Assembly Instructions for ACE
VDU Board

November 1980

PREPARATIONS

1. Before you begin, make sure you have the proper tools (ie. a low wattage soldering iron with a clean sharp tip, good solder, cutters, etc.).

   This particular board requires good soldering practice to successfully assemble. If you are unsure of your ability, get help from a friend.

2. In case you are unaware, never, never plug in a card with power on!

3. This board can be used as a memory board only. If you are using the board as a RAM board, then follow the instructions up to step 14.

ASSEMBLY

1. Begin by visually checking the board for possible shorts, missed or unaligned drill holes. Repair if required. Carefully examine the back of the board in the area of the memory array. It is a good idea to check all adjacent traces in this area for shorts.

   IMPORTANT! Failure to perform the above step may cause you many hours of grief trying to discover where a short in the board actually is.

2. Install sockets for all CMOS support chips. IC numbers are 1 to 12.

3. If you are going to use the board as a VDU, solder in the socket for the 6847, the MC1372 and the 4508 (IC number 13).

4. If you are going to use the board as a RAM board, solder in 2 sockets for the first bank in position marked 0 on the back of the board (ie. address C000).

5. If you are going to use the board as a VDU board, solder in 2 sockets for the 9th bank in position marked 8 on the back of the board (ie. address E000). Be careful here as you can easily cause a solder bridge.

6. Next solder in the .01 bypass caps. It is a good idea to check the capacitors for shorts with a multimeter before soldering them in.

7. Before inserting any ICs into the sockets, meter the memory array for possible shorts as you did in step 1.

8. Insert IC's numbered 1 to 12 in the appropriate sockets. Check and recheck that you have all IC keys in the proper position! See parts layout for details.

9. Turn off power to CPU. Insert board and then power on. If there are no signs of smoke, then try to address the board with a monitor. You should be able to find a boundary at C000. If not, you must determine what is wrong before inserting RAM chips into their respective sockets.

.../2

Assembly Instructions for ACE VDU Board - continued

10. Power off. Insert 2 2114 RAM chips in either position 0 or position 8. The positions decode as follows:

0	C000 - C3FF
1	C400 - C7FF
2	C800 - CBFF
3	CC00 - CFFF
4	D000 - D3FF
5	D400 - D7FF
6	D800 - DBFF
7	DC00 - DFFF
8	E000 - E3FF
9	E400 - E7FF
A	E800 - EBFF
B	EC00 - EFFF
C	F000 - F3FF
D	F400 - F7FF
E	F800 - FBFF
F	FC00 - FFFF

11. Power on. You should now have 1K of RAM at either C000 - C3FF or E000 - E3FF depending on whether you plugged the chips in position 0 or 8 respectively.

12. You should now run memory diagnostics to insure all the memory decoding and RAM chips are OK. A sample diagnostic program is included with these instructions.

13. The choice to add more memory at this step is up to you. If you are going to use the board as a VDU board, it is best to add the rest of the memory once you get the MC6847 and MC1372 going.

14. This completes the assembly of the board as a memory board. Just add as many 2114 chips and sockets as your budget allows. Do not forget the .01 bypass caps!

15. We now begin the assembly of the VDU support circuitry. Power off and remove board.

16. Solder in the resistors and capacitors as shown in the parts layout. If you can not get a 9-35 pf. variable cap to fit in the space provided, you can get away using a fixed value such as 10 or 15 pf.

17. Solder in the XTAL. Make sure the can of the XTAL does not touch any traces on the board.

18. Next comes the tricky part, the .1 uh coil. You can make your own by winding about 5 turns of number 20 wire on a ¼ inch form. A ¼ inch drill can be used as the form. Spread the turns out to about ½ inch. With the above coil and a 150 pf cap instead of the suggested 180 pf cap, the RF output turned out to be channel 7. If you want a lower channel add more turns to the coil—if you want a higher channel remove turns from the coil.

19. Put the MC1372 in its socket. Power on. If you have a counter, you can get the XTAL on exact frequency by adjusting the 9-35 pf cap. This adjustment does not appear to be very critical.

20. Note that a .01 pf cap will have to be placed between the RF output and the TV antenna input. Run a single wire between the RF output and the TV. Now try to find the RF output on the TV. Check all channels and once you see the "picture" (hash), you are in business.

.../3

Assembly Instructions for ACE VDU Board - continued

21. Now power off and place the MC6847 in its socket. Make sure that you have the keyway correct as this is an expensive chip to burn out! Power on.

22. With your monitor, set address FF00 to 0. This sets the MC6847 into alpha mode. You should now get a square picture on the TV. You can "fine tune" the RF frequency by squeezing or opening up the coil.

