THE MICRO WORKS

DIGISECTOR
DS-65
OWNER'S MANUAL

P.O. BOX 1110 DEL MAR, CA 92014 714-942-2400

DS-65 DIGISECTOR

TABLE OF CONTENTS

INTRODUCTION

Thank you for purchasing a Micro Works Digisector. Every effort has been made in the development of the DS-65 to provide you with a long lasting, trouble free computer accessory. We suggest that you read this manual thoroughly before installing the DS-65.

The Micro Works DS-65 Digisector is a video digitizer which accepts input from a video source such as a closed circuit television camera, converts the analog video signal to digital data, and transfers this data to the computer under software control. The DS-65 can resolve 256 X 256 picture elements (pixels) and provide up to 64 levels of grey scale. Conversion time for 6 bits is approximately 12 µsec. The DS-65 accepts either standard (NTSC) or industrial input from the video source. Its output to the system monitor is the video plus an intensified cursor located at the point where the Digisector is currently looking.

This intelligent Apple peripheral has on-board software in 2708 EPROM, featuring:

- --Full screen scans directly to Apple Hi-Res screen
- --Easy random-access digitizing by Basic programs
- --Line-scan digitizing for reading charts or tracking objects
- --Utility functions for clearing and copying the Hi-Res screen

The Micro Works is certain you will find the DS-65 a true enhancement to your computer system. We look forward to hearing any suggestions or comments from our customers and are interested in the various applications to which the Digisector will be put.

UNPACKING AND INSTALLATION

Carefully remove the DS-65 from the box and unwrap the packing material. Take time to inspect the PC board for any damage which may have been incurred in shipping. If there is any damage, save all packing materials and notify the carrier immediately.

Your DS-65 contains MOS integrated circuitry which may easily be damaged by static electricity. Avoid overhandling and do not allow anything to come into contact with the conductors on the board. Never lift the board out of, or plug it into, a computer which is turned on.

GETTING STARTED

Your DS-65 Digisector is an intelligent Apple peripheral card which plugs into one of the Apple's I/O slots. By "intelligent" it is meant that the card requires no complex software to make it work; the software for its basic functions is in a read-only memory right on the card.

To give your DS-65 a basic checkout without a camera, plug it into slot 3 of your Apple (with the power off, of course) and, after turning on the power and going into Basic, type in the following program:

```
10  PR#3
20  PRINT "!"
30  END
```

Run the program. The screen goes blank; congratulations. The Digisector board's code for "Clear Hi-res Screen and Display It" is an exclamation point. The statement "PR#3" means that the DS-65 is in slot 3; it tells the Apple to send all PRINT statements to slot 3 instead of to the screen. This is how all BASIC programs will communicate with the DS-65.

Now, to take a picture, you will need to hook up a camera, as described in the following section.

HOOKING UP A CAMERA

The camera sold with the DS-65 is the Advanced Video FS-II, and is described in the next paragraph. If you have your own camera or are using another video source, connect it to the Video In phono plug (fig. 1) coming from the DS-65 and skip the remainder of this section.

The FS-II camera has two parts, one being the power supply box and the other the camera itself. The camera is connected to the power supply by means of the DIN cable which screws onto the camera and plugs into the power supply. The DS-65 Video In plug plugs into the Video Out jack on the power supply (See fig. 1). Put the camera on a tripod or stable surface and make sure that the cord is not where it can be tripped over.

The lens cap on the front of the camera slides upward, and the camera is focused by turning the outermost lens (the inner lever on the lens is a zoom). Turn on the camera with the slide switch on the power supply, which should light the red light on the top of the camera.

The camera should be used with sufficient light, preferably with photo floodlights or outdoors. Pictures can be taken with ordinary room light but it is harder to obtain good picture quality. The "bright/overcast" switch on the side of the camera should be used to decrease the lens opening in bright sunlight. Do not leave the camera pointing at bright light sources.

The foregoing has been a brief description of using the FS-II with the DS-65, but is by no means a complete description of the camera. The manual which comes with the camera should be reviewed for additional information and cautions.

HOOKING UP A MONITOR

A "monitor" is a TV set or video monitor which will enable you to see the picture as it comes from the camera. You already have one monitor on your Apple, which is the screen on which your Apple displays its text and graphics. That is also the screen on which the digitized pictures will be displayed. For focusing and composing pictures, it is recommended that you have another monitor hooked up to the Video Out jack of the DS-65 (See fig. 2).

It is not necessary that you have this second monitor. The DS-65 will digitize pictures without it and they will be displayed to the regular Hi-res screen. However, as it is extremely helpful to have a display of the picture before it is digitized (and in some applications necessary), we recommend that another monitor be used. If not, the Apple monitor may be unplugged from the Apple and used to peek at the DS-65 video output when desired, or a switch may be used to switch between the two.

The Video Out jack of the DS-65 must be "terminated" for proper functioning of the Digisector; that is, it must be connected to something. If it is not connected to a monitor, it should have a terminating resistor plugged into it. It is shipped with such a device plugged into it, which should be unplugged in order to plug a monitor into it. Running the DS-65 without termination will not damage it, but will cause picture scans to be slow and hesitant.

The picture which appears on the monitor is the camera output superimposed with a "cursor" - a white dot which appears when a point is being digitized. When points are being scanned quickly, as when taking a picture, the cursor appears to be a vertical line and will show you where the scan has reached in the picture.

Now, hook up your monitor (either your usual one or a

new one) to the DS-65 Video Out jack (first removing the terminating plug). Hook up the camera, and turn on the camera, monitor, and the Apple. A picture should appear on the screen. If it doesn't, then check the following:

1 - The camera isn't warmed up yet. Give it a minute.
2 - Lens cap down on camera.
3 - Video plug on DS-65 on backwards. See fig. 3.
4 - Something's hooked up wrong. See fig. 2.

TAKING A PICTURE

If the Apple's monitor was moved to the DS-65 output in the last step, put it back on the Apple output now and replace the terminator plug. Type in this program:

```
10  PR#3
20  PRINT "!*"
30  GOTO 20
```

This program differs from the one in section A by the inclusion of the asterisk, which is the DS-65 code for "Take a Picture". Also the GOTO 20 has been added which will cause a series of pictures to be taken.

This program assumes the DS-65 is in slot 3. If it is in another slot, change line 10; e.g., PR#5 for slot 5, etc. Don't use slot zero.

Run the program. A series of pictures should appear. If it is unrecognizable at first, don't panic; the brightness and contrast controls should be adjusted to obtain a satisfactory picture (see section F below).
Make sure the camera is pointing at a well-lit subject, for sufficient light is a very important factor in picture quality.

COMMANDS FROM BASIC

To operate the DS-65 from Basic, the PRINT statement must be assigned to the DS-65 slot. This is done with the PR statement. For example, if the DS-65 has been placed in slot 2, then the statement PR#2 will cause all subsequent PRINT statements to be directed not to the screen but to the DS-65. The statement PR#0 will return print statements to their normal function, and this should be done before the end of all programs which use the DS-65. Because PR#0 returns print statements to normal, the DS-65 cannot be used in slot zero.

Once the print output is assigned to the slot in which your DS-65 resides, the following special print statements can be used.

`PRINT "!"`	Clears Hi-res Page 1 and displays it.
`PRINT "*"`	Takes a picture and puts it into Hi-res Page 1. (Page 1 should be cleared first.)
`PRINT "="`	Copies Hi-res Page 1 into Page 2.

`POKE 64,X`
`POKE 65,Y`
`PRINT "."`
`B=PEEK(66)`
	Finds the brightness of one point (X,Y). B will be zero for very black through 63 for very white. Points (0,0) and (255,255) should not be used due to the nature of the Digisector.

`POKE 64,X`
`POKE 65,B`
`PRINT "<"`
`Y=PEEK(66)`
	Finds the highest point on the vertical line at X which has a brightness less than B. This function can be used for reading graphs, tracking objects, etc.

`PRINT "!*"`	Note that multiple commands may be sent in one print statement. This particular combination means "clear screen and take picture" and is commonly used.

BRIGHTNESS, CONTRAST, AND WIDTH

On the top of the DS-65 there are three small knobs which are marked B, C, and W. These are controls for Brightness, Contrast, and Width. The Brightness control makes the picture as seen by the Apple brighter or darker; when the digitized image is placed on the Hi-res there will be a greater or lesser number of dots on according to the setting of the control. The contrast affects how much gray area there is in the picture; with the contrast set very high everything tends to be either black or white, while low contrast makes everything shades of gray. These controls should be set for the best picture in your application, as the optimum setting depends on your lighting, subject matter, camera, and type of result wanted.

The Width knob controls the width of the area scanned by the DS-65. When it is at its narrowest setting, the cursor on the monitor will stop noticably short of the right-hand edge. This control is used to adjust the aspect ratio of the resulting picture; that is, how square it is. If digitized pictures look tall and thin, or short and fat, then the width control should be adjusted.

THE PAGE SELECT JUMPER

There are two versions of the Digisector program, both of which are in the ROM on your DS-65. They differ only in the method used to take pictures under the PRINT "*" command. One method uses eight seconds to complete the scan, and scans all 256 points across the video screen; the other method takes only two seconds and takes every other point vertically and horizontally, resulting in a slightly rougher picture. Which version you will want to use will depend on your applications.

The two versions are selected by means of a jumper located just over the ROM on the DS-65 (See fig. 4). It is marked "P S" for page select. When it is cut (using a knife or scissors point) the scan will take two seconds, and when not cut (or jumpered again with a small piece of wire) the scan will take eight seconds.

