DMC Floppy Disk Controller

Hardware Manual

Preliminary Version

Sardis Technologies
Ph. (604) 255-4485 (Pacific Time)

2261 East 11th Ave.
Vancouver, B.C.
Canada V5N 1Z7

Printed in Canada

CREDITS

Special thanks to Jim B., whose help and encouragement made this product possible.

TRADEMARKS

"Radio Shack" and "Color Computer" are trademarks of Tandy Corp.
"OS-9" is a trademark of Microware Systems Corp. and Motorola

COPYRIGHT NOTICE

DISCLAIMER

Although much effort has been made to ensure the accuracy of the hardware, software and documentation, Sardis Technologies disclaim any and all liability for consequential damages, economic loss, or any other injury arising from or on account of the use of, possession of, defect in, or failure of the supplied material.

This manual last revised July 14, 1987

TABLE OF CONTENTS

SYSTEM PREREQUISITES - or, what you need to run the DMC on a CoCo

1) A Tandy / Radio Shack Color Computer 1 or 2 with 64K memory, or a Color Computer 3 with either 128K or 512K memory.

2) A Radio Shack "Multi-Pak Interface" (MPI), or similar unit. The DMC's power consumption is somewhat greater than what Radio Shack specifies may be drawn from the CoCo's cartridge slot. Although the DMC controller will run when plugged directly into the CoCo, you risk overheating and damaging the CoCo's internal power supply if you omit the MPI.

3) At least one 5 1/4" floppy disk drive, with its own power supply, and a 34 conductor ribbon cable with edge-card connector on its end to connect to the DMC. Up to 3 double-sided or 4 single-sided drives may be connected, and they may be 48 tpi (35 or 40 tracks) or 96 tpi (80 tracks), or even 3.5" drives (135 tpi). However, for software compatibility reasons, drive 0 should be some type of 48 tpi drive (35 or 40 tracks).

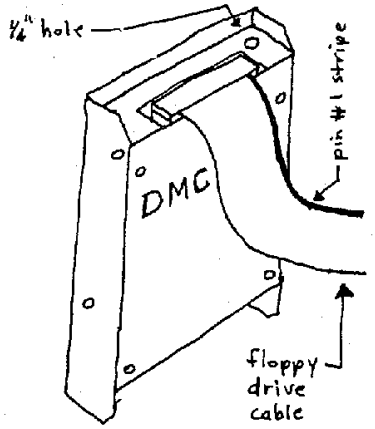
SETTING UP -- INSTALLATION OF THE DMC CARTRIDGE

1) If you bought the DMC controller without a ROM installed, first carry out the instructions in the following 2 sections to install your ROM.

2) Before connecting the floppy drive cable to the DMC controller, and before plugging the DMC controller into the Multi-Pak Interface (MPI), TURN OFF THE POWER TO - the CoCo
   - the MPI
   - the floppy disk drives

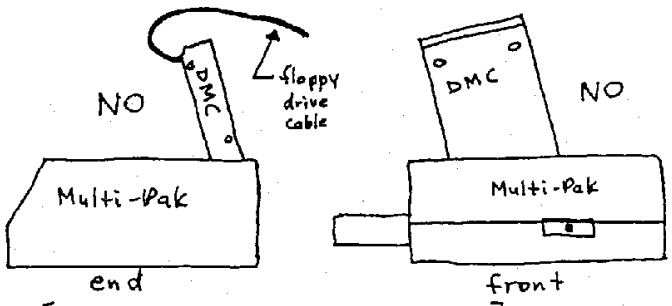
```
+-------------------------------------------------------------------+
! Connecting or disconnecting the various parts of the system with  !
! the power on in even only one of the components can severely damage
! the system.                                                       !
+-------------------------------------------------------------------+
```

3) First plug the 34 conductor ribbon cable (from the disk drives) into the DMC controller, into the end that also has the 1/4" round hole. Refer to the drawing below for the correct orientation of the cable:



Floppy drive cable orientation

4) Now plug the other end of the DMC controller cartridge into the MPI (usually slot 4) with the DMC's venting slots facing the rear of the MPI. Be careful to ensure that the controller cartridge is firmly and properly seated into the connector in the slot, and that it is not installed crookedly. Refer to the drawings below for what NOT to do:



[Should not be installed crooked]

5) Now turn on the power to the TV set or monitor, then the disk drives, then the Multi-Pak, and finally the CoCo itself. You should see the sign-on message indicating "Disk Extended BASIC 1.1" for the CoCo 1 or 2, and "Disk Extended BASIC 2.1" for the CoCo 3. If not, immediately turn the power off, and refer to the section entitled "Trouble-Shooting and Maintenance".

NOTE - since the 1773 FDC chip used in the DMC controller doesn't handle head load delay, any disk drives that have head load solenoids should be configured to load their heads with the motor-on signal, not with the drive select signal. Refer to the OEM manual or service manual for your particular drive on how to change or check this setting.

INSTALLING A 24 PIN ROM

If you bought the DMC controller without the RS-DOS 1.1 ROM chip installed and want to install your own 24 pin ROM, read this section. If you will be installing a 28 pin EPROM, read the next section.

We advise you to remove the circuit board from the metal case to install the ROM -- the board might bend too much from the pressure exerted to insert the ROM chip, causing damage to the board.

1) Read the warning in the next section regarding static electricity, then follow the instructions in step #1 in that section to remove the board from the metal case.

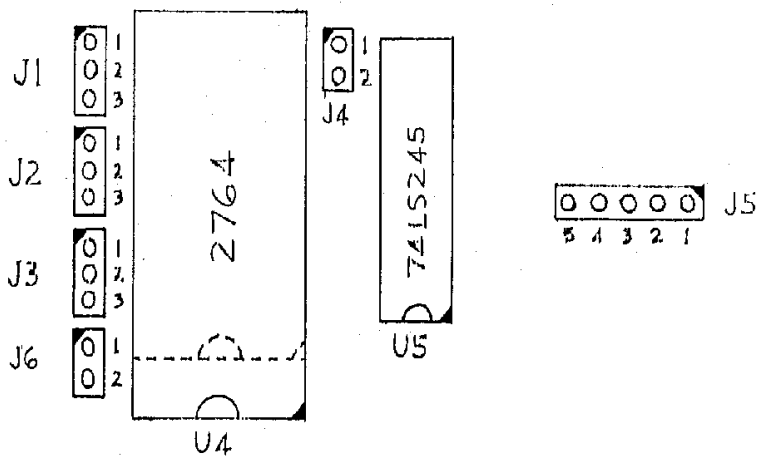
2) Perform step #6 in the next section to install the ROM chip.

