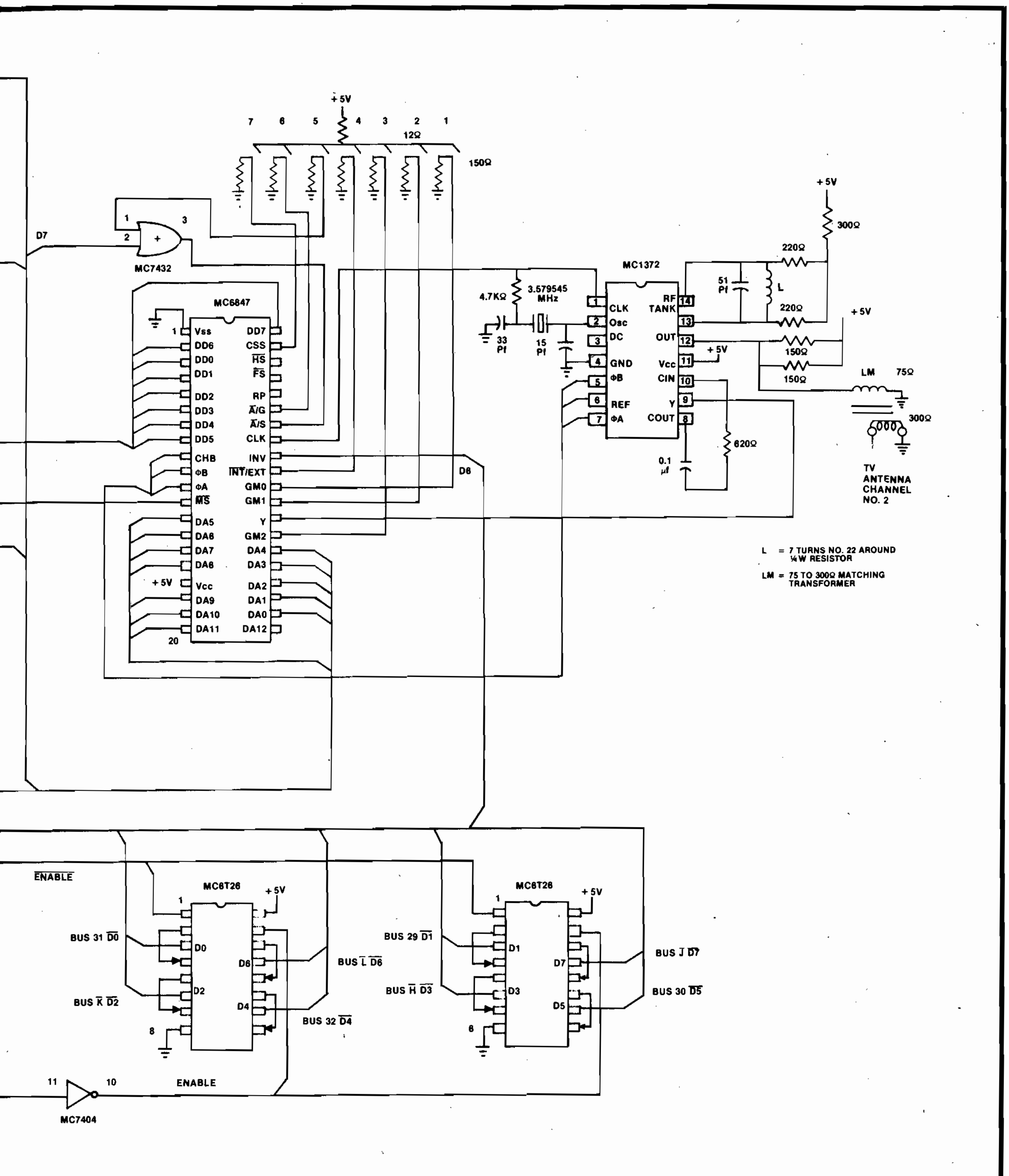
Figure 9. VDG Mode Switches

Note the OR gate with the output to $\overline{A}$/S of the VDG. This gate enables either software or hardware control of the Semigraphics modes. Software control is desired when switching between Alpha and Semigraphics-4 mode. The hardware switch input to the OR gate is to cancel out the effects of Data bit 7 on the control pins when Semigraphics-6 mode is desired. If the gate were not present the switch and resistors would conflict with the data bus bit 7 causing sections on the screen to flicker, resulting in unreliably displayed data.