23. If all is OK, you can now verify that alpha mode is working.

    At address E000 key in 00
    E001 key in 01
    E002 key in 02
    •
    •
    •
    •
    E03F key in 3F

    alpha 00/02 ↑ light ↑ dark
    add more RAM

24. You can try inverse video by setting the first bit of memory byte, (ie. X'80') on.

    At address E000 key in 80
    E001 key in 81
    E002 key in 82
    •
    •
    •
    •
    E03F key in BF

25. Semi-graphics can be tested by setting the second bit in the byte on. (ie. X '40').

26. Other modes can be tested by changing the memory mapped I/O port at address FF00. See the Motorola spec sheet for details.

TROUBLESHOOTING

1. Getting the memory section of the board going should be fairly straighforward.

2. If you have problems, make sure the address decoding is working. The best way to do this is to use your micro and have it read a constant address, say, E000.

   The program to do this is:

```
F8E0  LDI  #E0
BE    PHI  R14
F800  LDI  #00
AE    PLO  R14
0E    LDN  R14
3000  BR   #00
```

   You should be able to get a $\overline{CS}$ at pin 18 of the 4515 if you are using address E000 as in the above example. If this is good, next check the $\overline{MRD}$, $\overline{MWR}$ circuitry (ie. IC's 1, 10, and 9).

3. If you can not get a $\overline{CS}$ pulse, then go backwards checking IC 5 and 8.

4. If decoding is OK, then verify that the 4066 and 2114 chips are OK. The best way to test 2114 chips is to use the memory diagnostic.

.../4

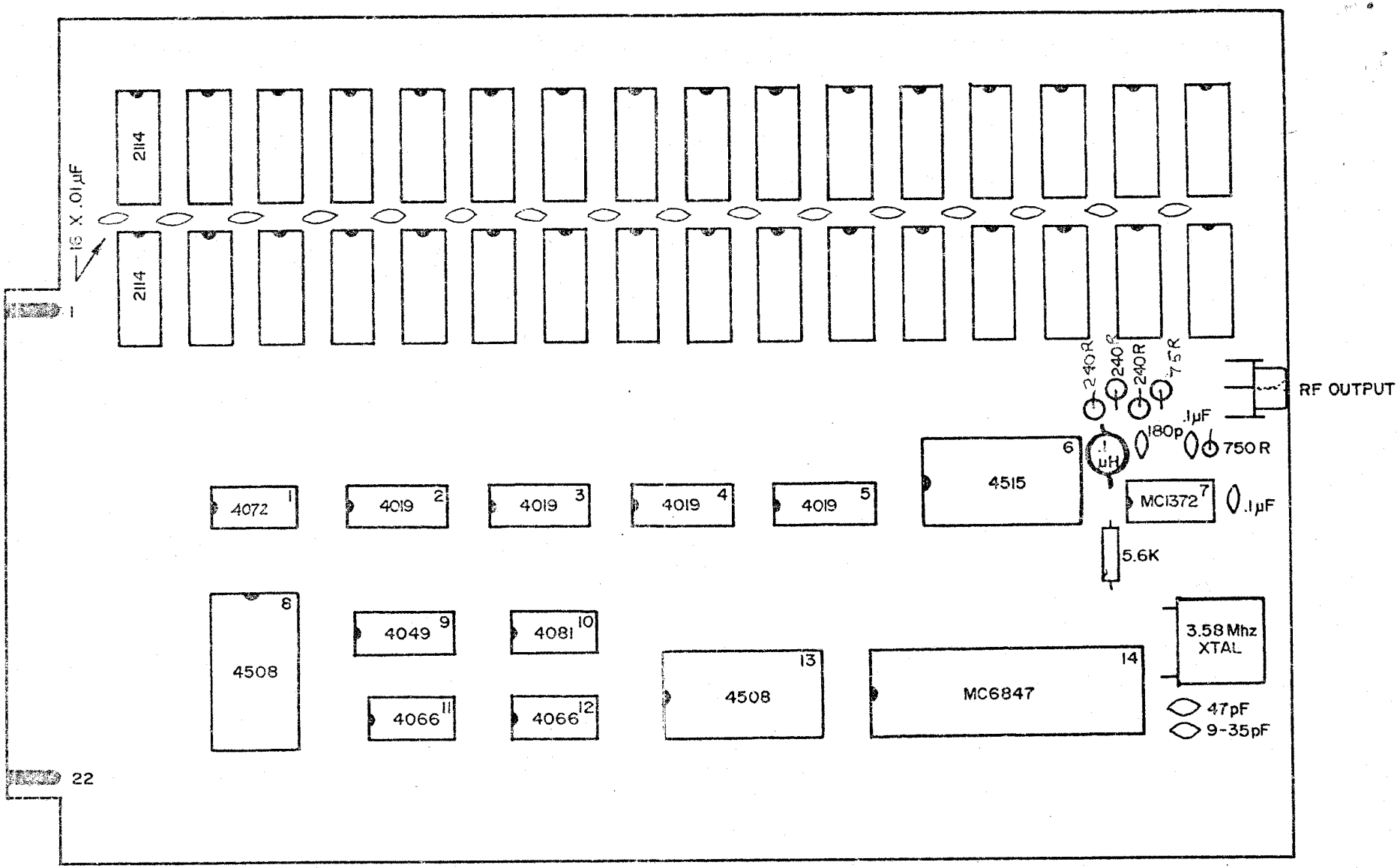
Assembly Instructions for ACE VDU Board - continued