3) Perform steps #7 and #9 in the next section to re-install the board in the metal case.

MODIFICATIONS FOR 28 PIN EPROMS, ETC.

The DMC controller, as it leaves the factory, is set up to accept a 24 pin ROM in the ROM socket, and a 24 pin ROM containing Radio Shack Disk BASIC 1.1 is usually installed. The configuration of the ROM socket for the several kinds of ROMs and EPROMs that may be installed is handled by jumper locations J1 to J6, which are hardwired by traces on the printed circuit board (PCB) for 24 pin ROMs:

Location and pin numbering of jumpers J1 to J6



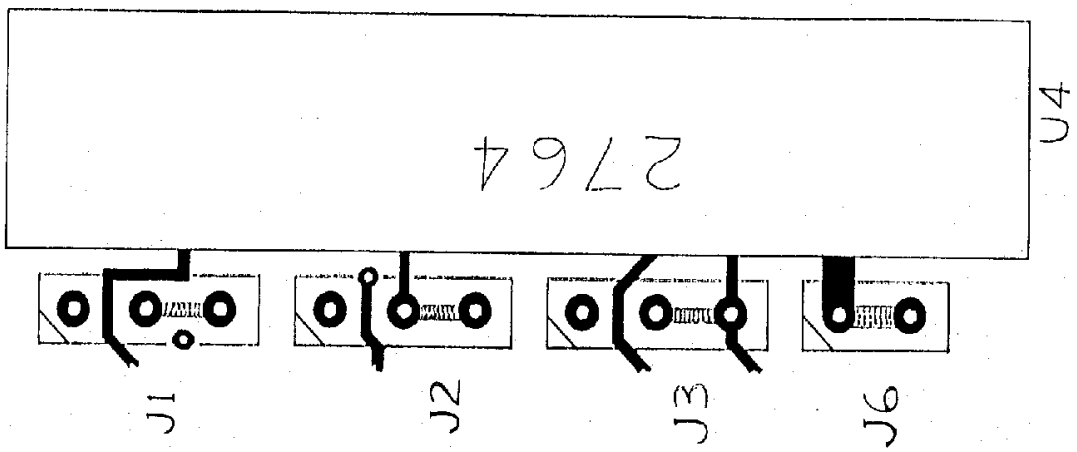
In order to install 28 pin EPROMs (2764, 27128, or 27256) you will need to do three things: a) cut four default jumper traces; b) solder in 6 male header strips; c) install 4 or 5 shorting blocks to set the desired configuration. Once you have made these changes, you will be able to quickly switch between using 24 and 28 pin ROM chips.

NOTE -- Static electricity can damage MOS integrated circuit chips (ICs). Even a static electricity charge too weak to feel can cause problems, and the damage, if it merely weakened the chip, might not cause the chip to fail until weeks or months later. While handling any MOS IC's or circuit boards containing MOS IC's you should ground your body and all tools and work areas that will touch the IC leads or the circuit board. Use a 1 Megohm resistor in series between you and ground to protect yourself against dangerous shocks. Remember -- just because the IC's have been installed in their sockets or soldered on the circuit board doesn't make them immune to static electricity. Better safe than sorry!

1) Remove the circuit board from the metal case. To do this, first remove the 4 sheet metal screws on the sides of the metal case; there are two on each side. The top of the case (it has the venting holes in it) should now lift off the base unit. Remove the four screws fastening each corner of the circuit board to the case bottom, and the circuit board can now be lifted out. Set it onto a grounded work surface as suggested above. Locate (ie. don't loose) the 4 fiber washers -- they should either be stuck onto the bottom of the board at the mounting holes, or stuck onto each standoff in the base unit.

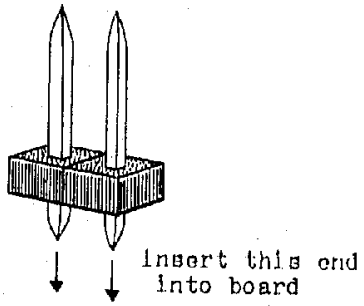
2) If a ROM is currently installed on the board at U4, remove it carefully using a small screwdriver, and set it on the grounded work surface.

3) Using a small, very sharp knife (such as an X-ACTO knife), cut the four jumper traces between pins 2 and 3 of jumpers J1 to J3, and between pins 1 and 2 of J6. In the diagram below they are represented by a light colored bar filled with dots. This is a very tricky operation, as the traces are hard to cut. You must cut deep enough to remove the trace, yet by cutting too deep you can cut other traces deeper inside the board thereby damaging it (remember the board is a 4 layer sandwich). If the knife slips, you can cut yourself, or damage the board. If you haven't done this kind of PCB modification before, you should find a friend who has, or for a reasonable service fee we will do it for you at the factory.