All control pins (as well as the data bus) must have "solid" information on them and must *not be left floating* at an unknown state. Good grounding of the connectors around the RF output and shielding high frequency areas will enhance the appearance of the television display. The capacitor values around the 3.58 crystal are not extremely critical, but for individual systems a trimmer capacitor may replace the 15 pf capacitor. The mode choice, via the switches, is shown in Figure 9.☐

Figures & Listings follow


MCM2114
+5V
MC 8T26
MC7420
6.49K
(BUS 5)
RESET
R/W
VMA
R/W (BUS 8)
(BUS F)
VMA
Φ2
Φ2 (BUS J)
WE
MC8T97
A13 (BUS N)
A12 (BUS 34)
A14 (BUS M)
(BUS 33) A15
MC7404
MC7432
MC7400
MS
0000-
1FFF
8K
MEMORY
BLOCK
ENABLE 4
ENABLE 2
(BUS P) A10
(BUS R) A9
(BUS S) A8
A7 (BUS 37)
A6 (BUS 36)
A11 (BUS 35)
(BUS V) A1
(BUS U) A2
(BUS T) A5
A0 (BUS 40)
A3 (BUS 39)
A4 (BUS 38)
A10
D0
D1
D2
D3
D4
D5
D6
D7



+5V
7 6 5 4 3 2 1
12Ω
150Ω
D7
MC7432
MC6847
Vss
DD6
DD0
DD1
DD2
DD3
DD4
DD5
CHB
ΦB
ΦA
MS
DA5
DA6
DA7
DA8
+5V
Vcc
DA9
DA10
DA11
20
DD7
CSS
HS
FS
RP
A/G
A/S
CLK
INV
INT/EXT
GM0
GM1
Y
GM2
DA4
DA3
DA2
DA1
DA0
DA12
D6
MC1372
4.7KΩ
3.579545 MHz
33 Pf
15 Pf
CLK
Osc
DC
GND
ΦB
REF
ΦA
RF TANK
OUT
Vcc
CIN
Y
COUT
+5V
51 Pf
L
220Ω
220Ω
300Ω
150Ω
150Ω
LM
75Ω
300Ω
620Ω
0.1 µf
TV ANTENNA CHANNEL NO. 2
L = 7 TURNS NO. 22 AROUND ¼W RESISTOR
LM = 75 TO 300Ω MATCHING TRANSFORMER
ENABLE
MC8T26
+5V
BUS 31 D0
BUS K D2
D0
D2
D6
D4
BUS L D6
BUS 32 D4
8
BUS 29 D1
BUS H D3
D1
D3
D7
D5
BUS J D7
BUS 30 D5
11
10
ENABLE
MC7404

Figure 8.

LISTING 1

```
                     NAM VDGTEST
               *   THIS PROGRAM DEMONSTRATES THE ALPHANUMERICS MODE
               *  AND THE SEMI GRAPHICS 4 ( 8 COLORS PLUS BLACK MODE)
               *
0210                 ORG $0210         BEGINING OF DISPLAY PROGRAM
0210 CE 00 00        LDX #$0000        START ADD OF DISPLAY IN X REG
0213 86 00           LDA A #$00        ACC A INCREMENTS DATA
0215 A7 00     CONT  STA A 00,X        LOAD UP DISPLAY MEMORY
0217 4C              INC A             NEXT PIECE OF INFO
0218 08              INX               NEXT DISPLAY LOCATION
0219 8C 02 00        CPX #$0200        ENTIRE DISPLAY FULL
021C 27 03           BEQ STOP          IF FULL THEN STOP PROG
021E 7E 02 15        JMP CONT          IF NOT INCREMENT DATA AND MEMORY
0221 3F        STOP  SWI               RETURN TO MONITOR
                     END
    NO ERROR(S) DETECTED


  SYMBOL TABLE:

S1130210CE00008600A7004C088C020027037E0253
S1050220153F84S9
MIKBUG 2.0
*

S1130210CE00008600A7004C088C020027037E0253
S1050220153F84S9
MIKBUG 2.0
*
```

LISTING 2

```
                     NAM VDGMOT
               *   THIS IS A VIDEO DISPLAY GENERATOR MINI-OUTPUT TERMINAL
               *   IT ACCEPTS INPUTS FROM THE TERMINAL AND DISPLAYS THE INFO
               *   ON THE TELEVISION SCREEN: NORMALLY INVERTED FOR
               *   ALPHA AND NONINVERTED FOR NUMERICS AND SPECIAL CHARACTERS
               *   THE OTHER (INVERT OR NONINVERT) IS ACHIEVED BY
               *   DEPRESSING THE CONTROL KEY AND THEN THE DESIRED CHARACTER