HOW IT WORKS

Sync Stripper and Video Amplifier

Composite video appearing at the input is amplified to 2 volts peak-to-peak by U22. DC restoration clamps the sync tips to ground through Q1. This clamp pulse, appearing at the collector of Q1, is sliced by Q2 and applied to sync stripper logic U11 and U17. This provides separated horizontal and vertical sync pulses, $\overline{\text{VSYNC}}$, HBLANK, and $\overline{\text{HBLANK}}$. These pulses control the horizontal and vertical position counters and will be discussed later.

Analog to Digital Converter

The amplified video is connected to a sample and hold circuit consisting of Q3, C5 and buffer amplifier U21. When the sample line is pulsed high, Q3 conducts, charging C5 up to the value of of video signal at that instant. Since amplifier U21 has FET inputs, leakage current is low and the cap remembers this voltage level during the subsequent A-D conversion.

U15, U16 and U20 comprise a high speed successive approximation A-D converter. Most of the smarts in this converter are contained in the successive approximation register (SAR) U15, which controls the converstion sequence. Conversion is accomplished as follows: R18 sets a DC level of about .6 volt at pin 3 of comparator U20. The signal level to be digitized is present at the output of buffer amplifier U21. The SAR controls a binary weighted current output D-A converter, U16; all bits are initially off.

The SAR first turns on the most significant bit of this D-A converter, causing 1 ma. of current to flow through R17 and R20. If this current is sufficient to drop the voltage at pin 2 of the comparator to below the threshold voltage at pin 3, the comparator output goes high, telling the SAR that that's too much current. In that case the SAR turns the MSB back off and turns on the next most significant bit (.5 ma.). This process continues for all 6 bits; if the comparator output stays low after a given bit is turned on, the SAR leaves that bit on. If it goes high, the SAR turns it off again before proceeding to the next most significant bit.

In this way, the SAR sneaks up on the correct value of current to supply to the summing junction at pin 2 of U20 so that when the conversion is done, pin 2 is within a few millivolts of the threshold voltage at pin 3, and the binary weighted bits that supplied the current necessary to "null out" the input voltage are present at the outputs of the SAR. This binary number is the digitized value of the brightness level at the sample point. The brightness adjustment, R19, is set to make the "0" brightness level equal to the blanking, or black level of the incoming video. The contrast adjustment, R17, adjusts the gain of the A-D converter. Since conversion is now complete, the data is ORed onto the A side of the PIA for reading by the computer.

Position Counters

U8, U9 and U13 comprise a self latching counter with a maximum count of 255; that is, when started, it counts up to 255 and then quits until it is reset again. This counter is reset by the occurrence of the vertical sync pulse (60 Hz.) and incremented by the horizontal sync pulses (15,750 Hz.) and as such, contains the vertical position (in scan lines) of the scanning spot at all times. This is our "Y" coordinate. U6, U7 and U13 make up a similar counter, but this one is reset by the horizontal sync and incremented by the high speed dot clock composed of U11 and gates in U12. The frequency of this dot clock determines the width of the picture scanned and is adjustable by R16. This counter contains our "X" coordinate.

Control Circuitry

We now have two counters which keep track of the X and Y coordinates of the scanning spot in the incoming video at all times, and a means of digitizing the brightness of any individual spot. Getting them together is easy. The outputs of the X and Y position counters are connected to one set of inputs of a 16 bit digital comparator, U2-U5. The other inputs to this comparator are connected to the two 8 bit ports of the PIA chip. When the scanning spot coordinates (counter outputs) are equal to the desired X-Y coordinates (PIA A and B side outputs), a 200 ns. wide EQU pulse is produced by the comparator. This pulse signifies that the dot is in exactly the spot we wish to digitize, so we use it to provide the sample pulse to the sample and hold circuit, capturing the brightness of the spot we want. This pulse also triggers the SAR to begin its A-D conversin, as described before, and is summed with the video output to provide a cursor which shows where the Digisector is looking. The PIA's control lines CB2 and CA1 complete the handshaking necessary for Digisector operation.

Programming Information

Two features of the versatile 6821 PIA make operation of the Digisector extremely simple. The first is the PIA's ability to generate a strobe pulse on the CB2 line immediately following a write operation to the B side data register (CONTROL REGISTER B = 2C hex). This means that all one needs to do to start the Digisector's search is to write a Y coordinate to the PIA's B side; the Digisector start pulse issues automatically. (When the Digisector is done, it sets the high order bit of control register A through the CA1 line.) The other feature of interest is in the structure of the outputs of the A side. These outputs are effectively open drain outputs with a passive pull up to +5. When the A side is configured as an output, the output FETS in the PIA turn on wherever a Ø bit is written to the data register. When this register is read by the CPU, the bits will be read as low or Ø if the output side FET is on or if an exernal connection is holding the line low. Hence we may use the A side of the PIA as a bi-directional register without changing the control register contents at all--simply writing FF hex to the A side will allow external open

collector wired OR data to be read correctly from the A side. This trick saves an additional 8 bit port and/or a lot of CPU time in passing address (X coordinate) and data (brightness) bi-directionally through the PIA. The brightness data is ORed onto the PIA by U10 whenever its supply voltage is turned on through Q4. Whenever the X coordinate data must be output to the comparators, Q4 is turned off by the control logic and U10 is effectively disconnected from the A lines.

These tricks are used to minimize the coding required to operate the Digisector, as its speed is generally software limited when sampling blocks of video data (such as the portrait program provided). When randomly addressing the video, remember that a given point on the raster occurs only once every 16.66 ms. and plan digitizing algorithms accordingly. For example, if the following points are desired:

(1,1), (11,11), (21,21), (31,31), (41,41), (51,51)

they should be acquired in ascending order as written. Time for acquisition would be (8 ms. average latency) + 5 ms. = 13 ms. If they were to be acquired in the reverse order, acquisition would take (8 ms. average latency) + 5 X 16 ms. = 88 ms. Such is the nature of raster scan devices. Remember that the upper left hand corner of the screen is (1,1) and the lower right is (FE,FE).

Due to the latching nature of the X and Y position counters, the addresses 0,0 hex and FF,FF hex are illegal and will produce undefined results, since these states occur for much longer times than one picture element (200 ns.).

We have found that proper lighting is the most important factor in obtaining quality images using the DS-65.

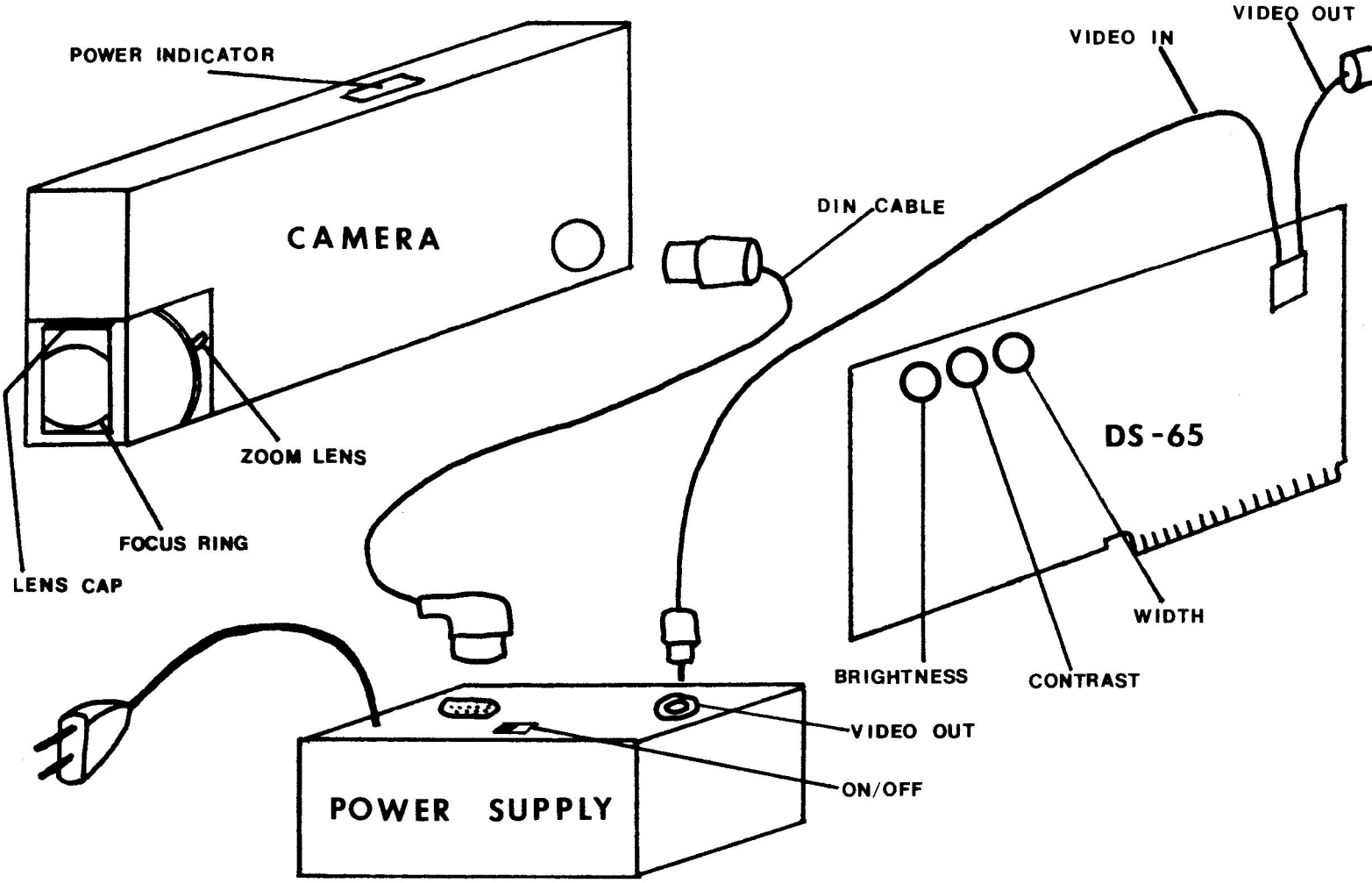


FIGURE 1.

