



User's Manual

DISK JOCKEY 2D (tm)

MODEL B

REVISION 2

Table of Contents

Introduction	1
Programming Specifications	3
ROM Jump Table	3
Serial I/O	4
Disk I/O	5
ROM Subroutines.	7
Error Bits Recap	12
Diskette Initialization.	13
Utilizing Disk Jockey Firmware	14
Sample Read Routine.	15
Sample Write Routine	17
Disk System Software	19
I/O Connectors P1 & P2	20
Patches for CP/M*.	21
Hardware Level Registers	25
Bank Selection	32
Interrupt Logic.	33
Bootstrap LED Indicator.	34
Phantom Logic.	35
4 MHz Operation.	36
Power-On Jump Logic.	37
Drive Cable Conventions.	38
Serial I/O Switch Settings	39
Concise DIP Switch Reference	40
Parts List	41
Assembly Instructions.	45
Parts Installation	48
Initial Check-out and Power-up	52
Concise Firmware Memory Map.	53
Software listings.	54
Cold Boot Loader	55
CBIOS Drivers for CP/M	57
Disk Jockey 2D Firmware.	64
Schematics	94

FOR TECHNICAL SUPPORT

OR REPAIR SERVICE

CALL

(415) 524-2104

User's Manual

DISK JOCKEY 2/Dtm

INTRODUCTION

The Morrow Designs DISK JOCKEY 2/D Model B (DJ) board features four distinct subsections:

1. A floppy disk controller, capable of reading and writing data in either single density FM code or double density MFM code with write precompensation, which can be connected to any floppy disk drive plug compatible with the Shugart 800/850.
2. A baud rate selectable hardware UART serial interface that allows communication with a terminal device at TTY 20ma current loop or RS-232 levels.
3. Automatic address generation upon reset or power-up which allows a "jump start" to the boot strap program in the ROM contained on the board.
4. Bank select logic which allows the board to be enabled or disabled under software control. This logic also can be programed to force the board to be enabled or disabled during power-on/reset sequences.

The DJ plugs into an S-100 bus slot in a system with an 8080, 8085, or Z80 (1.7MHz - 5MHz) CPU. The controller has a cable connector for attaching a flat cable to the first floppy disk drive, and can control a chain of up to four drives daisy chained on this cable. A second connector on the DJ is provided for attaching a terminal device.

The DJ uses memory mapped I/O. Device registers used to input from and output to the floppy disk and the serial port are accessed from the CPU board of the S-100 system by references to memory addresses. Some registers differ in function depending on whether they are being read or written.

Most users will not wish to use the hardware level registers directly. Instead, they can call standard disk and serial I/O subroutines contained in 1016 bytes of EPROM memory on the DJ board. This EPROM occupies a 1024 byte block of S-100 bus memory address space. A 1024 byte RAM is also provided which is used by the EPROM firmware for the storage of various disk related variables such as the current track number, the current drive number, etc. An exact map of these variables is included at the end of the PROM listings.

Introduction

The actual addresses where the I/O registers, EPROM, and RAM appear are controlled by another PROM, referred to as the address selection PROM. The PROM is supplied with standard addresses burned into it for these registers. If the standard addresses would conflict with some other device on the system bus, a PROM burned with non-standard addresses can be substituted.

The DISK JOCKEY 2/D uses 2048 bytes of memory starting at 340:000 or E000H (standard version). The first 1016 bytes are occupied by EPROM, the next 8 bytes constitute the memory mapped I/O, and the last 1024 bytes contain the RAM buffer.

PROGRAMMING SPECIFICATIONS

ROM JUMP TABLE

Most users will wish to take advantage of the standard I/O subroutines supplied in PROM on the DJ.

The user should branch to the appropriate address in a jump table in the first few words of the system ROM. Since each subroutine ends with a RET instruction, a CALL instruction should be used to branch to the subroutine.