5. Note that if you have an address line shorted to a data line, you can get some weird and wonderful symptoms!

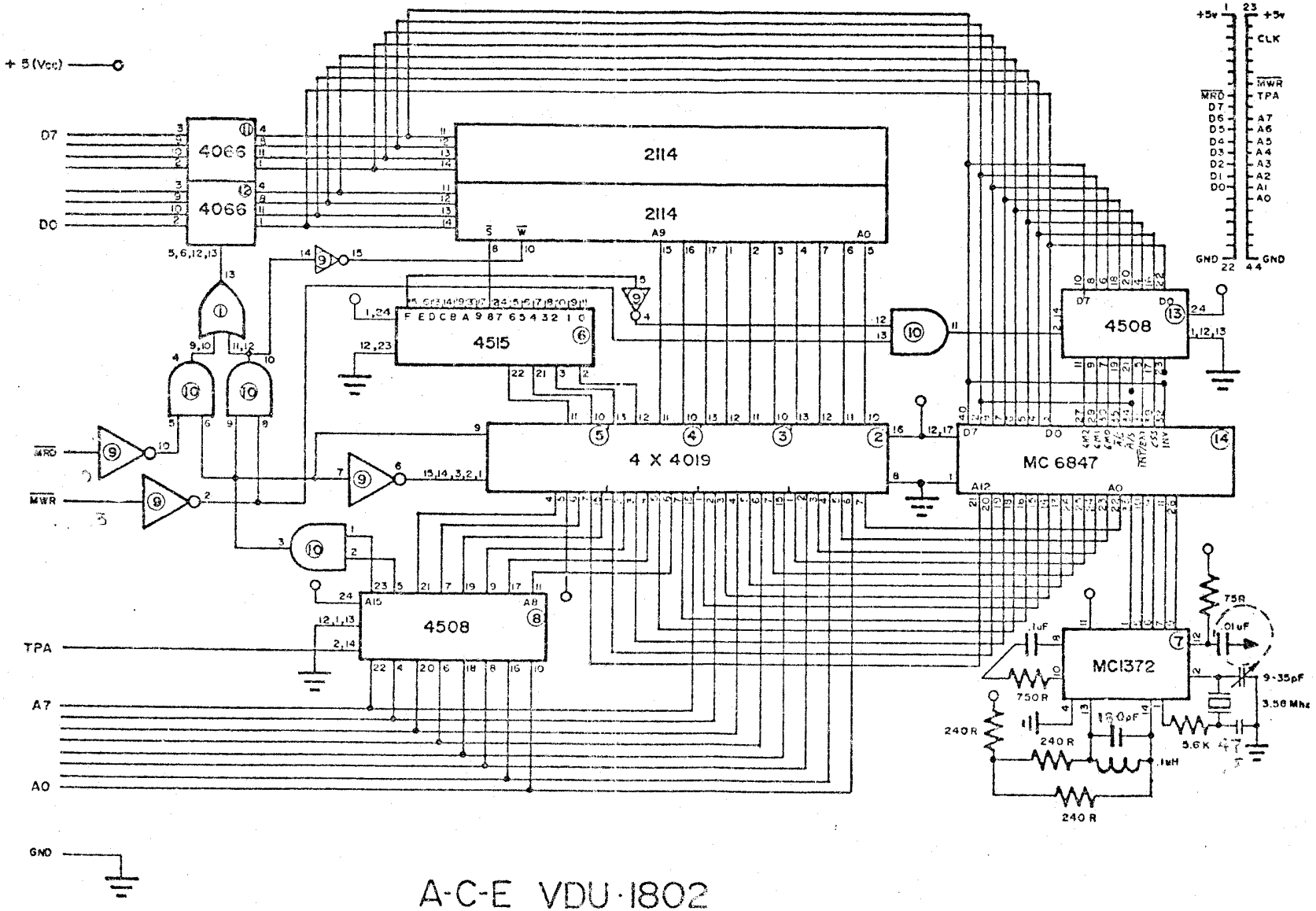
6. If the memory section is working but the VDU is not, then verify that IC 13 is in fact latching your commands. Note that the MC6847 will not function if the MC1372 is not providing a 3.58 MZ clock. You can check this with your logic probe.