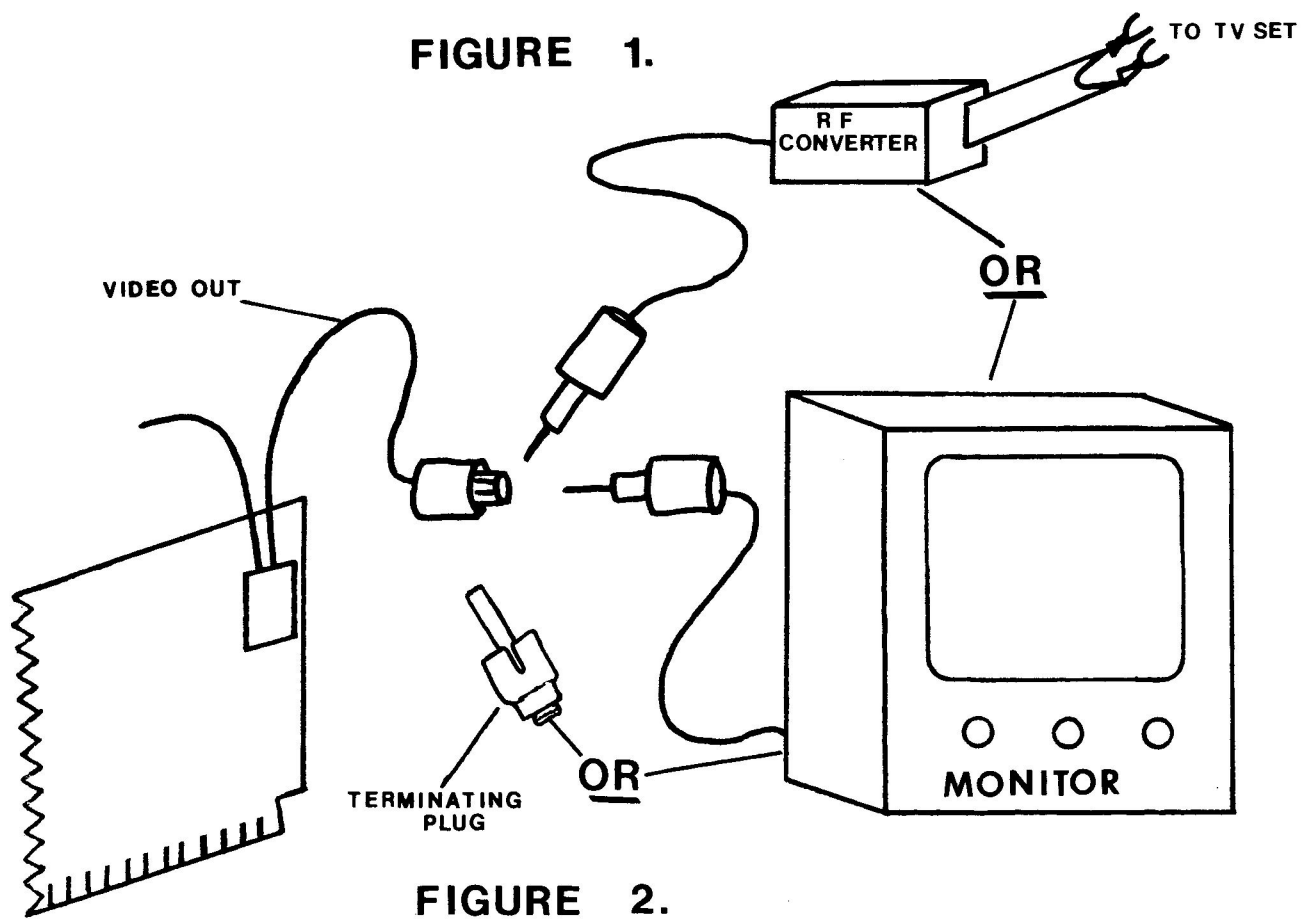


FIGURE 2.

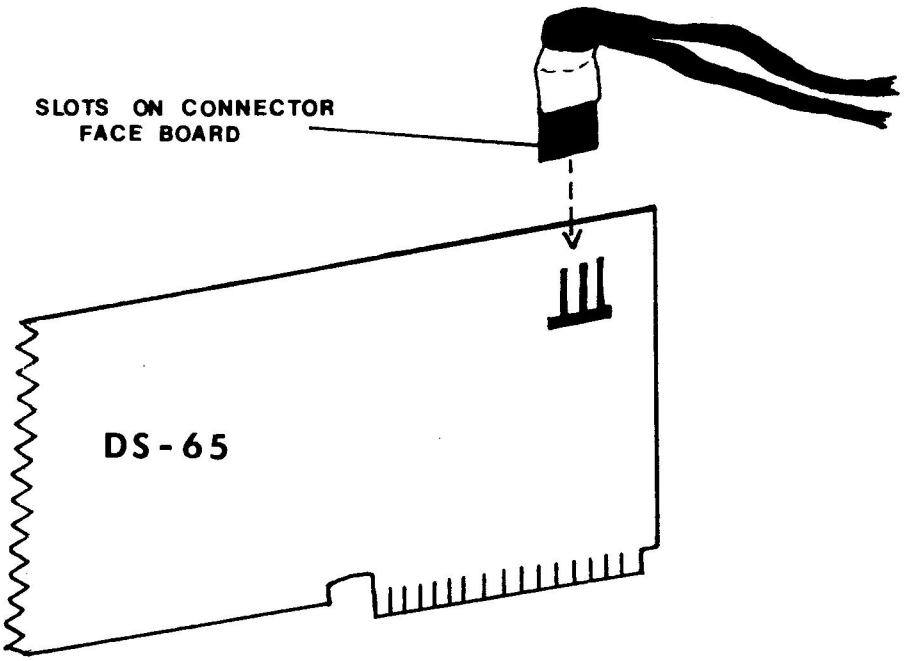


FIGURE 3.

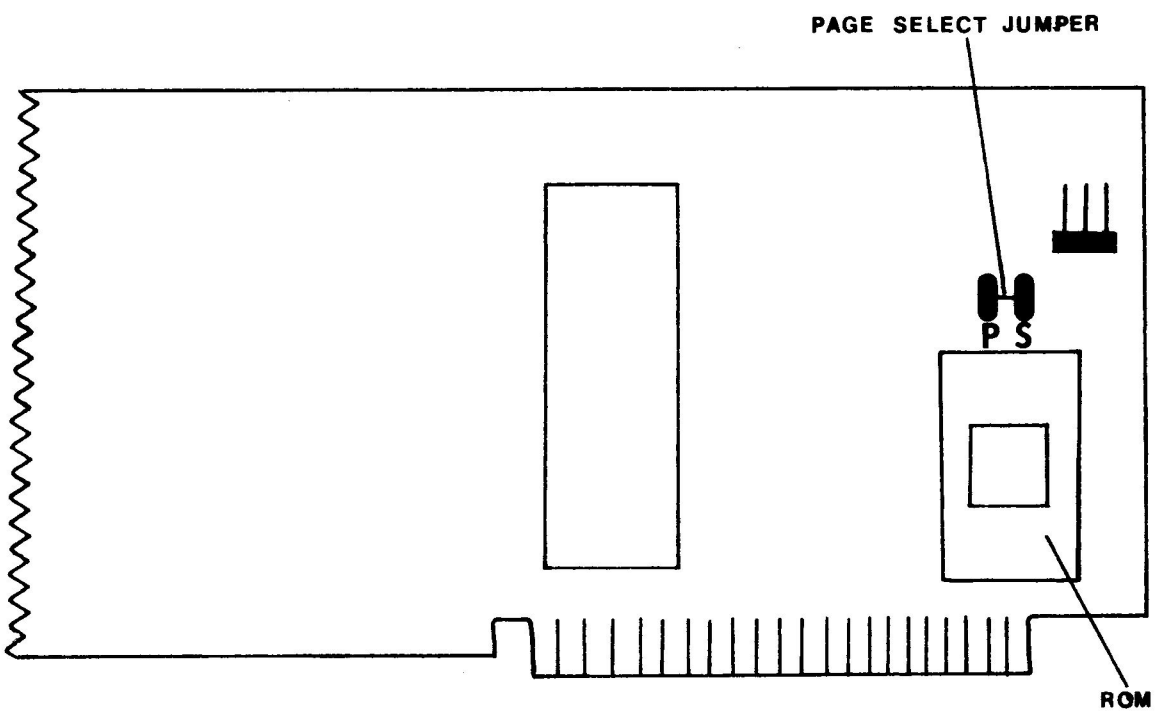


FIGURE 4.