The jump table contains jump instructions to the true address of the utility routines within the ROM. Having a jump table allows the individual routines to be updated and moved around within the ROM without having to change software that calls the routines. Let A represent the address of word 0 of the onboard ROM. In boards with standard address decoding PROMS, A = 340:000Q (E000H). The address to call for the utility routines are then:

ADDRESS	STANDARD VALUE		SYMBOLIC VALUE		FUNCTION
	Octal	Hex			
A	340:000	E000	DBOOT		DOS bootstrap routine
A+3	340:003	E003	TERMIN		Serial input
A+6	340:006	E006	TRMOUT		Serial output
A+9	340:011	E009	TKZERO		Recalibrate (seek to TRK0)
A+12	340:014	E00C	TRKSET		Seek
A+15	340:017	E00F	SETSEC		Select sector
A+18	340:022	E012	SETDMA		Set DMA address
A+21	340:025	E015	DREAD		Read a sector of disk data
A+24	340:030	E018	DWRITE		Write a sector of disk data
A+27	340:033	E01B	SELDRV		Select a disk drive
A+30	340:036	E01E	TPANIC		Test for panic character
A+33	340:041	E021	TSTAT		Serial status input
A+36	340:044	E024	DMAST		Read current DMA address
A+39	340:047	E027	STATUS		Disk status input
A+42	340:052	E02A	DSKERR		Loop to strobe error LED
A+45	340:055	E02D	SETDEN		Set density
A+48	340:060	E030	SETSID		Set side for 2-headed drives

The specific function of each subroutine is described below.

The subroutine upon completion will execute a RET instruction. A disk subroutine that completes normally will return with the carry flag cleared to zero. A disk subroutine that detects an error condition will return with the carry flag set to 1. A program should always test the carry flag after a return from a disk utility subroutine and branch to an appropriate error handling routine if the carry flag is set.

SERIAL I/O

GENERAL

There is a hardware UART on the DJ board along with a crystal controlled baud rate generator. There are sixteen different baud rates available including 12 of the most common. The baud rate of the UART must match the baud rate of the terminal connected to the DJ board in order for the serial interface to function properly.

The UART (Universal Asynchronous Receiver-Transmitter) consists of two independent sections: a transmitter section and a receiver section. Each section has two registers. In the transmitter section one register is loaded by the system bus. The contents of this bus register are transferred to a shift register where start, stop, and (conditionally) parity bits are appended. The transmitted serial data originates from this shift register. Whenever the contents of the system bus register have been transferred to the second shift register the UART sets the TBRE (Transmitter Buffer Register Empty) bit in its status register.

In the receiver section there is a shift register which assembles a parallel data word from the input serial stream after start and stop bits have been removed. When a complete data word has been assembled in this register it is loaded into a second register that is accessible from the system bus. Whenever this bus register is loaded from the receiver shift register the UART sets the DR (Data Ready) bit in its status register.

TERMIN

This subroutine is used to collect input characters from a terminal which is connected to the serial port on the board. The routine waits for the UART to raise the DR bit of its status register. The character is then transferred to the A register and trimmed to seven bits. Reading the UART's data register automatically resets the DR bit. This routine will not return until a character arrives from the terminal.

TRMOUT

This subroutine is used to transmit characters to a terminal that is connected to the serial port on the board. The routine waits until the TBRE bit in the UART's status register is high. When this bit is high, the data in the C register of the CPU is transferred to the UART's system bus register. This automatically resets the TBRE bit.

TPANIC

This subroutine is used to detect the presence of a "panic" character in the input data stream from the terminal. A program which uses this routine must load the C register with the desired "panic" character. If the UART has collected a character (i.e. the DR bit of the UART's status register is high) and it matches the character in the C register, the routine SETS the ZERO flag of the CPU's FLAGS register. On the other hand, the routine will CLEAR this flag if 1) the DR bit is not high or 2) the character in the UART's system bus register does not match the character in the C register.