7. If you have a scope, you can verify that the MC6847 is generating video by looking at pin 28.

8. Good Luck!



A-C-E VDU 1802 VERSION 2


+ 5 (Vcc)
D7
D0
4066
4066
2114
2114
4515
4 X 4019
4508
4508
MC 6847
MC1372
MRD
MWR
TPA
A7
A0
GND
+5v
CLK
A9
A0
A12
A15
A8
75R
.01uF
.1uF
750R
240R
240R
240R
150pF
.1mH
5.6K
9-35pF
3.58 Mhz

A-C-E VDU·1802

ALPHANUMERIC DISPLAY MODES–All alphanumeric modes occupy an 8 x 12 dot character matrix box and there are 32 x 16 character boxes per TV frame. One of two colors for the lighted dots may be selected by the color set select pin. An internal ROM will generate 64 ASCII display characters in a standard 5 x 7 box. Six bits of the eight-bit data word are used for the ASCII character generator and the two bits not used can be used to implement inverse video or color switching on a character by character basis. A 512 word display memory is required for this class of display.

The ALPHA SEMIGraphics -4 mode translates bits zero through three into a 4 x 6 dot element in the standard 8 x 12 dot box. Three data bits may be used to select one of eight colors for the entire character box. The extra bit is available to implement mode switching on the fly. A 512 word display memory is required. A density of 64 x 32 elements is available in the display area. The element area is four dot-clocks wide by six lines high.

The ALPHA SEMIGraphic -6 mode maps six 4 x 4 dot elements into the standard 8 x 12 dot alphanumeric box, a screen density of 64 x 48 elements is available. Six bits are used to generate this map and two data bits may be used to select one of four colors in the display box. The element area is four dot-clocks wide by four lines high.

FULL GRAPHIC MODE – There are eight full graphic modes available from the VDG. These modes require 1K to 6K bytes of memory. The eight full-graphic modes include an outside color border in one of two colors depending upon the color set select pin (CSS). The CSS pin selects one of two sets of four colors in the four color graphic modes.

The 64 x 64 Color Graphics Mode – The 64 x 64 color graphics mode generates a display matrix of 64 elements wide by 64 elements high. Each element may be one of four colors. A 1K x 8 display memory is required.

The 128 x 64 Graphic Mode – The 128 x 64 graphic mode generates a matrix 128 elements wide by 64 elements high. Each element may be either ON or OFF. However, the entire display may be one of two colors, selected by using the color set select pin. A 1K x 8 display memory is required.

The 128 x 64 Color Graphic Mode – The 128 x 64 color graphics mode generates a display matrix 128 elements wide by 64 elements high. Each element may be one of four colors. A 2K x 8 display memory is required.

The 128 x 96 Graphics Mode – The 128 x 96 graphics mode generates a display matrix 128 elements wide by 96 elements high. Each element is two dot-clocks wide by two lines high. Each element may be either ON or OFF. However, the entire display may be one of two colors selected by using the color select pin. A 2K x 8 display memory is required.

The 128 x 96 Color Graphics Mode – The 128 x 96 color graphics mode generates a display 128 elements wide by 96 elements high. Each element is thus two dot-clocks wide by two lines high. Each element may be one of four colors. A 3K x 8 display memory is required.

The 128 x 192 Graphics Mode – The 128 x 192 graphics mode generates a display matrix 128 elements wide by 192 elements high, making each element two dot-clocks wide by one line high. Each element may be either ON or OFF, but the ON elements may be one of two colors selected with color set select Pin. A 3K x 8 display memory is required.

The 128 x 192 Color Graphics Mode – The 128 x 192 color graphics mode generates a display 128 elements wide by 192 elements high. Each element is two dot-clocks wide by one line high. Each element may be one of four colors. A 6K x 8 display memory is required. A detailed description of the VDG modes is given in Table 3.

The 256 x 192 Graphics Mode – The 256 x 192 graphics mode generates a display 256 elements wide by 192 elements high. Each element is one dot-clock wide by one line high. Each element may be either ON or OFF, but the ON element may be one of two colors selected with the color set select pin. A 6K x 8 display memory is required.