DS-65 DIGISECTOR PARTS LIST

Integrated Circuits	U1	MC6821P
	U2, U3, U4 and U5	74LS85
	U6, U7, U8 and U9	74LS161
	U10	7406
	U11	74123
	U12	74LS00
	U13 and U17	74LS74
	U14	74LS132
	U15	MC14559B
	U16	MC1408L6
	U18	74LS04
	U19	74LS02
	U20	LM311
	U21	CA3140
	U22	LM318
	U23	74LS244
	U24	MCM2708L
Resistors (Ohms)	R1	220
	R2, R6 and R9	75
	R3, R14, R15, R22, R23 and R28	2.2K
	R4	22K
	R5 and R11	10K
	R7	6.8K
	R8 and R19	4.7K
	R10	2.2M
	R12, R21, R27, R29 and R30	1K
	R13	470
	R16 and R18	2K Trim
	R17	500 Trim
	R20	100
	R24	910
	R25 and R26	1.5K
Capacitors	C1, C6 and C10	.1mf.
	C2	.005mf.
	C3	30pf.
	C5	220pf.
	C7	20pf.
	C8	.0022mf.
	C9	.01mf.
Diodes	CR1, CR2, CR3, CR4, CR5 and CR6	1N914
Transistors	Q1 and Q4	2N3906
	Q2 and Q5	2N3904
	Q3	SD211

LIMITED WARRANTY

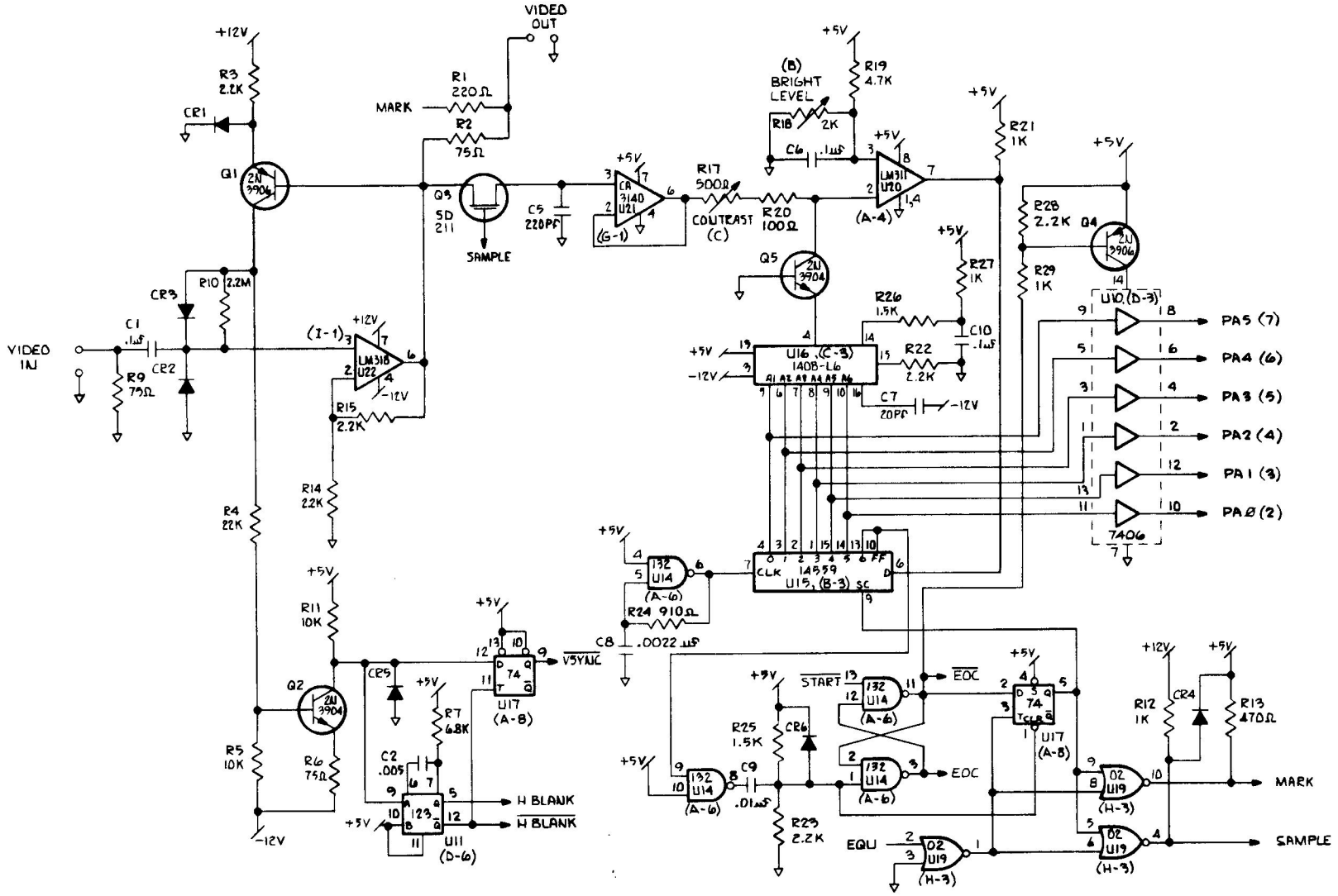
The Micro Works warrants its products to be free from defects in workmanship and materials for a period of ninety (90) days from the date of purchase. IT IS EXPRESSLY AGREED THAT THIS NINETY (90) DAY WARRANTY SHALL BE IN LIEU OF OTHER EXPRESS WARRANTIES, WARRANTIES OF FITNESS AND IN LIEU OF THE WARRANTY OF MERCHANTABILITY. No agent, representative, or employee of the Company has authority to increase or alter the obligation of this warranty.

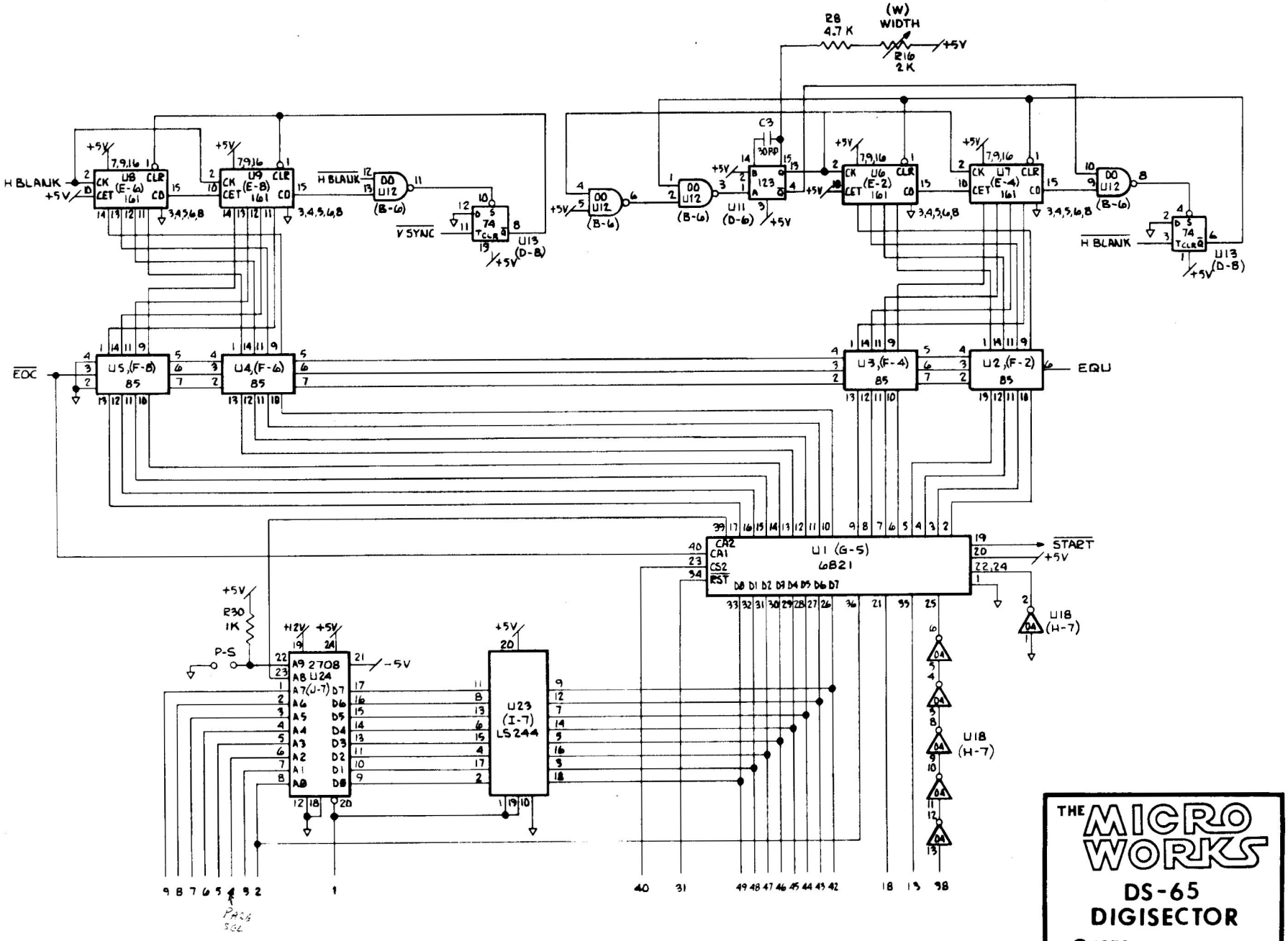
This warranty shall not apply to any Micro Works product which has been modified, repaired or altered in any way. This warranty shall not apply to any product damaged as a result of abuse, misuse, accident or neglect.