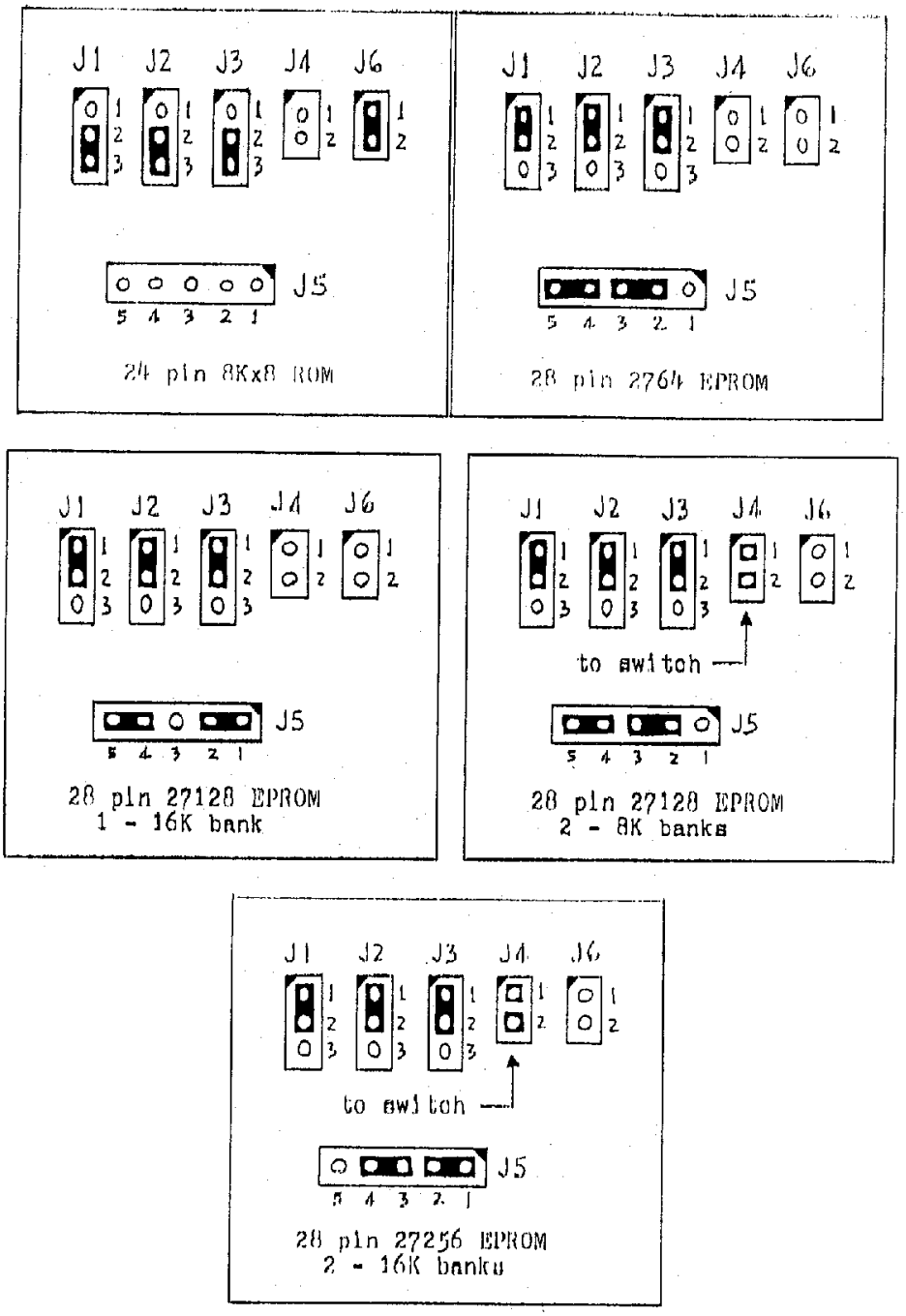
Make sure you blow or brush off of the board all of the bits of traces you just cut away, as they could cause the board to malfunction by creating a short circuit.

4) Now install and solder the 6 male header strips (three 3-pin, two 2-pin, one 5-pin). These can be cut or snapped from a single long strip. Check with your local Radio Shack store for part number 276-1658. If you can't find these locally, one mailorder source is JDR Microdevices, whose ads appear in various magazines including BYTE. Ask for their "1x40 straight lead snappable header". The shorter end of the pins is inserted into the top side of the PCB, with the longer end sticking up. If you haven't done much soldering on PC boards before, this is best left to someone who is experienced.



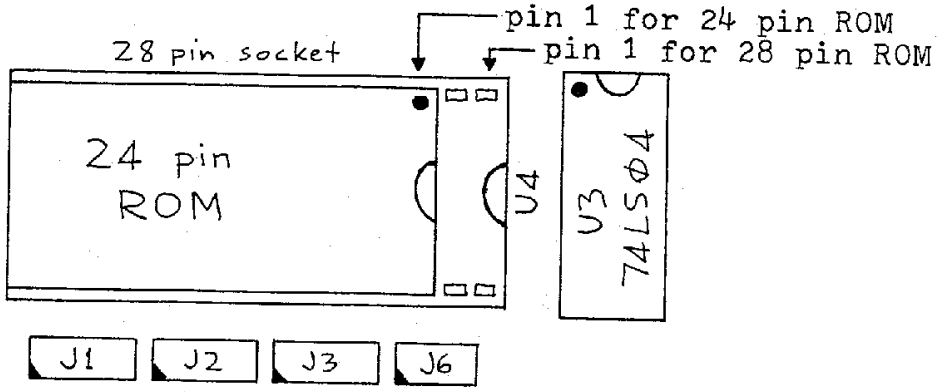
5) Now slip shorting blocks (also available from JDR Microdevices) onto the pins of the header strips using the diagrams below as a guide.

= shorting block installed



6) Install the ROM or EPROM chip. Be very careful to orient the chip properly. Look for a dot or dimple on the top of the chip at one of the corners -- this indicates pin #1. Or, look for a semi-circular indentation at one end -- this is the end that has pin #1. If using a 24 pin ROM, make sure you "lower justify" it in the 28 pin socket, as shown below. After the chip has been inserted into the socket, check that all pins went into the socket OK, that none are bent or curled underneath the chip's body.

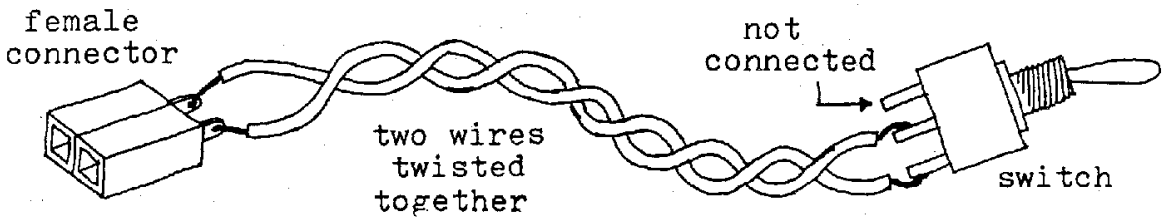
(See diagram next page)



7) Re-install the circuit board into the metal case. First make sure that the four fiber washers are glued to either the bottom of the board at the mounting holes, or onto the tops of each of the standoffs in the case bottom. Place the PC board into the case bottom, so that the computer end of the board (the end with the 40 pin connector) sticks through the wide slot of the case. The 34 pin connector will just barely stick out of its slot (the end with the 1/4" round hole). Install the four screws that hold the board to the case. Be careful not to cross-thread them. Don't put the metal case cover on yet.

8) If you will be implementing the dual-DOS capability of the DMC controller, you will need:
   1 - SPDT toggle switch designed for panel mounting in a 1/4" hole
   1 - 2 pin female connector (for .025" square pins, .100" centers)
   2 - 7" lengths of insulated wire

   Assemble as per the diagram below. Mount the switch using the 1/4" round hole provided. The female connector at the other end plugs into the male header at J4 (it doesn't matter which is pin 1).



(note - some of the parts mentioned above are also available via mailorder from Active Components. Refer to their ads in Radio-Electronics magazine, or phone 1-800-343-0874 in U.S.A., 1-800-361-5884 in Canada.)