TABLE 1 – TABLE OF MODE CONTROL LINES (INPUTS)

A/G	A/S	INT/EXT	INV	GM2	GM1	GM0	ALPHA/GRAPHIC MODE SELECT
0	0	0	0	X	X	X	Internal Alphanumerics
0	0	0	1	X	X	X	Internal Alphanumerics Inverted
0	0	1	0	X	X	X	External Alphanumerics
0	0	1	1	X	X	X	External Alphanumerics Inverted
0	1	0	X	X	X	X	Semigraphics - 4
0	1	1	X	X	X	X	Semigraphics - 6
1	X	X	X	0	0	0	64 x 64 Color Graphics
1	X	X	X	0	0	1	128 x 64 Graphics
1	X	X	X	0	1	0	128 x 64 Color Graphics
1	X	X	X	0	1	1	128 x 96 Graphics
1	X	X	X	1	0	0	128 x 96 Color Graphics
1	X	X	X	1	0	1	128 x 192 Graphics
1	X	X	X	1	1	0	128 x 192 Color Graphics
1	X	X	X	1	1	1	256 x 192 Graphics

TABLE 2 – SUMMARY OF MAJOR MODES

MAJOR MODE ONE

TABLE OF ALPHA MINOR MODES

Title	Memory	Colors	Display Elements
Alphanumeric (Internal)	512 x 8	2	
Alphanumeric (External)	512 x 8	2	
Alpha Semig-4	512 x 8	8	Box Element
Alpha Semig-6	512 x 8	4	Box Element

MAJOR MODE TWO

TABLE OF MINOR GRAPHICS MODES

Title	Memory	Colors	Comments
64 x 64 Color Graphic	1K x 8	4	Matrix 64 x 64 Elements
128 x 64 Graphics*	1K x 8	2	Matrix 128 elements wide by 64 elements high
128 x 64 Color Graphic	2K x 8	4	
128 x 96 Graphics*	2K x 8	2	Matrix 128 elements wide by 96 elements high
128 x 96 Color Graphic	3K x 8	4	
128 x 192 Graphics*	3K x 8	2	Matrix 128 elements wide by 192 elements high
128 x 192 Color Graphic	6K x 8	4	
256 x 192 Graphics*	6K x 8	2	Matrix 256 elements wide by 192 elements high

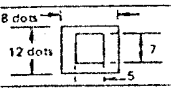
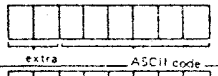
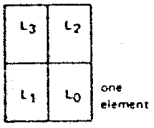
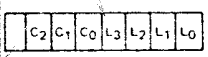
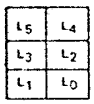
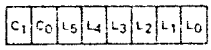
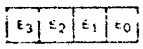
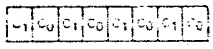
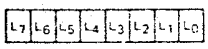
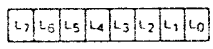
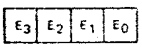
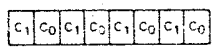
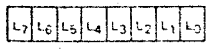
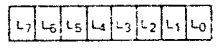
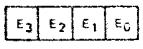
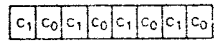
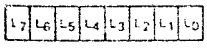
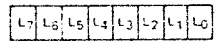
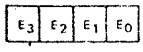
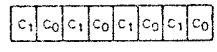
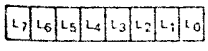
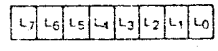
*Graphics mode turns on or off each element. The color may be one of two.

TABLE 3 – DETAILED DESCRIPTION OF VDG MODES

VDG PINS									COLOR			TV SCREEN
$\overline{MS}$	$A/\overline{G}$	$A/\overline{S}$	INT/EXT	GM2	GM1	GM0	CSS	INV	Character Color	Background	Border	Display Mode
1	0	0	0	X	X	X	0	0	Green	Black	Black	32 Characters in columns
								1	Black	Green		16 Characters in rows
							1	0	Orange	Black	Black	
								1	Black	Orange		
1	0	0	1	X	X	X	0	0	Green	Black	Black	32 Characters in columns
								1	Black	Green		16 Characters in rows
							1	0	Orange	Black	Black	
								1	Black	Orange		
1	0	1	0	X	X	X	X	X	Lx C2 C1 C0	Color	Black	64 Display elements in columns
									0 X X X	Black		32 Display elements in rows
									1 0 0 0	Green		
									1 0 0 1	Yellow		
									1 0 1 0	Blue		
									1 0 1 1	Red		
									1 1 0 0	Buff		
									1 1 0 1	Cyan		
									1 1 1 0	Magenta		
									1 1 1 1	Orange		
1	0	1	1	X	X	X	0	X	Lx C1 C0	Color	Black	64 Display elements in columns
									0 X X	Black		48 Display elements in rows
							1		1 0 0	Green		
									1 0 1	Yellow		
									1 1 0	Blue		
									1 1 1	Red		
									0 X X	Black		
									1 0 0	Buff		
									1 0 1	Cyan		
									1 1 0	Magenta		
									1 1 1	Orange		
1	1	X	X	0	0	0	0	X	C1 C0	Color	Green	64 Display elements in columns
									0 0	Green		
									0 1	Yellow	Buff	64 Display elements in rows
									1 0	Blue		
							1		1 1	Red		
									0 0	Buff		
									0 1	Cyan		
									1 0	Magenta		
									1 1	Orange		
1	1	X	X	0	0	1	0	X	Lx	Color	Green	128 Display elements in columns
									0	Black		
									1	Green		64 Display elements in rows
							1		0	Black	Buff	
									1	Buff		
1	1	X	X	0	1	0	0	X	Same color as Graphics one C		Green	128 Display elements in columns
							1				Buff	64 Display elements in rows
1	1	X	X	0	1	1	0	X	Same color as Graphics one R		Green	128 Display elements in columns
							1				Buff	96 Display elements in rows
1	1	X	X	1	0	0	0	X	Same color as Graphics one C		Green	128 Display elements in columns
							1				Buff	96 Display elements in rows
1	1	X	X	1	0	1	0	X	Same color as Graphics one R		Green	128 Display elements in columns
							1				Buff	192 Display elements in rows
1	1	X	X	1	1	0	0	X	Same color as Graphics one C		Green	128 Display elements in columns
							1				Buff	192 Display elements in rows
1	1	X	X	1	1	1	0	X	Same color as Graphics one R		Green	256 Display elements in columns
							1				Buff	192 Display elements in rows