0210                 ORG $0210        BEGINING OF PROGRAM
0210 CE 00 00  START LDX #$0000       LOAD X REG WITH START ADD OF SCREEN RAM
0213 86 20           LDA A #$20       LOAD AN ASCII BLANK INTO ACC A
0215 A7 00           STA A 00,X       DISPLAY CONTENTS OF ACC A
0217 BD F8 78  CONT  JSR $F878        JUMP TO INPUT ROUTINE INCH
021A 81 08           CMP A #$08       ASCII BS?  BACKSPACE
021C 27 0F           BEQ BS           IF SO GO DECREMENT X REG
021E 81 1B           CMP A #$1B       ASCII ESC?  EXCAPE
0220 27 0E           BEQ ESC          IF SO JUMP OUT OF PROGRAM
0222 A7 00           STA A 00,X       PUT DATA IN DISPLAY
0224 08              INX              NEXT LOCATION IN DISPLAY
0225 8C 02 00  TOP   CPX #$0200       AT END OF DISPLAY?
0228 27 E6           BEQ START        IF SO: GO TO TOP OF SCREEN
022A 7E 02 17        JMP CONT         IF NOT:  GO INPUT NEXT CHARACTER
022D 09        BS    DEX              BACKSPACE BY DECREMENTING X REG
022E 20 F5           BRA TOP          CONTINUE THROUGH PROGRAM
0230 3F        ESC   SWI              RETURN TO MONITOR
               *   VDGMOT IS USABLE IN THE SEMIGRAPHICS & GRAPHICS MODES
                     END
    NO ERROR(S) DETECTED
  SYMBOL TABLE:

S1130210CE00008620A700BDF878810827UF811B37
S113022027UEA700088C020027E67E0217092UF596
S10402303F8AS9
MIKBUG 2.0
*
```

LISTING 3

```
PAGE  001    CURSOR

00002                        NAM    CURSOR
00003                *
00004                *   THIS PROGRAM DISPLAYS A CURSOR WITH MOVEMENT AND COLOR SELECTION
00005                *   THE MC6847 VIDEO DISPLAY GENERATOR
00006                *   IS IN ANY FOUR COLOR GRAPHICS MODE
00007                *
00008                *
00009                *    MEMORY SETUP:   THERE ARE FOUR INDIVIDUAL 'CHARACTERS'
00010                *    BY THREE SCAN LINES OF INFORMATION PER LOCATION (OR BLOCK) IN MEMORY
00011                *    COLOR GOES INTO ONE OF FOUR CHAR POSITIONS
00012                *    WITHIN THE BLOCK OR MEMORY LOCATION EVERY TIME
00013                *    COLOR INFORMATION IS INPUT FROM THE TERMINAL
00014                *
00015                *    ****************************
00016                *    *      *      *      *      *

00119A 4073 A7 00     A        STAA  00,X     *   LAST LOCATION
00120A 4075 09                 DEX            DECREMENT INDEX REG
00121A 4076 A6 00     A        LDAA  00,X
00122A 4078 B7 4001   A        STAA  TEMP5    UPDATE TEMP REG
00123A 407B 7F 4000   A        CLR   TEMPO
00124A 407E 20 8E 400E CURIN4  BRA   CURINC   STEPPING STONE
00125A 4080 81 49     A INCZ   CMPA  #$49     ASCII I INCREMENT ONE LOCATION & PUT ZEROES IN LAST LOCATION
00126A 4082 26 65 40E9         BNE   DCJMP1   CHECK NEXT INSTRUCTION IF NOT I
00127A 4084 86 00     A        LDAA  #$00     CLEAR CURSOR
00128A 4086 A7 00     A        STAA  00,X     DISPLAY  CURSOR
00129A 4088 08                 INX            NEXT DISPLAY LOCATION
00130A 4089 A6 00     A        LDAA  00,X
00131A 408B B7 4001   A        STAA  TEMP5    UPDATE TEMP REG
00132A 408E 7F 4000   A        CLR   TEMPO