TSTAT

This subroutine is used to test the condition of the DR bit in the UART's status register. If the DR bit is high, TSTAT will SET the ZERO flag of the CPU's FLAGS register. If the DR bit is low, TSTAT will CLEAR the ZERO flag of the CPU's FLAGS register. The routine does NOT alter the state of the DR bit.

DISK I/O

To understand the significance of the disk utility subroutines, it is necessary to say a few words about how data is organized on the disk.

Information on the disk is organized into 77 concentric tracks. The disk read/write head can be moved to any track by a series of step in or step out commands. A step in command moves the read/write head one track towards the center of the disk. A step out command moves the head one track away from the center of the disk. The numbering of the tracks is arranged so that track zero is the farthest from the center of the disk. One of the responsibilities of the Western Digital 1791 / Fujitsu 8866 controller is to know the current track number over which the read/write head is located and to calculate how many step in or step out commands are necessary to move the head to a new track.

Once the read/write head has been moved to the desired track, the rotation of the disk will move a circle of magnetic material beneath the head. Within this circle of material, data is recorded in distinct regions called sectors. The sector is the smallest amount of information that can be separately read from or written to the disk. There are three different sector formats that IBM currently supports. The table below details the relationship between the size of a sector and the number of sectors that can fit on a single track.

bytes of data per sector sectors per track

SINGLE DENSITY	128	26
	256	15
	512	8
DOUBLE DENSITY	256	26
	512	15
	1024	8

In the header field which preceeds the data field of a sector, the track number, the side, the sector number and the sector length are recorded. During read or write commands, this header is read before data transfers take place. Whenever a seek command is issued which causes the the read/write head to move to a new track the firmware on the DJ board performs a verify which reads this sector header to make sure the head is positioned correctly and to determine if there is any change in the sector length or the density of the recorded information. If there is an error as to the track number, the firmware automatically issues a seek to track zero command to position the head over a known track.

The disk drive has a sensor that reports when the read/write head is physically positioned at track zero. A series of step out commands must be issued by the 1791/8866 controller until this status line becomes active. This operation will always position the head to the same physical track. The seek to track zero command is often called a recalibrate command and is a standard utility subroutine supplied with the disk firmware.

Transferring a sector of disk data between memory and the disk therefore involves the following steps, each corresponding to a subroutine call to the Disk Jockey firmware (with the exception of error checking):

Specify the track number the read/write head should be positioned over during subsequent data transfers between the disk and memory.

Check for error conditions.

Specify the sector number that will be involved in subsequent data transfers between the disk and memory.

Specify the starting memory address of block of data that is to be transfered to or from the disk.

Check for error conditions.

Actually perform the read or write operation.

Check for error conditions.

ROM SUBROUTINES

- TRKSET - The value in the C register of the CPU specifies what track the read/write head will be positioned over when the next disk read or disk write operation is issued. A bounds check is made for a value greater than or equal to zero and less than or equal to 76. If the value in the C register is within these bounds, the contents of the C register is written into the RAM location TRACK. Otherwise no action is taken, the carry flag is set and the subroutine returns to the calling program.
- SECTOR - The value in the C register of the CPU specifies what sector will be involved in the next disk read or write operation. If the C register contains a zero, the carry flag is set and the routine returns immediately. If the C register is non-zero, the low order five bits are transferred to the RAM location SECTOR, the carry flag is cleared and the routine returns to the calling program. Just prior to a disk transfer operation a comparison is made between the value in SECTOR and the maximum number of sectors on the track that the transfer is to take place on. If the value in SECTOR exceeds the maximum number of sectors, the transfer operation is aborted and error information is reported.
- SETDMA - During disk transfer operations blocks of data are moved to and from the disk. These blocks can be 128, 256, 512, or 1024 bytes long. The starting address of a data block that will be involved in the next disk transfer operation is specified by the B-C register pair when the SETDMA subroutine is called. Since the disk registers are memory mapped, the firmware has been designed to try to protect them from being written into or read from during disk transfer operations. Accordingly, a bounds check is performed before the DMA address is recorded in the Disk Jockey RAM. If a 1024 byte data transfer to or from the disk would cause memory references to the I/O registers of the disk controller, the carry flag is set and the routine returns with no action taken. If the value of the B-C pair is such that there could not be any memory references to the last eight locations of the Disk Jockey ROM during a subsequent disk operation, the contents of the B-C pair are written into the memory location of the Disk Jockey RAM specified by the label DMAADR. The carry flag is cleared and the routine ends.