D4 - D2 D7 D6 D5 D1 -

TABLE 3 – DETAILED DESCRIPTION OF VDG MODES

TV SCREEN Detail	VDG DATA BUS	COMMENTS
		The ALPHA-NUMERIC INTERNAL mode uses an internal character generator which contains the following five dot by seven dot characters. @ABCDEFGHIJK LMNOPQRSTUVWXYZ ↑ [ ] ↑ _ SP, ! $ % & ' ( ) * + , - . / 0123456789 : ; < = > ? The six bit ASSCII code leaves two bits free and these may be externally connected to the mode pins (A/G, A/S, INT/EXT, GM2, GM1, GM0, CSS or INV)
		The ALPHA-NUMERIC 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.
		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 VDC data bus. The color of illuminated parts is determined by three bits.
		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.
		The GRAPHICS ONE C mode uses a maximum of 1024 bytes of display RAM in which one pair of bits specifies one picture element.
		The GRAPHICS ONE R mode uses a maximum of 1024 bytes of display RAM in which one bit specifies one picture element.
		The GRAPHICS TWO C mode uses a maximum of 2048 bytes of display RAM in which one pair of bits specifies one picture element.
		The GRAPHICS TWO R mode uses a maximum of 1536 bytes of display RAM in which one bit specifies one picture element.
		The GRAPHICS THREE C mode uses a maximum of 3072 bytes of display RAM in which one pair of bytes specifies one picture element.
		The GRAPHICS THREE R mode uses a maximum of 3072 bytes of display RAM in which one bit specifies one picture element.
		The GRAPHICS SIX C mode uses a maximum of 6144 bytes of display RAM in which one pair of bits specifies one picture element.
		The GRAPHICS SIX R mode uses a maximum of 6144 bytes of display RAM in which one bit specifies one picture element.

MEMORY TEST ROUTINE FOR THE TEC 1802

- BY A. DUNLOP, I.F. #4, p. 20

A recent algorithm by Knaizuk and Hartmann (IEEE Transactions on Computers, April 1977) outlines an ultra-fast RAM test. This algorithm is employed in the test routine shown in Listing 1.

The memory test routine will detect any stuck-at-1 or stuck-at-0 fault in a RAM, including the memory itself, the address and data lines, and the address decoders. It sacrifices testing efficiency to a small degree so as to achieve simplicity. run time is short compared to some other test routines.

Operation of the routine is described in three steps processed for three phases with complemented data used in alternate passes. Each phase is made up of the following three steps:

1. A Data Byte is stored in every location to be tested
2. The complement of the Data Byte is stored in every third location
3. Memory is checked for all locations in the test region

It is important to note that the above three steps must be done as three seperate iterations.

The three steps are performed three times, once for each phase, with the position of the complemented Data Byte changing each time. Then the Data Byte is complemented and the next pass is begun.

The Data Byte is initially #FF giving an FF FF 00 FF FF 00 FF... pattern for the first phase and FF 00 FF FF 00 FF FF... for the second phase and 00 FF FF 00 FF FF 00... for the third phase of the first pass. The second pass first phase generates a 00 00 FF 00 00 FF... pattern.

Memory is tested as a group of bytes between, and including, any two arbitrary addresses allowing small sections of memory to be tested. The low address value of the area to be tested is placed in bytes 2 and 3 of the program at the label 'BEGIN' and the high address limit is placed in bytes 4 and 5 at the label 'END'. Testing progresses from high to low address so as to simplify the limit-checking. The program starts at byte 0 with a branch around the data area to the initialization section.

After 256 complete passes the program will halt unconditionally. The user may wish to NOP the halt instruction to let the test run continuously.