00133A 4091 20 EB 407E         BRA   CURIN4
00134A 4093 20 CD 4062 INCH2   BRA   INCH3    STEPPING STONE
00135A 4095 F6 4000   A MAIN1  LDAB  TEMPO
00136A 4098 C1 00     A        CMPB  #$00
00137A 409A 27 38 40D4         BEQ   SHIFT6
00138A 409C C1 01     A        CMPB  #$01
00139A 409E 27 2A 40CA         BEQ   SHIFT4
00140A 40A0 C1 02     A        CMPB  #$02
00141A 40A2 27 15 40B9         BEQ   SHIFT2   THESE DECIDE WHICH PORTION OF A LOCATION NEEDS DATA
00142A 40A4 7F 4000   A        CLR   TEMPO    THIS IS THE FOURTH CHAR IN BLOCK OF CHARS
00143A 40A7 E6 00     A        LDAB  00,X
00144A 40A9 C4 FC     A        ANDB  #$FC     LAST CHAR IN BLOCK
00145A 40AB E7 00     A        STAB  00,X     CONCATONATE NEW INFO WITH OLD
00146A 40AD AA 00     A        ORAA  00,X
00147A 40AF A7 00     A        STAA  00,X     DISPLAY FINAL UPDATED CHAR BLOCK
00148A 40B1 B7 4001   A        STAA  TEMP5    UPDATE TEMPORARY STORAGE
00149A 40B4 08                 INX            NEXT CURSOR BLOCK
00150A 40B5 20 C7 407E CURIN3  BRA   CURIN4   STEPPING STONE
00151A 40B7 20 A9 4062 INCH7   BRA   INCH3    STEPPING STONE
00152A 40B9 48         SHIFT2  ASLA
00153A 40BA 48                 ASLA
00154A 40BB E6 00     A        LDAB  00,X
00155A 40BD C4 F3     A        ANDB  #$F3     THIRD CHAR IN BLOCK
00156A 40BF E7 00     A STORE  STAB  00,X
00157A 40C1 AA 00     A        ORAA  00,X     CONCATONATE NEW INFO WITH OLD DISPLAY INFO
00158A 40C3 A7 00     A        STAA  00,X
00159A 40C5 7C 4000   A        INC   TEMPO
00160A 40C8 20 C9 4093         BRA   INCH2
00161A 40CA 48         SHIFT4  ASLA
00162A 40CB 48                 ASLA
00163A 40CC 48                 ASLA
00164A 40CD 48                 ASLA
00165A 40CE E6 00     A        LDAB  00,X
00166A 40D0 C4 CF     A        ANDB  #$CF     SECOND CHAR IN BLOCK
00167A 40D2 20 EB 40BF         BRA   STORE
00168A 40D4 E6 00     A SHIFT6 LDAB  00,X
00169A 40D6 F7 4001   A        STAB  TEMP5    SAVE OLD DATA
00170A 40D9 48                 ASLA           SHIFT COLOR INFO IN ACC A
00171A 40DA 48                 ASLA           OVER TO CORRECT POSITION
00172A 40DB 48                 ASLA           TO 'OR' WITH DISPLAY INFO
00173A 40DC 48                 ASLA
00174A 40DD 48                 ASLA
00175A 40DE 48                 ASLA
00176A 40DF E6 00     A        LDAB  00,X
00178A 40E1 C4 3F     A        ANDB  #$3F     FIRST CHAR IN BLOCK
00179A 40E3 20 DA 40BF         BRA   STORE
00180A 40E5 20 CE 40B5 CURIN2  BRA   CURIN3   STEPPING STONE
00181A 40E7 20 CE 40B7 INCH8   BRA   INCH7    STEPPING STONE
00182A 40E9 20 00 40EB DCJMP1  BRA   DECJMP   STEPPING STONE
00183                  *
00184                  *
00185                  *
00186A 40EB 81 4A     A DECJMP CMPA  #$4A     ASCII  J DECREMENT JUMP 10 LOCATIONS DOWN & LEAVE OLD INFORMATION
00187A 40ED 26 18 4107         BNE   INCOLD
00188A 40EF F6 4001   A        LDAB  TEMP5
00189A 40F2 E7 00     A        STAB  00,X
00190A 40F4 C6 09     A        LDAB  #$09
00191A 40F6 09         AGAIN   DEX
00192A 40F7 C1 00     A        CMPB  #$00
00193A 40F9 27 EA 40E5         BEQ   CURIN2