Programming Specification - ROM Subroutines

- SELDRV** - The value of the C register determines which of 4 disk drives will be selected for the next disk transfer operation. Accordingly, the data in C is trimmed to the low order two bits and stored in the RAM location DISK. The carry flag is cleared and the routine returns to the calling program.
- SETSID** - Double sided floppy disk drives have two read/write heads so that information can be stored and retrieved from both sides of the diskette. The two heads are positioned so that they are both on the same track one directly below the other. They also share common read/write electronics. Therefore only one of these heads can be selected at a time. Bit 0 of the C register is used to select which of the two heads on a double sided drive will be used during the next disk transfer operation. A zero in bit 0 will select the bottom head and a 1 will select the top head. Selecting a side and selecting a disk are independent operations. If side zero is selected then regardless of the disk selected, side zero will always be accessed until SETSID is called. Finally, if the selected disk is single sided, side zero will always be selected regardless of the results of the SETSID routine.
- SETDEN** - The 1791/8866 Floppy Disk Controller operates in two modes: single density FM (Frequency Modulation) mode or double density MFM (Modified Frequency Modulation) mode. Bit 0 of the C register determines what density the 1791/8866 will operate in when the next disk transfer operation is initiated (0=single,1=double). Care must be exercised in the use of this routine. Under certain conditions, if the density is changed in between disk transfers that occur on the same track, the micro-program that the 1791/8866 controller executes could fall into an error loop from which it could not recover. In such a case the system would have to be reset before further disk operations could be performed. The density mode of the 1791/8866 can safely be changed when a subsequent disk transfer operation will occur on a different track than the last. It should be noted that the firmware of the Disk Jockey has the ability to automatically set the density mode of the 1791/8866. Whenever a new drive is to be selected or whenever the head is not loaded, the Disk Jockey firmware performs a "read header" operation just after positioning the read/write head (if necessary) and just before attempting to perform a disk transfer. This "read header" operation is used to establish the density of the (possibly new) track and to determine the length of the sectors on this track. If the density has not changed from the last "read header" operation or if the calling program has set the density correctly through the use of SETDEN, the process of reading the sector header is slightly faster (by approximately one and a

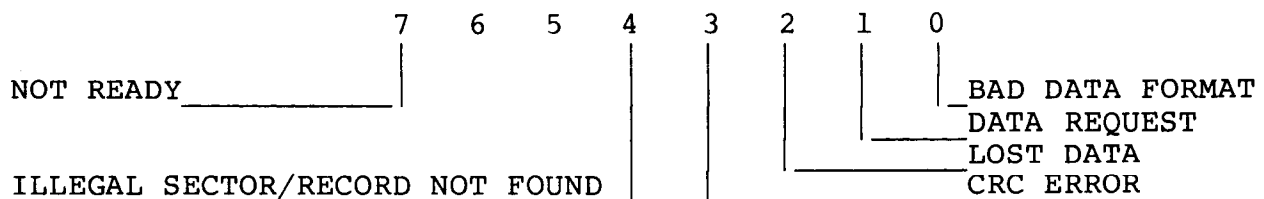
half diskette revolutions) than it would be if the initial assumption concerning the density was wrong.