9) Put the metal cover over the case bottom. The end with the venting slots in it should be at the disk drive end; ie. the slots should be over the 1773 and static RAM chips. Replace the 4 sheet metal screws. Be careful to have the screws use the existing "threads" in the case bottom -- if you make them cut new threads, the screws may not hold as well, and they will generate a lot of small loose metal particles that can cause the board to malfunction if they cause a short circuit. [In any case, after the four screws are in place, shake and/or blow out any loose metal particles.]

UPGRADING THE CACHE RAM TO 32K BYTES

The current version of the DMC controller is shipped with 8K bytes of cache RAM installed. It is a simple matter to upgrade this to 32K bytes by merely replacing the single static RAM chip. Note, however, that the preliminary version of our software will NOT take advantage of the additional memory.

The new memory chip can be obtained from several sources, one of which is Microprocessors Unlimited, who advertise in such magazines as BYTE. Their part number is 43256L-12, which is a CMOS 120 nsec. 32Kx8 static RAM chip.

We advise you to remove the circuit board from the metal case to install the new RAM chip -- the board might bend too much from the pressure exerted to insert the RAM chip, causing damage to the board.

1) Read the warning in the section above regarding static electricity, then follow the instructions in step #1 in that same section to remove the PC board from the metal case.

2) Carefully remove the RAM chip at U23 (it is one of the large 28 pin chips, and is usually marked "4264") using a small screwdriver, and set it on the grounded work surface. (or SRM 2264 c-12)

3) Install the new RAM chip. Be very careful to orient the chip properly. Look for a dot or dimple on the top of the chip at one of the corners -- this indicates pin #1. Or, look for a semi-circular indentation at one end -- this is the end that has pin #1. Pin #1 should be positioned next to capacitor C14 (refer to the Component Layout Diagram elsewhere in this manual. After the chip has been inserted into the socket, check that all pins went into the socket OK, that none are bent or curled underneath the chip's body.

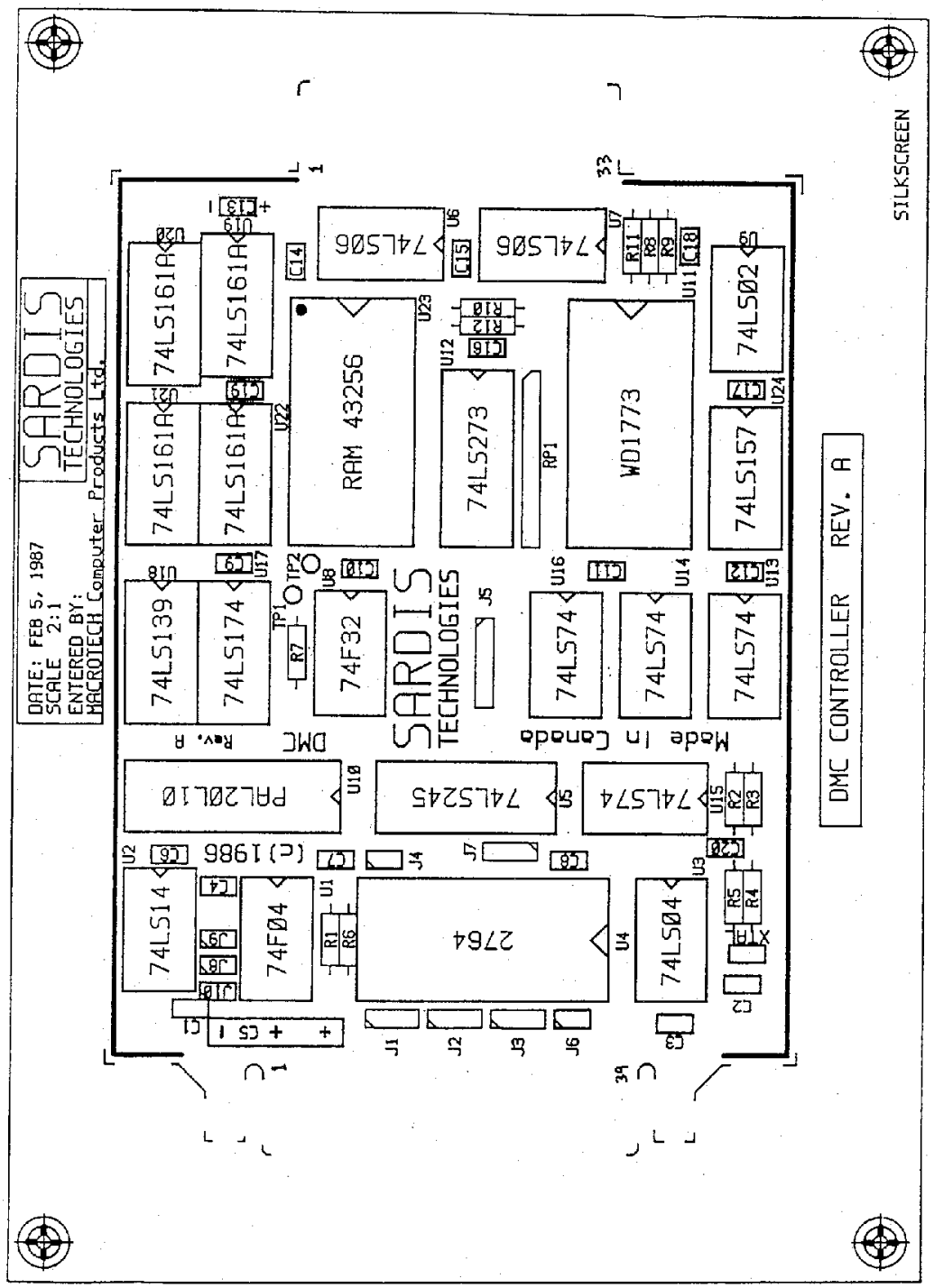
4) Follow steps #7 and #9 in the section above to reassemble the cartridge.

5) Boot up the OS-9 operating system, then run the TESTBUFR diagnostic program supplied with our SDISK-DMC Level I or SDISK3-DMC Level II software packages.