00194A 40FB A6 00     A        LDAA  00,X
00195A 40FD B7 4001   A        STAA  TEMP5    UPDATE TEMP DATA
00196A 4100 5A                 DECB
00197A 4101 20 F3 40F6         BRA   AGAIN
00198A 4103 20 E0 40E5 CURIN1  BRA   CURIN2   STEPPING STONE
00199A 4105 20 E0 40E7 INCH9   BRA   INCH8    STEPPING STONE
00200A 4107 81 55     A INCOLD CMPA  #$55     ASCII U INCREMENT JUMP 10 LOCATIONS UP & LEAVE OLD INFORMATION
00201A 4109 26 14 411F         BNE   LOLD
00202A 410B F6 4001   A        LDAB  TEMP5    RESTORE OLD INFO
00203A 410E E7 00     A        STAB  00,X
00204A 4110 C6 09     A        LDAB  #$09
00205A 4112 08         PLUS    INX
00206A 4113 C1 00     A        CMPB  #$00
00207A 4115 27 EC 4103         BEQ   CURIN1
00208A 4117 A6 00     A        LDAA  00,X
00209A 4119 B7 4001   A        STAA  TEMP5    UPDATE TEMP DATA
00210A 411C 5A                 DECB
00211A 411D 20 F3 4112         BRA   PLUS
00212A 411F 81 4C     A LOLD   CMPA  #$4C     ASCII L  LEAVE OLD INFO AND INCREMENT CURSOR
00213A 4121 26 0D 4130         BNE   CLEAR    NEXT INSTRUCTION
00214A 4123 F6 4001   A        LDAB  TEMP5
00215A 4126 E7 00     A        STAB  00,X
00216A 4128 08                 INX
00217A 4129 E6 00     A        LDAB  00,X
00218A 412B F7 4001   A        STAB  TEMP5    UPDATE CURSOR
```

```
00017                    *    *    *    *    *    *    *
00018                    *    ****************************
00019                    *    THE CURSOR MOVES ALONG EACH INDIVIDUAL
00020                    *    BLOCK AS THE DESIRED COLOR IS SELECTED.
00021                    *    A CHOICE OF ONE OF TWO 4 COLOR SETS IS POSSIBLE
00022                    *    WITH EITHER A SWITCH, A PIA OR RANDOM LOGIC
00023                    *    THE SAME HOLDS TRUE ABOUT THE OTHER CONTROL PINS
00024                    *
00025                    *
00026                    *          ACCEPTABLE INPUTS
00027                    *
00028                    *    W    WHITE OR BUFF COLOR GOES IN BLOCK
00029                    *    G    GREEN COLOR GOES IN BLOCK
00030                    *    C    CYAN    "     "    "    "
00031                    *    Y    YELLOW  "     "    "    "
00032                    *    M    MAGENTA COLOR GOES IN BLOCK
00033                    *    B    BLUE     "    "    "    "
00034                    *    O    ORANGE   "    "    "    "
00035                    *    R    RED      "    "    "    "
00036                    *    S    START PROGRAM OVER FROM BEGINNING
00037                    *    N    DECREMENT CURSOR ONE LOCATION AND LEAVE OLD DATA
00038                    *    E    END OR FINISH PROGRAM BY GOING BACK TO MONITOR
00039                    *    F    FINISH OR END PROGRAM BY GOING BACK TO MONITOR
00040                    *    D    DECREMENT ONE LOCATION AND CLEAR LAST LOCATION
00041                    *    I    INCREMENT ONE LOCATION AND CLEAR LAST LOCATION
00042                    *    J    JUMP 10 LOCATIONS DOWN IN MEMORY AND UP ON SCREEN & LEAVE OLD INFO
00043                    *    U    JUMP 10 LOCATIONS UP IN MEMORY AND DOWN ON SCREEN AND LEAVE OLD INFO