In order to make a claim against this warranty the defective board must be returned by private carrier or the U.S. Postal Service to THE MICRO WORKS, P.O. BOX 1110, DEL MAR, CALIFORNIA, 92014. Boards must be accompanied by return shipping charges and the sales receipt showing date of purchase. It is suggested that boards shipped through the United States mails be insured.

REPAIRS

At any time after the expiration of the 90 day warranty period, The Micro Works will repair your PC board for a fee of $25.00, provided that the board is not physically damaged, and that not more than two chips require replacement. If the flat fee is not applicable, you will be notified before further repairs are made. If repairs are necessary, repack the board carefully and enclose a check to The Micro Works, P.O. Box 1110, Del Mar, CA., 92014.


VIDEO OUT
VIDEO IN
+12V
-12V
+5V
R1 220Ω
R2 75Ω
MARK
R3 2.2K
CR1
Q1 2N3906
Q3 SD 211
SAMPLE
C5 220PF
R4 22K
R5 10K
R6 75Ω
R9 75Ω
C1 .1μF
CR2
CR3
R10 2.2M
(I-1)
LM318 U22
R15 2.2K
R14 22K
R11 10K
Q2 2N3904
CR5
R7 6.8K
C2 .005
U11 (D-6)
H BLANK
H BLANK
U17 (A-8)
V5YNC
CA 3140 U21
(G-1)
R17 500Ω CONTRAST (C)
R20 100Ω
(B) BRIGHT LEVEL
R18 2K
R19 4.7K
C6 .1μF
LM311 U20
(A-4)
Q5 2N3904
R21 1K
R28 2.2K
Q4 2N3906
R29 1K
R27 1K
R26 1.5K
C10 .1μF
R22 2.2K
C7 20PF
U16 (C-3) 1408-L6
U10 (D-3)
7406
PA5 (7)
PA4 (6)
PA3 (5)
PA2 (4)
PA1 (3)
PAØ (2)
U15 (B-3) SC
14559
R24 910Ω
C8 .0022 μF
132 U14 (A-6)
START
EOC
R25 1.5K
CR6
C9 .01μF
R23 2.2K
EQU
02 U19 (H-3)
R12 1K
CR4
R13 470Ω
MARK
SAMPLE