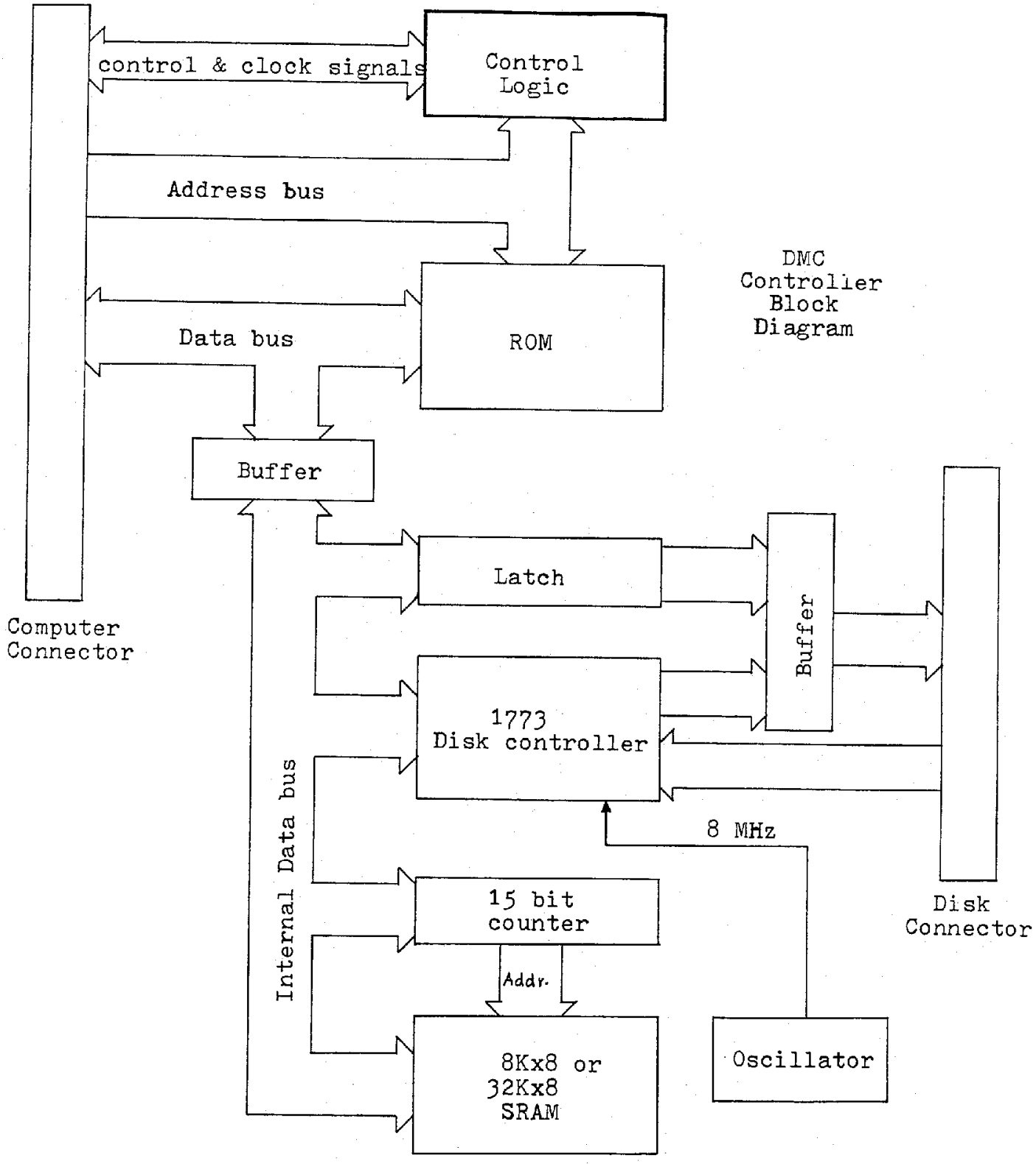
SPECIFICATIONS

SPECIFICATIONS

Component Layout Diagram



BLOCK DIAGRAM



DMC Controller Block Diagram

TROUBLE-SHOOTING AND MAINTENANCE

PARTS LIST

Part #	Capacitors
C1	33 pF miniature ceramic disk capacitor, radial leads
C2	(not installed)
C3	470 pF miniature ceramic disk capacitor, radial leads
C4	33 pF miniature ceramic disk capacitor, radial leads
C5	47 uF electrolytic capacitor, axial leads
C6-C10	.1 uF monolithic bypass capacitor, radial leads
C11,C12	.22 uF monolithic bypass capacitor, radial leads
C13	10 uF tantalum capacitor, radial leads
C14-C20	.1 uF monolithic bypass capacitor, radial leads

Part #	Resistors
R1	470 ohm 1/4 watt resistor
R2	10K ohm 1/4 watt resistor
R3	2.2K ohm 1/4 watt resistor
R4,R5	1K ohm 1/4 watt resistor
R6	470 ohm 1/4 watt resistor
R7,R8	10K ohm 1/4 watt resistor
R9	470 ohm 1/4 watt resistor
R10	330 ohm 1/4 watt resistor
R11	470 ohm 1/4 watt resistor
R12	150 ohm 1/4 watt resistor
RP1	10K ohm resistor SIP, 10 pin, 9 bussed resistors

Part #	I.C.'s
U1	74F04
U2	74LS14
U3	74LS04
U4	Radio Shack MXP-0156 DOS 1.1 ROM, or 2764/27128/27256 EPROM
U5	74LS245
U6,U7	7406 or 7416
U8	74F32
U9	74LS02
U10	PAL20L10 (specially programmed for the DMC)
U11	WD1773 floppy disk controller
U12	74LS273
U13-U16	74LS74A
U17	74LS174
U18	74LS139
U19-U22	74LS161A
U23	8Kx8 or 32Kx8 120 nsec. CMOS static RAM (4264, 43256, etc.)
U24	74LS157

Part #	Other parts
Y1	8 MHz crystal
J1-J10	(no parts installed)
TP1,TP2	(no parts installed)

MEMORY MAP

MEMORY MAP

Addr. (hex)	R/W	Register accessed; function
FF40	W	drive latch bit 7 = enable halt mode 6 = select drive 4 or 2nd side 5 = enable double density 4 = enable write precomp 3 = motor on 2 = select drive 3 1 = select drive 2 0 = select drive 1
FF42	W	D0-D3 = bits 8-11 of cache pointer; also clears bits 0-3 of cache pointer
FF44	R	disable DMA mode
FF44	W	1773 command register; also enables DMA write mode
FF46	W	D0-D2 = bits 12-14 of cache pointer; also clears bits 4-7 of cache pointer
FF48	R	1773 status register
FF48	W	1773 command register (non-DMA)
FF49	R/W	1773 track register
FF4A	R/W	1773 sector register
FF4B	R/W	1773 data register
FF4C	W	1773 command register; also enables DMA read mode
FF4E-FF4F	R/W	read/write cache memory; (cache pointer incremented after)

NOTE - all other locations in the $FF40 - $FF5F range are reserved for future use and should not be read or written.