00044                    *    L    LEAVE OLD INFO AND INCREMENT CURSOR ONE LOCATION
00045                    *    CNTRL C    CLEARS THE AMOUNT OF DISPLAY MEMORY INPUT AFTER
00046                    *         THE CNTRL C COMMAND: 5 CLEARS 512 BYTES, 1 CLEARS 1K BYTES
00047                    *         2 CLEARS 2K BYTES, 3 CLEARS 3K BYTES, 6 CLEARS 6K BYTES
00048                    *         CURSOR IS PLACED AT THE END OF THE CLEARED MEMORY SPACE
00049                    *
00050                    *
00051                    *
00052                    *
00053                    *
00054                    *
00056A 4000                   ORG    $4000
00057                    *         BEGINING OF PROGRAM :  STARTS AT $4005
00058          F878   A INCH   EQU    $F878
00059          6000   A VIEW   EQU    $6000      BEGINNING OF DISPLAY RAM
00060A 4000    0001   A TEMPO  RMB    1          COUNTER FOR THE CURRENT CHARACTER POSITION WITHIN A BLOCK (ONE OF 4)
00061A 4001    0002   A TEMP5  RMB    2
00062A 4003    0002   A CLRMEM RMB    2          NUMBER OF BYTES TO CLEAR COMPARISON REGISTER
00063                    *
00064A 4005 7F 4000   A BEGIN  CLR    TEMPO      CLEAR ALL TEMPORARY STORAGE BYTES
00065A 4008 7F 4001   A        CLR    TEMP5
00066A 400B CE 6000   A        LDX    #VIEW      PUT DISPLAY LOCATION IN X REG
00067A 400E 86 FF     A CURINC LDAA   #$FF       MAKE CURSOR FILL ENTIRE BLOCK
00068A 4010 A7 00     A        STAA   00,X       PUT CURSOR ON SCREEN
00069A 4012 BD F878   A INCH1  JSR    INCH       INPUT FROM TERMINAL
00070A 4015 81 57     A        CMPA   #$57       ASCII W ? FOR WHITE (BUFF)
00071A 4017 27 04 401D         BEQ    WAG        CHOICE WAS EITHER WHITE OR GREEN  DEPENDING ON THE
00072                    *              COLOR SET SELECT PIN
00073A 4019 81 47     A        CMPA   #$47       ASCII G? FOR GREEN
00074A 401B 26 04 4021         BNE    CYAN       NOT WHITE OR GREEN CHECK REST OF COLORS
00075A 401D 86 00     A WAG    LDAA   #$00       WHITE AND GREEN
00076A 401F 20 24 4045         BRA    MAIN       WHITE OR GREEN WAS CHOSEN NOW GO DIPLAY
00077A 4021 81 43     A CYAN   CMPA   #$43       ASCII C? FOR CYAN
00078A 4023 27 04 4029         BEQ    CAY        GO GET CYAN COLOR INFO IN ACC A IF COMPARE IS ZERO
00079A 4025 81 59     A        CMPA   #$59       ASCII Y? YELLOW
00080A 4027 26 04 402D         BNE    MAG        CHECK NEXT COLOR OR INSTRUCTION
00081A 4029 86 01     A CAY    LDAA   #$01       CYAN AND YELLOW
00082A 402B 20 18 4045         BRA    MAIN       C OR Y : NOW GO DISPLAY
00083A 402D 81 4D     A MAG    CMPA   #$4D       ASCII M? MAGENTA
00084A 402F 27 04 4035         BEQ    MAB
00085A 4031 81 42     A        CMPA   #$42       ASCII B? BLUE
00086A 4033 26 06 403B         BNE    ORG
00087A 4035 86 02     A MAB    LDAA   #$02       MAGENTA AND BLUE
00088A 4037 20 0C 4045         BRA    MAIN
00089A 4039 20 D7 4012 INCH4   BRA    INCH1      TO INCH INPUT TERMINAL INFO
00090A 403B 81 4F     A ORG    CMPA   #$4F       ASCII O? ORANGE
00091A 403D 27 04 4043         BEQ    OAR