THE MICRO WORKS

DS-65
DIGISECTOR

APPENDIX 1

SOURCE LISTING
OF DIGISECTOR ROM

THE FOLLOWING IS THE SOURCE LISTING OF THE PROGRAM WHICH IS SUPPLIED WITH THE DIGISECTOR ON A 2708 EPROM. THIS LISTING IS INCLUDED NOT ONLY TO COMPLETE THE DOCUMENTATION OF THE DS-65 AS SUPPLIED, BUT ALSO TO SERVE AS AN EXAMPLE OF HOW TO USE THE DIGISECTOR IN ASSEMBLY LANGUAGE. THIS LISTING SHOULD BE ESPECIALLY BENEFICIAL TO PROGRAMMERS WHO WANT TO WRITE PROGRAMS TO USE THE DS-65 EFFICIENTLY IN A SPECIFIC APPLICATION.

```
****************************************
*                                      *
*         APPLE ][ DIGISECTOR          *
*                                      *
*              D S - 6 5               *
*                                      *
*       VIDEO DIGITIZER FIRMWARE       *
*                                      *
*   COPYRIGHT 1979 BY THE MICRO WORKS  *
*                                      *
*      WRITTEN BY ANDREW E. PHELPS     *
*                                      *
****************************************

0026            PSTN EQU  $26          CURRENT BYTE ON SCREEN
002A            SAVA EQU  $2A          SAVE A REGISTER HERE
002B            BBIT EQU  $2B          CURRENT BITS ON SCREEN
0040            XCRD EQU  $40          X COORDINATE
0041            YCRD EQU  $41          Y COORDINATE
0042            APRM EQU  $42          ANSWER TO USER
0042            RBIT EQU  $42          RIGHT BIT ON 2-SEC SCAN
0043            LBIT EQU  $43          LEFT BIT ON 2-SEC SCAN
0044            SUM  EQU  $44          RUNNING TOTAL ON 8-SEC
0045            BITS EQU  $45          A THREE FOR A BIT TEST

****************************************
*
*  THE APPLE DIGISECTOR CARD HAS ON IT
*  A 1K READ ONLY MEMORY WHICH CONTAINS
*  THE FOLLOWING PROGRAM.  ONLY 1/4 OF
*  THE SOFTWARE IS AVAILABLE TO THE APPLE
*  AT ANY MOMENT; THAT IS, THE ADDRESS
*  SPACE IS ONLY 256 BYTES.  TWO PAGES
*  MAY BE SWAPPED BY MEANS OF AN OUTPUT
*  LINE FROM THE PIA (CONTROL LINE A2).
*  WHICH PAIR OF PAGES MAY BE SWAPPED IS
*  CONTROLLED BY A PAGE SELECT ("PS")
*  JUMPER ON THE PC BOARD WHICH MAY BE
*  CUT TO ACCESS THE UPPER 1/2 K BYTES.
*
*  ALL FOUR PAGES OF THE ROM START WITH
*  A PROGRAM WHICH SELECTS THE LOWER
*  PAGE OF WHICHEVER HALF OF THE ROM IS
*  IN USE.  THAT LOWER PAGE, WHICH IS THE
*  SAME IN BOTH HALVES (IE, UNAFFECTED BY
*  THE PS JUMPER), HANDLES ALL FUNCTIONS
*  EXCEPT THE FULL SCREEN SCAN.  THE
*  FULL SCREEN SCAN PROGRAM IS IN TWO
*  VERSIONS - EIGHT SECOND OR TWO SECOND -
*  WHICH CAN BE SELECTED BY THE PS JUMPER.
*
```

```
*********************************************
*
*  THE MAP OF THE PROM IS AS FOLLOWS:
*
*    0000-00FF  PART 1 - MISC FUNCTIONS
*    0100-01FF  PART 2 - 2 SECOND SCAN
*    0200-02FF  PART 1 - MISC FUNCTIONS
*    0300-03FF  PART 3 - 8 SECOND SCAN
*
*  ALL FOUR SECTIONS, HOWEVER, SHOW UP IN
*  THE SAME 256 BYTES OF MEMORY SPACE,
*  THAT IS, FROM CX00 TO CXFF WHERE X IS
*  THE SLOT NUMBER WHERE THE BOARD IS
*  PLUGGED IN.  ALL THE PROGRAMS ARE
*  "RUN-ANYWHERE", SO THE PROGRAM IS ORIGINED
*  AT C000 JUST FOR LOOKS.
*
*  THE FUNCTIONS HANDLED BY THE FIRMWARE
*  ARE SPECIFIED BY CALLING THE PROGRAM
*  WITH AN APPROPRIATE CHARACTER IN THE
*  A REGISTER, AS BY SAYING
*    PR#X         (WHERE X IS THE SLOT NUMBER)
*    PRINT "Y";   (WHERE Y IS THE CHARACTER)
*
*  ANY ILLEGAL CHARACTER IS IGNORED.  THE
*  LEGAL CHARACTERS ARE:
*    !  CLEAR HIRES SCREEN AND SET HIRES MODE
*    *  FULL SCREEN SCAN TO HIRES PAGE 1
*    .  DIGITIZE ONE POINT
*    <  SEARCH ONE VERTICAL LINE
*    =  COPY HIRES PAGE 1 TO PAGE 2
*    BELL
*
*  IF SENT A BELL CHARACTER, THE FIRMWARE WILL
*  CALL THE APPLE'S BELL ROUTINE.  THIS WAS
*  INCLUDED SO THAT, IN CASE OF ERROR WHILE
*  PRINT OUTPUT IS SET TO THE DIGISECTOR, THE
*  USER WILL BE AWARE AT LEAST THAT THERE IS
*  AN ERROR CONDITION.
*
```

```
*************************************************************
*
*  PART 1
*
*  THIS ROUTINE APPEARS TWICE IN THE ROM AND IS
*  THUS UNAFFECTED BY THE PAGE SELECT JUMPER.
*
*  IT HANDLES ALL FUNCTIONS EXCEPT THE FULL SCREEN
*  SCAN, FOR WHICH IT TRANSFERS CONTROL TO THE
*  ALTERNATE PAGES.
*
*  IT IS CALLED FROM A PRINT STATEMENT WITH A CHARACTER
*  CODE IN THE A REGISTER.
*
*
C000                ORG  $C000
C000  85 2A         STA  SAVA          SAVE OUTPUT CHARACTER
C002  8A            TXA
C003  48            PHA                SAVE X REGISTER
C004  98            TYA
C005  48            PHA                SAVE Y REGISTER
C006  08            PHP                SAVE INTERRUPT MASK
C007  78            SEI                INHIBIT INTERRUPTS
C008  20 58FF       JSR  $FF58         TO AN RTS
C00B  BA            TSX                GET STACK ADDRESS
C00C  BD 0001       LDA  $100,X        GET MS BYTE OF RET ADDR
C00F  28            PLP                RESTORE INTERRUPT MASK
C010  A8            TAY                SAVE 'CX' IN Y REG
C011  0A            ASL
C012  0A            ASL
C013  0A            ASL
C014  0A            ASL
C015  AA            TAX                SAVE 'X0' IN X REG
C016  A9 34         LDA  #$34          PAGE SELECT = 0
*
*  THE FOLLOWING LINE DOES THE PAGE SELECTION;
*  A $34 IN A SELECTS PAGE 0, AND A $3C SELECTS PAGE 1.
C018  9D 81C0 PGSL  STA  $C081,X       TO CONTROL REG A OF PIA
C01B  BD 83C0       LDA  $C083,X       CHECK CONTRL REG B
C01E  29 3F         AND  #$3F          IGNORE FLAG BITS
C020  C9 2C         CMP  #$2C          SEE IF ALREADY INIT
C022  F0 1C         BEQ  NINT          BRANCH IF NO INIT NEEDED
C024  A9 30         LDA  #$30          SELECT DIRECTION REG
C026  9D 81C0       STA  $C081,X       TO CONTROL REG
C029  A9 00         LDA  #0            OTHER DIRECTION REG
C02B  9D 83C0       STA  $C083,X       OTHER CONTROL REG
C02E  A9 FF         LDA  #$FF          ALL OUTPUTS
C030  9D 80C0       STA  $C080,X       TO DIR REG
C033  9D 82C0       STA  $C082,X       TO OTHER DIR REG
C036  A9 34         LDA  #$34          CONTRL LINE 2 LOW
C038  9D 81C0       STA  $C081,X       SET CONTROL REG A
C03B  A9 2C         LDA  #$2C          AUTO STROBE LINE 2
C03D  9D 83C0       STA  $C083,X       SET CONTROL REG B
C040  A5 2A   NINT  LDA  SAVA          GET BYTE TO OUTPUT
C042  29 3F         AND  #$3F          IGNORE CASE & PARITY
C044  C9 2A         CMP  #$2A          IS IT AN '*'?
C046  D0 04         BNE  NAST          SKIP OVER FULL SCAN
C048  A9 3C         LDA  #$3C          PAGE SELECT HIGH
```

```
C04A  D0 CC        BNE   PGSL        GO SWITCH PAGES
C04C  C9 2E  NAST  CMP   #$2E        IS IT A PERIOD?
C04E  D0 1C        BNE   NPER        SKIP OVER READ POINT
C050  A5 40        LDA   XCRD        GET X PARAMETER
C052  9D 80C0      STA   $C080,X     TO X SIDE OF PIA
C055  A5 41        LDA   YCRD        GET Y PARAMTER
C057  9D 82C0      STA   $C082,X     TO Y SIDE OF PIA
C05A  BD 81C0 RLP  LDA   $C081,X     GET DONE FLAG
C05D  10 FB        BPL   RLP         LOOP TIL DONE
C05F  A9 3F        LDA   #$3F        BITS USED FOR INPUT
C061  9D 80C0      STA   $C080,X     SET THOSE BITS HIGH
C064  5D 80C0      EOR   $C080,X     READ, AND MAKE WHITES ONES
C067  85 42        STA   APRM        PUT WHERE USER CAN FIND
C069  18           CLC               ENABLE MESS OF BRANCHES
C06A  90 22        BCC   OUT1        LEAVE
C06C  C9 21  NPER  CMP   #$21        IS IT AN '!'?
C06E  D0 20        BNE   NEXL        SKIP OVER CLEAR PAGE
C070  A9 20        LDA   #$20        MSB OF HIRES PAGE 1
C072  85 41        STA   YCRD        USED AS TEMP
C074  A9 00  CPAG  LDA   #0          LSB OF HIRES
C076  85 40        STA   XCRD        TEMP LSB
C078  A8           TAY               CLEAR INDEX
C079  91 40  CLOP  STA   (XCRD),Y    CLEAR BYTE
C07B  C8           INY               NEXT BYTE
C07C  D0 FB        BNE   CLOP        CLEAR 256 BYTES
C07E  E6 41        INC   YCRD        NEXT MSB
C080  A5 41        LDA   YCRD        CHECK MSB
C082  0A           ASL               PUT $40 INTO MS BIT
C083  10 EF        BPL   CPAG        GO CLEAR NEXT BUNCH
C085  AD 50C0      LDA   $C050       SET GRAPHICS
C088  AD 54C0      LDA   $C054       SET PAGE 1
C08B  AD 57C0      LDA   $C057       SET HIRES
C08E  90 23  OUT1  BCC   OUT2        LEAVE
C090  C9 3D  NEXL  CMP   #$3D        IS IT AN '='?
C092  D0 26        BNE   NEQL        SKIP OVER COPY PAGE
C094  A9 20        LDA   #$20        MSB OF HIRES PAGE 1
C096  85 41        STA   YCRD        USED AS A TEMP
C098  A9 40        LDA   #$40        MSB OF HIRES PAGE 2
C09A  85 27        STA   PSTN+1      WHERE TO PUT IT
C09C  A9 00        LDA   #0          LSB OF BOTH
C09E  85 40        STA   XCRD        TEMP LSB
C0A0  85 26        STA   PSTN        DEST LSB
C0A2  A8           TAY               CLEAR INDEX
C0A3  B1 40  MLOP  LDA   (XCRD),Y    GET BYTE
C0A5  91 26        STA   (PSTN),Y    PUT IT IN PAGE 2
C0A7  C8           INY               BUMP INDEX
C0A8  D0 F9        BNE   MLOP        LOOP TO MOVE NEXT
C0AA  E6 41        INC   YCRD        NEXT MSB
C0AC  E6 27        INC   PSTN+1      NEXT MSB ALSO
C0AE  A5 41        LDA   YCRD        GET MSB
C0B0  0A           ASL               MOVE $40 TO SIGN
C0B1  10 F0        BPL   MLOP        LOOP FOR NEXT BUNCH
C0B3  68     OUT2  PLA
C0B4  A8           TAY               RESTORE Y REG
C0B5  68           PLA
C0B6  AA           TAX               RESTORE X REG
C0B7  A5 2A        LDA   SAVA        RESTORE A REG
C0B9  60           RTS               RETURN
```

```
C0BA  C9 3C    NEQL CMP  #$3C        IS IT A '<'?
C0BC  D0 2D         BNE  NLST        SKIP OVER SEARCH FUNCTION
C0BE  A9 10         LDA  #$10        START Y AT 16
C0C0  85 42         STA  APRM        CURRENT Y POSTION
C0C2  A5 40    LDRK LDA  XCRD        GET X PARAMETER
C0C4  9D 80C0       STA  $C080,X     TO X SIDE OF PIA
C0C7  A5 42         LDA  APRM        GET CURRENT Y
C0C9  9D 82C0       STA  $C082,X     TO Y SIDE OF PIA
C0CC  18            CLC
C0CD  69 02         ADC  #2          BUMP Y BY TWO
C0CF  85 42         STA  APRM        SAVE NEXT Y
C0D1  F0 13         BEQ  LA1         STOP IF HIT BOTTOM
C0D3  BD 81C0  LA2  LDA  $C081,X     CHECK DONE FLAG
C0D6  10 FB         BPL  LA2         LOOP TIL READY
C0D8  A9 3F         LDA  #$3F        BITS FOR INPUT
C0DA  9D 80C0       STA  $C080,X     SET TO INPUT
C0DD  5D 80C0       EOR  $C080,X     INPUT AND COMPLEMENT
C0E0  C5 41         CMP  YCRD        CHECK REQUISITE DARKNESS
C0E2  B0 DE         BCS  LDRK        NOT GOOD ENOUGH
C0D4  90 CD         BCC  OUT2        OK - WE'RE DONE
C0E6  9D 82C0  LA1  STA  $C082,X     CLEAR Y PIA
C0E9  F0 C8         BEQ  OUT2        LEAVE
C0EB  C9 07    NLST CMP  #7          IS IT A BELL?
C0ED  D0 C4         BNE  OUT2        IGNORE ANYTHING ELSE
C0EF  20 DDFB       JSR  $FBDD       GO RING BELL
C0F2  18            CLC              TO ALLOW JUMP
C0F3  90 BE         BCC  OUT2        LEAVE
```

```
*************************************************
*
*  PART 2
*
*  EIGHT SECOND SCAN
*
C100                    ORG   $C100
C100   85 2A            STA   SAVA          SAVE OUTPUT CHARACTER
C102   8A               TXA
C103   48               PHA                 SAVE X REGISTER
C104   98               TYA
C105   48               PHA                 SAVE Y REGISTER
C106   08               PHP                 SAVE INTERRUPT MASK
C107   78               SEI                 INHIBIT INTERRUPTS
C108   20 58FF          JSR   $FF58         TO AN RTS
C10B   BA               TSX                 GET STACK ADDRESS
C10C   BD 0001          LDA   $100,X        GET MS BYTE OF RET ADDR
C10F   28               PLP                 RESTORE INTERRUPT MASK
C110   A8               TAY                 SAVE 'CX' IN Y REG
C111   0A               ASL
C112   0A               ASL
C113   0A               ASL
C114   0A               ASL
C115   AA               TAX                 SAVE 'X0' IN X REG
C116   A9 34            LDA   #$34          PAGE SELECT = 0