NOTE - once the DMA mode has been enabled (by writing to $FF44 or $FF4C), NONE of the registers in the $FF40-$FF4F range can be accessed -- only the "disable DMA mode" function (read from location $FF44) still functions. After the DMA mode has been disabled, all the registers are accessible again. Remember that if the DMA mode is disabled in the middle of an I/O operation like read sector or write sector, the 1773 controller chip will try to continue, but will eventually generate a "Lost Data" error code. If you disable the DMA mode before the 1773 asserts the *NMI interrupt line to indicate command completion, you should execute the $D0 "force interrupt" command to reset the 1773 chip.

COLOR COMPUTER CONNECTOR - P1

Pin	Function	Source
1	(n/c)	
2	(n/c)	
3	*HALT	Controller
4	*NMI	Controller
5	*RESET	Computer
6	E Clock	Computer
7	Q Clock	Computer
8	(n/c)	
9	+5 volts power	Computer
10	D0	Bi-Directional
11	D1	Bi-Directional
12	D2	Bi-Directional
13	D3	Bi-Directional
14	D4	Bi-Directional
15	D5	Bi-Directional
16	D6	Bi-Directional
17	D7	Bi-Directional
18	R/*W	Computer
19	A0	Computer
20	A1	Computer
21	A2	Computer
22	A3	Computer
23	A4	Computer
24	A5	Computer
25	A6	Computer
26	A7	Computer
27	A8	Computer
28	A9	Computer
29	A10	Computer
30	A11	Computer
31	A12	Computer
32	*CTS	Computer
33	GROUND	Computer
34	GROUND	Computer
35	(n/c)	
36	*SCS	Computer
37	A13	Computer
38	(n/c)	
39	(n/c)	
40	(n/c)	

DISK DRIVE CONNECTOR - P2

Pin	Function	Source
2	(n/c)	
4	(n/c)	
6	(n/c)	
8	*INDEX HOLE	Drive
10	*DRIVE 1	Controller
12	*DRIVE 2	Controller
14	*DRIVE 3	Controller

16	*MOTOR ON	Controller
18	*DIRECTION IN	Controller
20	*STEP	Controller
22	*WRITE DATA	Controller
24	*WRITE GATE	Controller
26	*TRACK 0	Drive
28	*WRITE PROTECT	Drive
30	*READ DATA	Drive
32	*SIDE 1/*DRIVE 4	Controller
34	(n/c)	

All odd numbered pins 1 - 33 are ground.