00092A 403F 81 52     A        CMPA   #$52       ASCII R
00093A 4041 26 04 4047         BNE    START
00094A 4043 86 03     A OAR    LDAA   #$03       ORANGE AND RED INFO FOR VDG
00095A 4045 20 4E 4095 MAIN    BRA    MAIN1      A STEPPING STONE ALONG THE WAY TO KEEP RELATIVE CODING
00096A 4047 81 53     A START  CMPA   #$53       ASCII S ? GO TO BEGINNING OF PROGRAM
00097A 4049 26 06 4051         BNE    NEGOLD
00098A 404B 86 00     A        LDAA   #$00       CLEAR CURSOR
00099A 404D A7 00     A        STAA   00,X       STORE CURSOR
00100A 404F 20 B4 4005         BRA    BEGIN      TO BEGINNING OF PROGRAM
00101A 4051 81 4E     A NEGOLD CMPA   #$4E       ASCII N NEGATIVE DIRECTION AND  LEAVE OLD INFO
00102A 4053 26 0F 4064         BNE    END
00103A 4055 F6 4001   A        LDAB   TEMP5
00104A 4058 E7 00     A        STAB   00,X
00105A 405A 09                 DEX
00106A 405B E6 00     A        LDAB   00,X
00107A 405D F7 4001   A        STAB   TEMP5      UPDATE TEMPORARY REG
00108A 4060 20 AC 400E         BRA    CURINC     GO DISPLAY CURSOR AND GET INPUT
00109A 4062 20 D5 4039 INCH3   BRA    INCH4
00110A 4064 81 45     A END    CMPA   #$45       ASCII E; END PROGRAM
00111A 4066 27 04 406C         BEQ    MONIT      USE SOFTWARE INT IF = E
00112A 4068 81 46     A        CMPA   #$46       ASCII   F; FINISH PROGRAM: SAME  THING
00113A 406A 26 01 406D         BNE    DECLER     CHECK NEXT INSTRUCTION
00114A 406C 3F           MONIT SWI               GO BACK TO MONITOR
00115A 406D 81 44     A DECLER CMPA   #$44       ASCII D DECREMENT ONE LOCATION & CLEAR LAST LOCATION
00117A 406F 26 0F 4080         BNE    INCZ       GO TO NEXT INSTRUCTION
00118A 4071 86 00     A        LDAA   #$00       CLEAR CURSOR

00219A 412E 20 D3 4103 CURINO BRA    CURIN1
00220A 4130 81 03     A CLEAR  CMPA   #$03       EXT OR CONTROL C
00221A 4132 26 D1 4105         BNE    INCH9      IGNORE ERRONEOUS INPUT CHAR
00222A 4134 BD F878   A        JSR    INCH       IF GOOD THEN CHECK HOW MUCH MEMORY TO CLEAR
00223A 4137 81 35     A        CMPA   #$35       INPUT AN ASCII 5 ?
00224A 4139 26 08 4143         BNE    ONEK       NO? GO SEE IF ONE KILO BYTE
00225A 413B CE 01FF   A        LDX    #$01FF     512 BYTES
00226A 413E FF 4003   A        STX    CLRMEM     PUT IN COMPARISON REG
00227A 4141 20 2E 4171         BRA    MEMCLR     GO CLEAR MEMORY
00228A 4143 81 31     A ONEK   CMPA   #$31       INPUT ASCII 1 ?
00229A 4145 26 08 414F         BNE    TWOK       IF NOT CHECK IF WANT 2 KILO BYTES OF MEMORY CLEARED
00230A 4147 CE 03FF   A        LDX    #$03FF     1K BYTES
00231A 414A FF 4003   A        STX    CLRMEM
00232A 414D 20 22 4171         BRA    MEMCLR     GO CLEAR
00233A 414F 81 32     A TWOK   CMPA   #$32       INPUT AN ASCII 2?
00234A 4151 26 08 415B         BNE    THREEK     NO. GO CHECK IF 3 IS INPUT
00235A 4153 CE 07FF   A        LDX    #$07FF     2K BYTES
00236A 4156 FF 4003   A        STX    CLRMEM     STORE IN COMPARISON REG FOR INDEX REG
00237A 4159 20 16 4171         BRA    MEMCLR     GO CLEAR
00239A 415B 81 33     A THREEK CMPA   #$33       INPUT ASCII 3?