C118   9D 81C0          STA   $C081,X       CONTROL REG A OF PIA
*
*  THE PRECEDING SECTION TURNS CONTROL OVER TO
*  PART 1.  WHEN SENT THE CODE FOR A SCAN,
*  CONTROL IS TRANSFERRED TO THE FOLLOWING
*  LINE.
*
C11B   A9 03            LDA   #3            PATTERN FOR BIT TEST
C11D   85 45            STA   BITS          SAVE IT FOR FUTURE
C11F   A9 01            LDA   #1            START POSTION
C121   85 26            STA   PSTN          IN HIRES SCREEN
C123   A9 01            LDA   #1            START X POSITION
C125   85 40            STA   XCRD          X COORDINATE
C127   9D 82C0          STA   $C082,X       SOMETHING TO CHEW ON
C12A   A9 01     NBYT   LDA   #1            FIRST BIT TO STORE
C12C   85 2B            STA   BBIT          PUT IN BIT COUNTER
C12E   A9 20     NBIT   LDA   #$20          FIRST Y POSITION
C130   85 41            STA   YCRD          Y COORDINATE
C132   A9 20            LDA   #$20          MSB HIRES PAGE 1
C134   85 27     NSCN   STA   PSTN+1        POSITION MSB
C136   A9 00            LDA   #0
C138   A8               TAY                 CLEAR INDEX
C139   18               CLC                 FOR FUTURE ADDS
C13A   BD 81C0   LP1    LDA   $C081,X       CHECK DONE FLAG
C13D   10 FB            BPL   LP1           LOOP TILL DONE
C13F   A5 40            LDA   XCRD          GET X COORDINATE
C141   9D 80C0          STA   $C080,X       TO X SIDE OF PIA
C144   A5 41            LDA   YCRD          GET Y COORDINATE
C146   9D 82C0          STA   $C082,X       TO Y SIDE OF PIA
C149   69 02            ADC   #2            ADD 2
C14B   85 41            STA   YCRD          NEXT Y COORDINATE
C14D   18        NEWY   CLC                 CLEAR CARRY
C14E   BD 81C0   LP2    LDA   $C081,X       CHECK DONE FLAG
```

```
C151  10 FB        BPL  LP2          LOOP TILL DONE
C153  A9 3F        LDA  #$3F         SET BITS FOR INPUT
C155  9D 80C0      STA  $C080,X
C158  5D 80C0      EOR  $C080,X      INPUT AND COMPLEMENT
C15B  65 44        ADC  SUM          ADD TO RUNNING TOTAL
C15D  B0 07        BCS  CARY         BRANCH IF CARRY
C15F  85 44        STA  SUM          SAVE TOTAL
C161  90 0D        BCC  NCAR         DON'T PUT DOT
C163  00           BRK
C164  D0 C4   B1   BNE  NBYT
C166  69 C0   CARY ADC  #$C0         CARRY EVERY 40
C168  85 44        STA  SUM          SAVE RUNNING SUM
C16A  B1 26        LDA  (PSTN),Y     GET DOTS THERE
C16C  05 2B        ORA  BBIT         ADD THIS DOT
C16E  91 26        STA  (PSTN),Y     PUT BACK IN SCREEN
C170  A5 40   NCAR LDA  XCRD         GET X COORDINATE
C172  9D 80C0      STA  $C080,X
C175  A5 41        LDA  YCRD         GET Y COORDINATE
C177  9D 82C0      STA  $C082,X
C17A  69 02        ADC  #2           INCREMENT Y
C17C  85 41        STA  YCRD         SAVE NEW Y
C17E  A5 27        LDA  PSTN+1       GET CURRENT POSITION
C180  69 08        ADC  #8           NEXT DOT BELOW
C182  85 27        STA  PSTN+1
C184  0A           ASL               MAKE 40 NEGATIVE
C185  10 C6        BPL  NEWY         NEXT Y IF OK
C187  98           TYA               GET INDEX
C188  69 80        ADC  #$80         NEXT DOT
C18A  A8           TAY               BACK TO INDEX
C18B  B0 0E        BCS  CAR2
C18D  A5 27        LDA  PSTN+1
C18F  69 E0        ADC  #$E0         PUT MSB BACK
C191  85 27        STA  PSTN+1       SAVE MSB
C193  B0 B8        BCS  NEWY         ALWAYS BRANCH
C195  00           BRK
C196  D0 9C   B2   BNE  NSCN         TO NEXT SCAN
C198  10 94   B3   BPL  NBIT         TO NEXT BIT
C19A  00           BRK
C19B  A5 27   CAR2 LDA  PSTN+1
C19D  69 E0        ADC  #$E0         PUT MSB BACK
C19F  24 45        BIT  BITS         CHECK BOTTOM 2 BITS
C1A1  F0 07        BEQ  NOBT
C1A3  85 27        STA  PSTN+1
C1A5  D0 A6        BNE  NEWY         ALWAYS BRANCH
C1A7  00           BRK
C1A8  EA           NOP
C1A9  EA           NOP
C1AA  69 FB   NOBT ADC  #$FB         CORRECT MSB
C1AC  85 27        STA  PSTN+1
C1AE  98           TYA               GET INDEX
C1AF  69 27        ADC  #$27         NEXT BIT IN SCREEN
C1B1  A8           TAY               BACK IN INDEX
C1B2  C9 78        CMP  #$78         AT END?
C1B4  D0 98        BNE  LP2
C1B6  EA           NOP
C1B7  EA           NOP
C1B8  EA           NOP
C1B9  EA           NOP
```

```
C1BA  46 41        LSR  YCRD      CHECK BOTTOM BIT
C1BC  B0 09        BCS  ODD       BRANCH IF Y ODD
C1BE  A9 21        LDA  #$21      OTHER HALF OF Y'S
C1C0  85 41        STA  YCRD
C1C2  A9 24        LDA  #$24      OTHER HALF OF SCREEN
C1C4  D0 D0        BNE  B2
C1C6  00           BRK
C1C7  A5 40   ODD  LDA  XCRD      X COORDINATE
C1C9  69 00        ADC  #0        ADD 1
C1CB  85 40        STA  XCRD
C1CD  B0 08        BCS  EXIT      LEAVE IF DONE
C1CF  06 2B        ASL  BBIT      NEXT BIT IN SCREEN
C1D1  10 C5        BPL  B2        LOOP IF STILL IN BYTE
C1D3  E6 26        INC  PSTN      NEXT LSB
C1D5  D0 8D        BNE  B1        NEXT BYTE
C1D7  68      EXIT PLA
C1D8  A8           TAY            RESTORE Y
C1D9  68           PLA
C1DA  AA           TAX            RESTORE X
C1DB  A5 2A        LDA  SAVA      RESTORE A
C1DD  60           RTS            RETURN TO USER
```

```
*************************************************
*
*  PART 3
*
*  TWO SECOND SCAN
*
C200                ORG  $C200
C200  85 2A         STA  SAVA        SAVE OUTPUT CHARACTER
C202  8A            TXA
C203  48            PHA              SAVE X REGISTER
C204  98            TYA
C205  48            PHA              SAVE Y REGISTER
C206  08            PHP              SAVE INTERRUPT MASK
C207  78            SEI              INHIBIT INTERRUPTS
C208  20 58FF       JSR  $FF58       TO AN RTS
C20B  BA            TSX              GET STACK ADDRESS
C20C  BD 0001       LDA  $100,X      GET MS BYTE OF RET ADDR
C20F  28            PLP              RESTORE INTERRUPT MASK
C210  A8            TAY              SAVE 'CX' IN Y REG
C211  0A            ASL
C212  0A            ASL
C213  0A            ASL
C214  0A            ASL
C215  AA            TAX              SAVE 'X0' IN X REG
C216  A9 34         LDA  #$34        PAGE SELECT = 0
C218  9D 81C0       STA  $C081,X     CONTROL REG A OF PIA
*
*  THE PRECEDING SECTION TURNS CONTROL OVER TO
*  PART 1.  WHEN SENT THE CODE FOR A SCAN,
*  CONTROL IS TRANSFERRED TO THE FOLLOWING
*  LINE.
*
C21B  A9 03         LDA  #3          BITS FOR BIT TEST
C21D  85 45         STA  BITS        SAVE FOR LATER TEST
C21F  A9 03         LDA  #3          STARTING POSITION
C221  85 26         STA  PSTN+1      POSTION LSB IN HIRES
C223  A9 01         LDA  #1          STARTING X COOR
C225  85 40         STA  XCRD        X COORDINATE
C227  9D 82C0       STA  $C082,X     REV UP PIA
C22A  A9 03    NCB  LDA  #3
C22C  85 2B         STA  BBIT        MASK TO STORE 2 BITS
C22E  4A            LSR              LDA  #1
C22F  85 42         STA  RBIT        MASK TO STORE RIGHT BIT
C231  0A            ASL              LDA  #2
C232  85 43         STA  LBIT        MASK TO STORE LEFT BIT
C234  A9 20    NCL  LDA  #$20        STARTING VALUE
C236  85 27         STA  PSTN+1      HIRES MSB
C238  A0 00         LDY  #0          CLEAR INDEX
C23A  BD 81C0  WT1  LDA  $C081,X     CHECK DONE FLAG
C23D  10 FB         BPL  WT1         LOOP TIL DONE
C23F  A5 40         LDA  XCRD        X COORDINATE
C241  9D 80C0       STA  $C080,X     TO X SIDE OF PIA
C244  A9 20         LDA  #$20        STARTING Y VALUE
C246  9D 82C0       STA  $C082,X     TO Y SIDE OF PIA
C249  A9 22         LDA  #$22        NEXT Y VALUE
C24B  85 41         STA  YCRD        SAVE AS Y VALUE
C24D  18       NPT  CLC              FOR FUTURE ADDS
C24E  BD 81C0  WT2  LDA  $C081,X     CHECK DONE FLAG
```

```
C251  10 FB        BPL  WT2          LOOP TIL DIGITIZED
C253  A9 3F        LDA  #$3F        BITS FOR INPUT
C255  9D 80C0      STA  $C080,X     SET PIA SIDE TO INPUT
C258  5D 80C0      EOR  $C080,X     INPUT AND COMPLEMENT
C25B  D0 08        BNE  BW1         BRANCH IF NOT ALL BLACK
C25D  A5 27        LDA  PSTN+1      HIRES POSITION MSB
C25F  69 04        ADC  #4          NEXT DOT
C261  85 27        STA  PSTN+1      SAVE IT BACK
C263  90 38        BCC  NCR         GO DO NEW POINT
C265  29 30   BW1  AND  #$30        GET HIGH BITS ONLY
C267  F0 28        BEQ  BW5         GO PUT ONE DOT
C269  C9 20        CMP  #$20        THREE DOTS?
C26B  F0 1A        BEQ  BW3         GO PUT 3 DOTS
C26D  90 1C        BCC  BW4         IF LESS PUT 2 DOTS
C26F  A5 2B        LDA  BBIT        FOR 4 DOTS
C271  11 26   BW2  ORA  (PSTN),Y    SET THE BITS
C273  91 26        STA  (PSTN),Y    AND PUT IN SCREEN
C275  A5 27        LDA  PSTN+1      MSB
C277  69 03        ADC  #3          ADD 4
C279  85 27        STA  PSTN+1      AND SAVE IT
C27B  A5 2B        LDA  BBIT        OTHER TWO DOTS
C27D  11 26        ORA  (PSTN),Y    SET THE BITS
C27F  91 26        STA  (PSTN),Y    AND PUT IN SCREEN
C281  90 1A        BCC  NCR         ALWAYS BRANCH
C283  D0 AF   NCL2 BNE  NCL
C285  D0 A3   NCB2 BNE  NCB
C287  A5 42   BW3  LDA  RBIT        RIGHTHAND BIT
C289  B0 E6        BCS  BW2         GO PUT INTO SCREEN
C28B  A5 42   BW4  LDA  RBIT        RIGHTHAND BIT
C28D  11 26        ORA  (PSTN),Y    ADD TO SCREEN
C28F  91 26        STA  (PSTN),Y    PUT INTO SCREEN
C291  A5 27   BW5  LDA  PSTN+1      MSB
C293  69 04        ADC  #4          NEXT DOT DOWN
C295  85 27        STA  PSTN+1      AND SAVE IT
C297  A5 43        LDA  LBIT        LEFTHAND BIT
C299  11 26        ORA  (PSTN),Y    ADD TO SCREEN
C29B  91 26        STA  (PSTN),Y    PUT INTO SCREEN
C29D  A5 40   NCR  LDA  XCRD        X COORDINATE
C29F  9D 80C0      STA  $C080,X     TO X SIDE OF PIA
C2A2  A5 41        LDA  YCRD        Y COORDINATE
C2A4  9D 82C0      STA  $C082,X     TO Y SIDE OF PIA
C2A7  69 02        ADC  #2          NEXT Y
C2A9  85 41        STA  YCRD        SAVE IT
C2AB  A5 27        LDA  PSTN+1
C2AD  69 04        ADC  #4          NEXT DOT DOWN
C2AF  85 27        STA  PSTN+1      NEW HIRES MSB
C2B1  0A           ASL              CHECK BIT 6
C2B2  10 99   NPT2 BPL  NPT         OK IF WAS ZERO
C2B4  98           TYA              CHECK INDEX
C2B5  69 80        ADC  #$80        NEXT DOT DOWN
C2B7  A8           TAY              BACK TO INDEX
C2B8  B0 0A        BCS  LQ1         BRANCH IF NEW MSB
C2BA  A5 27        LDA  PSTN+1
C2BC  69 E0        ADC  #$E0        CORRECT MSB
C2BE  85 27        STA  PSTN+1
C2C0  B0 8B        BCS  NPT         BRANCH ALWAYS
C2C2  D0 8A   WT22 BNE  WT2
C2C4  A5 27   LQ1  LDA  PSTN+1
```

```
C2C6  69 E0        ADC  #$E0        CORRECT MSB & ADD 1
C2C8  24 45        BIT  BITS        CHECK FOR END
C2CA  F0 04        BEQ  LQ2
C2CC  85 27        STA  PSTN+1      SAVE IT
C2CE  D0 E2        BNE  NPT2        BRANCH ALWAYS
C2D0  69 FB   LQ2  ADC  #$FB        CORRECT AGAIN
C2D2  85 27        STA  PSTN+1      AND SAVE IT
C2D4  98           TYA              GET INDEX AGAIN
C2D5  69 27        ADC  #$27        NEXT DOT DOWN AGAIN
C2D7  A8           TAY              BACK TO INDEX
C2D8  C9 78        CMP  #$78        AT END?
C2DA  D0 E6        BNE  WT22        OK; GET NEXT POINT
C2DC  06 43        ASL  LBIT
C2DE  06 43        ASL  LBIT
C2E0  06 42        ASL  RBIT
C2E2  06 42        ASL  RBIT
C2E4  06 2B        ASL  BBIT
C2E6  EA           NOP
C2E7  18           CLC              FOR THE ADD
C2E8  A5 40        LDA  XCRD        X COORDINATE
C2EA  69 02        ADC  #2          NEXT X
C2EC  85 40        STA  XCRD        SAVE IT
C2EE  B0 08        BCS  BYE         LEAVE IF OFF END
C2F0  06 2B        ASL  BBIT
C2F2  D0 8F        BNE  NCL2
C2F4  E6 26        INC  PSTN        NEXT BYTE OF SCREEN
C2F6  D0 8D        BNE  NCB2        GO DO NEXT BYTE
C2F8  68      BYE  PLA
C2F9  A8           TAY              RESTORE Y
C2FA  68           PLA
C2FB  AA           TAX              RESTORE X
C2FC  A5 2A        LDA  SAVA        RESTORE A
C2FE  60           RTS              LEAVE
C2FF  00           BRK

                   END
```