If a fault is detected, the program will halt conditionally with the most significant byte of the fault address displayed in the Data leds, and the Q led indicating what was expected to be in the memory byte at the fault address: Q reset when #00 was expected and Q set when #FF was expected. Pressing and holding the '1' key will permit the least significant byte of the fault address to be displayed. Upon releasing the '1' key the program will continue.

The test routine can be executed in any page of memory since it references the absolute page address at initialization time.

This test routine will not solve memory problems or indicate the cause directly. It serves only to direct the user's attention to specific problem areas if they exist.

```
                                .TITLE  --- MEMORY TEST ROUTINE ---
                                        ;MEMORY TEST   RCA 1802   JAN. 1978 AD
                                        ;REGISTER ASSIGMENT
                                        ;R0     P COUNTER
                                        ;R1     PRESENT BYTE ADDRESS IN MEMORY
                                        ;R2.1   PHASE INDEX (3 PHASES TO TEST)
                                        ;R2.0   INDEX REFLECTING PHASE INDEX
                                        ;R3     ADDRESS OF MEMORY CONTAINING BEGIN
                                        ;R4     ADDRESS OF MEMORY CONTAINING END
                                        ;R5     ADDRESS OF TEMPORARY BYTE
                                        ;R6     NUMBER OF COMPLETE PASSES
                                        ;R7.0   DATA BYTE
                                        ;
                                        ;** CAUTION ** RANGE FOR 'BEGIN'  1 - #FFFF
                                        ;
0000  30 07           START:  BR      INZ     ;
0002  FF FF           BEGIN:  .DBYTE  #FFFF   ;BEGINNING ADDRESS TO BE TESTED
0004  FF FF           END:    .DBYTE  #FFFF   ;ENDING ADDRESS TO BE TESTED
0006  00              TEMP:   .BYTE   0       ;TEMPORARY LOCATION FOR OUTPUT
0007  90              INZ:    GHI     R0      ;INITIALIZE HIGH REG TO
0008  B3                      PHI     R3      ;                     THIS PAGE ADDRESS
0009  B4                      PHI     R4      ;
000A  B5                      PHI     R5      ;
000B  F8 02                   LDI     BEGIN   ;
000D  A3                      PLO     R3      ;R3 SET TO ADDRESS OF BEGIN
000E  F8 04                   LDI     END     ;
0010  A4                      PLO     R4      ;R4 SET TO ADDRESS OF END
0011  F8 06                   LDI     TEMP    ;
0013  A5                      PLO     R5      ;R5 SET TO ADDRESS OF TEMP
0014  F8 00                   LDI     0       ;
0016  B6                      PHI     R6      ;
0017  A6                      PLO     R6      ;R6 SET TO #0000
0018  A7              P000:   PLO     R7      ;DATA BYTE INITIALIZED AS #00
0019  F8 02                   LDI     2       ;SET PHASE INDEX FOR 3 PHASES
001B  B2              P010:   PHI     R2      ;
001C  7A                      REQ             ;RESET Q FOR STEP 1 -  EVERY BYTE
001D  A2              P020:   PLO     R2      ;R2.0 SET FOR "FUNNY SUBTRACT"
001E  44                      LDA     R4      ;
001F  B1                      PHI     R1      ;R1 PRESENT TEST BYTE START AT END
0020  04                      LDN     R4      ;
0021  A1                      PLO     R1      ;
0022  24                      DEC     R4      ;RESET R4 TO POINT TO END AGAIN
0023  39 2C                   BNQ     P040    ;STEP 1 BYPASS
0025  82              P030:   GLO     R2      ;SPECIAL ACTION FOR STEP 2
0026  32 2C                   BZ      P040    ;SUBTRACTION COMPLETE
0028  22                      DEC     R2      ;
0029  21                      DEC     R1      ;"FUNNY SUNTRACT" R1=R1-R2.0
002A  30 25                   BR      P030    ;
002C  13              P040:   INC     R3      ;TO GET LEAST SIGNIFICANT BYTE
002D  81                      GLO     R1      ;                       OF 'BEGIN'
002E  E3                      SEX     R3      ;SET X FOR SUBTRACT 'BEGIN'
002F  F7                      SM              ;
0030  23                      DEC     R3      ;TO GET MOST SIGNIFICANT BYTE
0031  91                      GHI     R1      ;                       OF 'BEGIN'
0032  77                      SMB             ;PRESENT ADDRESS  -  BEGIN
0033  3B 43                   BM      P060    ;DONE THIS STEP WHEN DF=0
0035  87                      GLO     R7      ;GET DATA BYTE