SAMPLE DRIVERS for standalone machine language programs

The following sample machine language drivers use the new "no halt" mode of disk I/O. They are intended as a guide or starting point, and are not really intended to be used as-is, since they do not use the same control blocks as the Disk BASIC calls, and no provision is made for motor control.

```
                ****************************************
                *    "DRVSAMPL.TXT"          DW120
                * Sample routines to read/write using the
                *  "no halt" mode of the DMC controller
                *
                * (c) 1986 by Sardis Technologies, all rights reserved
                * Last modified   July 14, 1987  3:10 pm
                ****************************************

                *******************************
                *
                * EQUATES
                *
                *******************************
0001  Carry   EQU  $01        carry status bit
                *
                * INTERRUPT VECTORS
                *
0109  NMIV    EQU  $0109
                *
                * DISK CONTROLLER ADDRESSES
                *
FF40  FDCDRV  EQU  $FF40      drive/density latch (write only)

FF48  FDCCMD  EQU  $FF48      1773 command register (write only)
                *                - non-DMA mode
FF48  FDCSTA  EQU  $FF48      1773 status register (read only)
FF49  FDCTRK  EQU  $FF49      1773 track register (read/write)
FF4A  FDCSEC  EQU  $FF4A      1773 sector register (read/write)
FF4B  FDCDTA  EQU  $FF4B      1773 data register (read/write)
```

```
                FF44  FDWRIT EQU  $FF44     1773 command register (DMA write)
                FF4C  FDREAD EQU  $FF4C     1773 command register (DMA read)
                FF44  DISDMA EQU  $FF44     read register to disable DMA mode
                FF46  FDPTRH EQU  $FF46     high nibble of MSB of DMA buffer
                      *                       pointer (write only)
                FF42  FDPTRL EQU  $FF42     low nibble of MSB of DMA buffer
                      *                       pointer (write only)
                FF4E  BUFFER EQU  $FF4E     DMA buffer RAM location


7000                  ORG  $7000
                      ************************************
                      *
                      * JUMP TABLE
                      *
                      ************************************
7000 160014     JINIT  LBRA INIT
7003 16013F     JCLOSE LBRA CLOSE
7006 160045     JSEEK  LBRA SEEK
7009 16007D     JREAD  LBRA READ
700C 1600A5     JWRITE LBRA WRITE
700F 16010B     JHOME  LBRA HOME


                      ************************************
                      *
                      * VARIABLE STORAGE
                      *
                      ************************************
7012            STAT   RMB  1          1773 status
7013            READY  RMB  1          not-ready flag (zero = ready)
7014            NMISAV RMB  2          NMI vector save area
7016 00         STEP   FCB  0          $00 = 30 msec., $01 = 20 msec.
                      *                  $02 = 12 msec., $03 = 6 msec.


                      ********************************
                      * INITIALIZE
                      * IN - none
                      * OUT - A,B,X undefined
                      *       Y,U unchanged
                      * ERROR - CC CS;  B 1773 status code
                      ********************************
7017 4F         INIT   CLRA            1st try to disable halt mode
7018 B7FF40            STA  FDCDRV
701B FC010A            LDD  NMIV+1     save existing NMI vector
701E FD7014            STD  NMISAV
7021 308D 7137         LEAX NMISRV,PC  point NMI vector to new
7025 BF010A            STX  NMIV+1     . service routine
7028 B6FF44            LDA  DISDMA     disable DMA mode
702B 4F                CLRA            2nd try to disable halt mode and
702C B7FF40            STA  FDCDRV     . set to SD to disable NMI

702F 86FF              LDA  #$FF       set flag to "not ready"
7031 B77013            STA  READY
7034 86D0              LDA  #$D0       command = force interrupt
7036 B7FF4C            STA  FDREAD     (DMA mode)
7039 8E0FA0            LDX  #4000      timeout after .4 sec. @ .89 MHz
703C 170099            LBSR WAIT
703F C481              ANDB #$81       not-ready/busy?
```

```
7041 2704            BEQ  IN80
7043 1A01            ORCC #Carry     .Y
7045 2006            BRA  IN99

7047 86FF      IN80  LDA  #$FF       default track register to 255
7049 B7FF49          STA  FDCTRK
704C 5F              CLRB
704D 39        IN99  RTS


                     ****************************************
                     * SEEK
                     * SELECT DRIVE, SIDE, DENSITY, PRECOMP, AND
                     *  TURN MOTOR ON, THEN SEEK TO DESIRED TRACK.
                     * IN - A drive/density/side/precomp codes for drive reg.
                     *          (refer to memory map in section ?? for
                     *           explanation of each bit)
                     *      X MSB = track number;  LSB = sector number
                     * OUT - A,B undefined
                     *       X,Y,U unchanged
                     * ERROR - CC CS;  B 1773 status code
                     ****************************************
704E 3410      SEEK  PSHS X
7050 847F            ANDA #$7F       make sure halt mode is not used
7052 8A08            ORA  #$08       make sure motor is on
7054 B7FF40          STA  FDCDRV

7057 1F10            TFR  X,D        transfer track # to Reg A
7059 B1FF49          CMPA FDCTRK     are we already on desired track?
705C 2720            BEQ  SK60       .Y
705E B7FF4B          STA  FDCDTA     .N, seek to it
7061 1700A4          LBSR DELAY
7064 86FF            LDA  #$FF       set flag to "not ready"
7066 B77013          STA  READY
7069 861B            LDA  #$1B       command = seek
706B B87016          EORA STEP       set step rate
706E B7FF4C          STA  FDREAD     (DMA mode)
7071 8E7530          LDX  #30000     timeout after 3.0 sec. @ .89 MHz
7074 8D62            BSR  WAIT
7076 C490            ANDB #$90       not-ready/seek-error?
7078 2704            BEQ  SK60
707A 1A01            ORCC #Carry     .Y, error
707C 2009            BRA  SK99

707E A661      SK60  LDA  1,S        write sector number to 1773
7080 B7FF4A          STA  FDCSEC
7083 170082          LBSR DELAY
7086 5F              CLRB
7087 3590      SK99  PULS X,PC


                     ************************************
                     * READ SECTOR
                     * IN - X address of buffer in main memory to read
                     *        data into
                     *      (SEEK routine must be called immediately prior)
                     * OUT - A,B undefined
                     *       X,Y,U unchanged
                     * ERROR - CC CS;  B 1773 status code
                     ************************************
```

```
7089 3410      READ   PSHS X
708B 170082           LBSR POINTS     set DMA buffer pointer
708E 86FF             LDA  #$FF       set flag to "not ready"
7090 B77013           STA  READY
7093 8684             LDA  #$84       command = read
7095 B7FF4C           STA  FDREAD     (DMA mode)
7098 8E1F40           LDX  #8000      timeout after .8 sec. @ .89 MHz
709B 8D3B             BSR  WAIT
709D C49C             ANDB #$9C       not-ready/RNF/CRC/lost-data?
709F 2704             BEQ  RD40
70A1 1A01             ORCC #Carry     .Y, error
70A3 200D             BRA  RD99

70A5 8D69      RD40   BSR  POINTS     reset DMA buffer pointer
70A7 AEE4             LDX  0,S        get ptr. to buffer in main memory
70A9 5F               CLRB            count = 256 bytes
70AA B6FF4E    RD60   LDA  BUFFER     get byte from DMA RAM buffer
70AD A780             STA  ,X+        store byte into main memory buffer
70AF 5A               DECB
70B0 26F8             BNE  RD60

70B2 3590      RD99   PULS X,PC



                      ****************************************
                      * WRITE SECTOR
                      * IN - X addr. of buffer in main memory to write
                      *         data from
                      *      (SEEK routine must be called immediately prior)
                      * OUT - A,B undefined
                      *       X,Y,U unchanged
                      * ERROR - CC CS;  B 1773 status code
                      ****************************************
70B4 8D5A      WRITE  BSR  POINTS     set DMA buffer pointer
70B6 5F               CLRB            count = 256 bytes
70B7 A680      WR30   LDA  ,X+        get byte from main memory buffer
70B9 B7FF4E           STA  BUFFER     store byte into DMA RAM buffer
70BC 5A               DECB
70BD 26F8             BNE  WR30

70BF 8D4F             BSR  POINTS     reset DMA buffer pointer
70C1 86FF             LDA  #$FF       set flag to "not ready"
70C3 B77013           STA  READY
70C6 86A4             LDA  #$A4       command = write
70C8 B7FF44           STA  FDWRIT     (DMA mode)
70CB 8E1F40           LDX  #8000      timeout after .8 sec. @ .89 MHz
70CE 8D08             BSR  WAIT
70D0 4F               CLRA            clear carry bit
70D1 C4FC             ANDB #$FC       not-ready/write-protect/write-
70D3 2702             BEQ  WR99       . fault/RNF/CRC/lost-data?
70D5 1A01             ORCC #Carry
70D7 39        WR99   RTS
```

```
                ************************************
                * Wait until a DMA FDC operation is completed or
                *  timed-out
                * IN - X timeout value (10000 = 1.0 sec. @ .89 MHz)
                * OUT - B 1773 status code
                *       A,X undefined
                *       Y,U unchanged
                * ERROR - never
                ************************************
          70D8 WAIT    EQU  *
                *
                * LOOP WAITING FOR NMI OR TIMEOUT, WHICHEVER COMES FIRST
                *
70D8 17002D     WT20    LBSR DELAY      (9+55) total loop = 89 cycles
70DB 21FE               BRN  *          (3)
70DD 12                 NOP             (2)
70DE 12                 NOP             (2)
70DF 7D7013             TST  READY      (7) I/O operation completed?
70E2 2720               BEQ  WT65       (3) .Y
70E4 301F               LEAX -1,X       (5) .N, time-out?
70E6 26F0               BNE  WT20       (3) .      N
                *
                * PROCESS TIMEOUT ERROR
                *
70E8 B6FF44             LDA  DISDMA     disable DMA mode
70EB 4F                 CLRA            set single-density to disable NMI
70EC B7FF40             STA  FDCDRV
70EF 12                 NOP
70F0 7D7013             TST  READY
70F3 270C               BEQ  WT50
70F5 86D0               LDA  #$D0       force interrupt to abort command
70F7 B7FF48             STA  FDCCMD
70FA 8D0C               BSR  DELAY
70FC 8D0A               BSR  DELAY
70FE B6FF48             LDA  FDCSTA     read status register to reset 1773
7101 C680       WT50    LDB  #$80       simulate status code = "not ready"
7103 39                 RTS
                *
                * NORMAL EXIT
                *
7104 F67012     WT65    LDB  STAT
7107 39                 RTS


                ****************************
                * DELAY
                *     BSR  DELAY  = (7+55) cycles
                *     LBSR DELAY  = (9+55) cycles
                * IN - none
                * OUT - CC,A,B,X,Y,U unchanged
                * ERROR - never
                ****************************
7108 170000     DELAY   LBSR DELAY2     (9+23)        \
710B 170000     DELAY2  LBSR DELAY3     (9+7) \         > 55
710E 12         DELAY3  NOP             (2)    > 23     |
710F 39                 RTS             (5)   /        /
```

```
7110 C63C     POINTS LDB  #$3C
              ***************************************
              * POINTS/POINTT
              * PRESET DMA BUFFER POINTER TO BEGINNING OF SECTOR
              *  OR TRACK IN BUFFER RAM.
              * IN - B MSB of address (only for POINTT)
              *      DMA mode should be inactive
              * OUT - A,X,Y,U unchanged
              *       B,CC undefined
              * ERROR - never
              ***************************************
7112 F7FF42   POINTT STB  FDPTRL
7115 54              LSRB
7116 54              LSRB
7117 54              LSRB
7118 54              LSRB
7119 F7FF46          STB  FDPTRH
711C 39              RTS


              ***************************************
              * POSITION HEAD TO TRACK 0 (RECALIBRATE)
              * IN - none
              * OUT - A,B,X undefined
              *       Y,U unchanged
              * ERROR - CC CS;  B 1773 status code
              ***************************************
711D 86FF     HOME   LDA  #$FF      set flag to "not ready"
711F B77013          STA  READY
7122 860B            LDA  #$0B      command = restore
7124 B87016          EORA STEP      set step rate
7127 B7FF4C          STA  FDREAD    (DMA mode)
712A 8E7530          LDX  #30000    timeout after 3.0 sec. @ .89 MHz
712D 8DA9            BSR  WAIT
712F 4F              CLRA           clear carry bit
7130 C490            ANDB #$90      not-ready/seek-error?
7132 2702            BEQ  HM99
7134 1A01            ORCC #Carry    .Y, error
7136 39       HM99   RTS


              ****************************
              * NMI SERVICE ROUTINE
              ****************************
7137 B6FF44   NMISRV LDA  DISDMA    deactivate DMA mode
713A 12              NOP
713B F6FF48          LDB  FDCSTA    read 1773 status, reset INTRQ/NMI
713E F77012          STB  STAT      save status code
7141 7F7013          CLR  READY     set flag = "DMA transfer finished"
7144 3B              RTI


              ****************************
              * RESET NMI VECTOR
              ****************************
7145 FC7014   CLOSE  LDD  NMISAV    restore previous NMI vector
7148 FD010A          STD  NMIV+1
714B 39              RTS
```