APPENDIX 2: SAMPLE BASIC PROGRAMS

The following programs are included to give the user some examples of using the DS-65 from a BASIC program.

PROGRAM 1
Motion Detector

This program repeatedly scans the target area, watching for significant motion. Such a program could be used as a burglar alarm, sensing the presence of unwanted objects.

```
100  PRINT "DIGISECTOR IN WHICH SLOT";
110  INPUT S
120  IF S<1 THEN 100
130  IF S>7 THEN 100
140  PR#S
150  REM - F IS THE FIRST TIME THRU FLAG
160  F=1
170  REM - L IS POINTER INTO HI-RES AREA
180  REM   WHICH IS USED FOR STORAGE
190  L=8192
200  REM - N IS THE NUMBER OF CHANGED POINTS
210  N=0
220  FOR X=1 TO 250 STEP 20
230  FOR Y=1 TO 250 STEP 20
240  POKE 64,X
250  POKE 65,Y
260  REM - DIGITIZE POINT (X,Y)
270  PRINT "."
280  D=PEEK(66)
290  Z=ABS(PEEK(L)-D)
300  POKE L,D
310  REM - THE VALUE "5" CAN BE CHANGED FOR SENSITIVITY
320  IF Z>5 THEN N=N+1
330  L=L+1
340  NEXT Y
350  NEXT X
360  Q=F
370  F=0
380  IF Q=1 THEN 190
390  REM - THE VALUE "10" CAN BE CHANGED FOR OBJECT SIZE
400  IF N<10 THEN 190
410  REM - THE FOLLOWING SECTION OF CODE SHOULD DO
420  REM   WHATEVER YOU WANT DONE WHEN AN OBJECT
430  REM   IS FOUND
440  PR#0
450  FOR B=1 TO 7
460  REM - THE FOLLOWING LINE CONTAINS A BELL (CNTR-G)
470  PRINT "";
480  NEXT B
490  GOTO 140
500  END
```

PROGRAM 2
Lo-Res Graphing

This program is included as an example of using the PRINT "<" function. It will read a graph and produce a copy of it on the Lo-Res screen. It is recommended that the contrast on the DS-65 be turned up, and the brightness adjusted, whenever doing graphing, tracking, or other inherently high-contrast applications.

```
100  PRINT "DIGISECTOR IN WHICH SLOT";
110  INPUT S
120  IF S<1 THEN 100
130  IF S>7 THEN 100
140  V=-1
150  GR
160  COLOR=15
170  FOR X=0 TO 39
180  REM - THE FOLLOWING EXPRESSION COULD BE CHANGED
190  REM   TO SCAN A DIFFERENT AREA OF THE SCREEN
200  POKE 64,X*6+21
210  REM - THE FOLLOWING IS THE BRIGHTNESS THRESHOLD
220  POKE 65,5
230  REM - NOW DIGITIZE THE LINE
240  PR#S
250  PRINT "<"
260  PR#0
270  Y=PEEK(66)/5+1
280  IF V>-1 THEN 310
290  PLOT X,Y
300  GOTO 330
310  IF V>Y THEN VLIN Y,V AT X
320  IF V<=Y THEN VLIN V,Y AT X
330  V=Y
340  NEXT X
350  GOTO 140
360  END
```

N O T E S