0036  FB FF                   XRI     #FF     ;COMPLEMENT IT
0038  39 3F                   BNQ     P050    ;STEP 1 BYPASS
003A  87                      GLO     R7      ;SPECIAL ACTION FOR STEP 2
003B  51                      STR     R1      ;STORE DATA BYTE
003C  21                      DEC     R1      ;DECREMENT PRESENT ADDRESS TWICE TO
003D  21                      DEC     R1      ;BYPASS NEXT TWO BYTE ADDRESSES
003E  38                      SKP             ;
003F  51              P050:   STR     R1      ;STORE  COMPLEMENTED DATA BYTE
0040  21                      DEC     R1      ;DECREMENT PRESENT ADDRESS
0041  30 2C                   BR      P040    ;
0043  31 49           P060:   BQ      P100    ;FINISHED STEP 2
0045  7B                      SEQ             ;SET Q FOR STEP 2 - EVERY THIRD BYTE
0046  92                      GHI     R2      ;GET PHASE INDEX
0047  30 1D                   BR      P020    ;GO TO STEP 2 ENTRY
0049  44              P100:   LDA     R4      ;STEP 3 TEST FOR CORRECT DATA
004A  B1                      PHI     R1      ;
004B  04                      LDN     R4      ;
004C  A1                      PLO     R1      ;R1 SET TO END ADDRESS
004D  24                      DEC     R4      ;RESET R4 TO POINT TO END AGAIN
004E  92                      GHI     R2      ;GET PHASE INDEX
004F  A2                      PLO     R2      ;SET FOR INDEX TO EVERY THIRD BYTE
0050  13              P110:   INC     R3      ;                       FOR THIS PHASE
0051  81                      GLO     R1      ;
0052  E3                      SEX     R3      ;SET X FOR SUBTRACT 'BEGIN'
0053  F7                      SM              ;
0054  23                      DEC     R3      ;
0055  91                      GHI     R1      ;
0056  77                      SMB             ;PRESENT ADDRESS  -  BEGIN
0057  3B 7E                   BM      P200    ;DONE STEP 3 WHEN DF=0
0059  82                      GLO     R2      ;
005A  32 61                   BZ      P120    ;BRANCH TO TEST THIRD BYTE
005C  87                      GLO     R7      ;
005D  FB FF                   XRI     #FF     ;COMPLEMENT DATA BYTE
005F  30 65                   BR      P130    ;
0061  F8 03           P120:   LDI     3       ;
0063  A2                      PLO     R2      ;RESET R2.0 TO TEST THIRD BYTE
0064  87                      GLO     R7      ;                       FROM HERE
0065  CE              P130:   LSZ             ;
0066  7B                      SEQ             ;SET  Q  IF D=#FF
0067  38                      SKP             ;
0068  7A                      REQ             ;RESET Q  IF D=#00
0069  E1                      SEX     R1      ;SET X FOR EXCLUSIVE OR
006A  F3                      XOR             ;COMPARE WITH MEMORY BYTE
006B  32 7A                   BZ      P140    ;OK
006D  E5                      SEX     R5      ;SET X FOR OUTPUT FAULT ADDRESS
006E  91                      GHI     R1      ;
006F  55                      STR     R5      ;
0070  64                      OUT     4       ;OUTPUT MOST SIGNIFICANT BYTE OF
0071  25                      DEC     R5      ;                       FAULT ADDRESS
0072  3F 72                   BN4     @       ;WAIT FOR 'I' PUSHED
0074  81                      GLO     R1      ;
```

```
 1  0075  55                    STR     R5      ;
 2  0076  64                    OUT     4       ;OUTPUT LEAST SIGNIFICANT BYTE OF
 3  0077  25                    DEC     R5      ;                       FAULT ADDRESS
 4  0078  37 78                 B4      @       ;WAIT FOR 'I' RELEASED
 5  007A  22            P140:   DEC     R2      ;R2.0=R2.0-1
 6  007B  21                    DEC     R1      ;R1=R1-1
 7  007C  30 50                 BR      P110    ;
 8  007E  92            P200:   GHI     R2      ;
 9  007F  A2                    PLO     R2      ;
10  0080  22                    DEC     R2      ;DECREMENT PHASE INDEX
11  0081  3A 8C                 BNZ     P210    ;PHASE INDEX NOT -1, DO NEXT PHASE
12  0083  16                    INC     R6      ;INCREMENT COUNT OF PASSES
13  0084  96                    GHI     R6      ;
14  0085  3A 85                 BNZ     @       ;STOP AFTER COMPLETING 256 PASSES
15  0087  87                    GLO     R7      ;
16  0088  FB FF                 XRI     #FF     ;COMPLEMENT DATA BYTE FOR NEXT PASS
17  008A  30 18                 BR      P000    ;GO DO ANOTHER PASS
18  008C  82            P210:   GLO     R2      ;GET DECREMENTED PHASE INDEX
19  008D  30 18                 BR      P010    ;GO DO NEXT PHASE
20                              .END            ;
```