The following is a reference listing of the new commands and error codes that the Disk BASIC 1.1 ROM adds to those already present in the Extended Color BASIC ROM(s). The first line for each command shows the syntax of the command, while the indented second line gives an example of its use. Refer to Tandy's "Color Computer Disk System -- Owners Manual & Programming Guide" for more details.

Disk BASIC Commands

BACKUP source drive TO destination drive
```
BACKUP 0 TO 1
```

CLOSE # buffer,...
```
CLOSE #1, #2
```

COPY filename1 TO filename2
```
COPY "mouse.dat:0" TO "cat.dat:1"
```

CVN (string variable)
```
X = CVN(S$)
```

DIR drive number
```
DIR0
```

DOS

DRIVE drive number
```
DRIVE 2
```

DSKINI drive number
```
DSKINI0
```

DSKI$ drive number, track, sector, string variable1, string variable2
```
DSKI$ 1, 10, 5, A$, B$
```

DSKO$ drive number, track, sector, string1, string2
```
DSKO$ 1, 3, 9, "data abc", "data xyz"
```

EOF (buffer)
```
IF EOF(2) = -1 THEN CLOSE #2
```

FIELD # buffer, field size AS field name,...
```
FIELD #2, 32 AS N$, 6 AS P$, 15 AS C$
```

FILES buffer number, buffer size
```
FILES 2, 500
```

FREE (drive number)
```
PRINT FREE(1)
```

GET # buffer, record number
```
GET #2, 4
```

INPUT # buffer, variable name,...
```
INPUT #2, S$, T$
```

```
KILL filename
     KILL "fleas/bas:1"

LINE INPUT # buffer, data
     LINE INPUT #2, A$

LOAD filename, R
     LOAD "mortgage/bas:1", R

LOADM flename, offset address
     LOADM "pgrm/bin", 2800

LOC (buffer)
     PRINT LOC(2)

LOF (buffer)
     FOR N = 1 TO LOF(2)

LSET field name = data
     LSET N$ = "George"

MERGE filename, R
     MERGE "subrtn/bas", R

MKN$ (number)
     LSET N$ = MKN$(378510)

OPEN "mode", # buffer, filename, record length
     OPEN "I", #1, "info/dat", 64

PRINT # buffer, data list
     PRINT #2, "AABBCC"

PRINT # buffer, USING format; data list
     PRINT USING #1, "###.##"; 195.50

PUT # buffer, record number
     PUT #1, 3

RENAME old filename TO new filename
     RENAME "afile/bin:1" TO "xfile/bin:1"

RSET field name = data
     RSET A$ = "brass"

RUN filename, R
     RUN "poker/bas", R

SAVE filename, A
     SAVE "bestprog/bas:1", A

SAVEM filename, first address, last address, execution address
     SAVEM "seive/bin:1", &H4800, &H4CFF, &H4820

UNLOAD drive number
     UNLOAD 1

VERIFY ON
VERIFY OFF
```

WRITE # buffer, data list
     WRITE #0, X$, Y$, A

Error Messages

AE	file Already Exists
AO	file is Already Open
BR	Bad Record number
DF	Disk Full
DN	Drive Number or Device Number error
ER	write or input past End of Record
FD	bad File Data
FM	bad File Mode
FN	bad File Name
FO	Field Overflow
FS	bad File Structure
IE	Input past End of file
IO	Input/Output error
NE	disk file not found
NO	file Not Open
OB	Out of Buffer space
VF	VeriFication error
WP	Write Protected

AVAILABILITY OF SCHEMATICS
==========

WARRANTY
==========

USER FEEDBACK
==========

INDEX
==========