00240A 415D 26 08 4167         BNE    SIXK       NO , THEN GO CHECK IF 6K DESIRED TO CLEAR
00241A 415F CE 0BFF   A        LDX    #$0BFF     3K BYTES
00242A 4162 FF 4003   A        STX    CLRMEM
00243A 4165 20 0A 4171         BRA    MEMCLR     GO CLEAR
00244A 4167 81 36     A SIXK   CMPA   #$36       INPUT AN ASCII 6?
00245A 4169 26 9A 4105         BNE    INCH9      NO.  THEN IGNORE ERRONEOUS INPUT AND GO GET NEW COMMAND
00246A 416B CE 17FF   A        LDX    #$17FF     6K BYTES
00247A 416E FF 4003   A        STX    CLRMEM     STORE IN COMPARISON REG
00248A 4171 CE 6000   A MEMCLR LDX    #VIEW      BEGINNING OF SCREEN
00249A 4174 6F 00     A CLEAN  CLR    00,X       CLEAR MEMORY LOCATION
00250A 4176 08                 INX               NEXT LOCATION
00251A 4177 FF 4001   A        STX    TEMP5      SAVE CURRENT LOCATION'S ADDRESS
00252A 417A FE 4003   A        LDX    CLRMEM     GET MEMORY COUNTER, THRU CLEARING?
00253A 417D 26 05 4184         BNE    CONT
00254A 417F FE 4001   A        LDX    TEMP5
00255A 4182 20 AA 412E         BRA    CURINO     YES
00256A 4184 09           CONT  DEX               DECREMENT MEMORY COUNTER
00257A 4185 FF 4003   A        STX    CLRMEM     STORE THE COUNTER NUMBER
00258A 4188 FE 4001   A        LDX    TEMP5      GET CURRENT ADDRESS
00259A 418B 20 E7 4174         BRA    CLEAN      NO
00260                          END
TOTAL ERRORS 00000
```

CROSS REFERENCE TABLE

```
40F6 AGAIN  00191*00197
4005 BEGIN  00064*00100
4029 CAY    00078 00081*
4174 CLEAN  00249*00259
4130 CLEAR  00213 00220*
4003 CLRMEM 00062*00226 00231 00236 00242 00247 00252 00257
4184 CONT   00253 00256*
412E CURINO 00219*00255
4103 CURIN1 00198*00207 00219
40E5 CURIN2 00180*00193 00198
40B5 CURIN3 00150*00180
407E CURIN4 00124*00133 00150
400E CURINC 00067*00108 00124
4021 CYAN   00074 00077*
40E9 DCJMP1 00126 00182*
40EB DECJMP 00182 00186*
406D DECLER 00113 00115*
4064 END    00102 00110*
F878 INCH   00058*00069 00222
4012 INCH1  00069*00089
4093 INCH2  00134*00160
4062 INCH3  00109*00134 00151
4039 INCH4  00089*00109
40B7 INCH7  00151*00181
40E7 INCH8  00181*00199
4105 INCH9  00199*00221 00245
4107 INCOLD 00187 00200*
4080 INCZ   00117 00125*
411F LOLD   00201 00212*
4035 MAB    00084 00087*
402D MAG    00080 00083*
4045 MAIN   00076 00082 00088 00095*
4095 MAIN1  00095 00135*
4171 MEMCLR 00227 00232 00237 00243 00248*
406C MONIT  00111 00114*
4051 NEGOLD 00097 00101*
4043 OAR    00091 00094*
4143 ONEK   00224 00228*
403B ORG    00086 00090*
4112 PLUS   00205*00211
40B9 SHIFT2 00141 00152*
40CA SHIFT4 00139 00161*
40D4 SHIFT6 00137 00168*
4167 SIXK   00240 00244*
4047 START  00093 00096*
40BF STORE  00156*00167 00179
4000 TEMPO  00060*00064 00123 00132 00135 00142 00159
4001 TEMP5  00061*00065 00103 00107 00122 00131 00148 00169 00188 00195 00202 00209 00214 00218 00251 00254 00258
415B THREEK 00234 00239*
414F TWOK   00229 00233*
6000 VIEW   00059*00066 00248
401D WAG    00071 00075*
```