TKZERO - This subroutine positions the read/write head to the outer-most track of the diskette: track 00. The track zero sensor is used to determine this positioning and no "read header" verify operation is performed. There are several side effects of positioning the head at track zero: (1) a flag is set in the Disk Jockey RAM to force a "read header" density/position verify operation prior to the next disk transfer operation and (2) the mode of the 1791/8866 controller will be forced to single density as long as disk transfer operations occur on track zero. All IBM compatible diskettes have track zero formatted in single density and condition (2) above relieves the system software of the burden of conditionally changing density every time the head is moved to track zero. If the rest of the disk is recorded in double density, the Disk Jockey firmware will automatically switch back to double density when the head is moved away from track zero without the intervention of external software.

READ - This subroutine transfers information from the diskette to memory. The first task is to select the proper disk drive. If the new drive is not the same as the current drive, the load head time-out flag is set and the current drive is updated to be the new drive. Next, the "head loaded" flag is tested. If the head is not loaded or if the current drive was not the same as the new drive, the head load time-out flag is set. The firmware then merges the drive select bits with the head select bit and physically selects a drive, loads the head(s), and selects a side (if the drive is double sided). If the head load time-out bit is set, a 40 millisecond delay occurs to allow for the head to settle after loading. Next the "ready" line from the drive is tested. If the drive is not ready, the head is unloaded and the routine returns to the calling program with the carry bit set and an 80H in the A register. If the drive is ready, the head is positioned in accordance with the most recent seek operation. Head motion (including a head load) or a change of disk drive will cause the firmware to verify the track position by doing a "read header" operation. The correct density of the track is also determined during this operation and the density mode is changed if necessary. If the 1791/8866 controller cannot read the header information in either density, its status is copied into the CPU's A register, the head of the drive is positioned over track zero, and the operation is terminated with the carry set. When the Disk Jockey firmware positions the head to a new track, it reads a header both to determine the proper density and to find out the length and number of the sectors on the new track. The DJ RAM location SECLN is updated

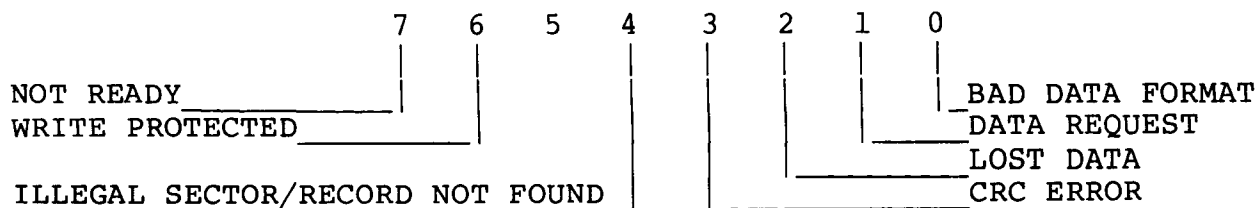
during read header operations and contains encoded data that determines both the number and the size of sectors on the current track. After (possibly) positioning the head the firmware takes the sector address determined by the most recent set sector operation and compares it to the total number of sectors on the current track. If the desired sector is too large, the carry flag is set and the routine returns with a 10H in the A register. If the value is acceptable, the data from this sector is transferred to memory starting at the address specified by the most recent set DMA operation. The length of this transfer is determined by the length of the sectors on the current track. The last two bytes of data on the sector are not read into memory. These are the CRC check sum bytes and are used to detect data transfer errors. The 1791/8866 chip processes these bytes and then updates its status register. The last operation that the routine performs is to place the status information in the A register and conditionally set the carry flag. The details of these status bits are illustrated below.

"DREAD" REGISTER A ERROR BITS



DWRITE - The flow of logic for this routine is exactly the same as described above in the read data operation up to the point where the information transfer is to take place. If all the conditions for a data transfer as described above are satisfied, a write sector command is issued to the 1791/8866 controller and information is transferred from memory to the disk drive starting at the memory address specified by the most recent DMA operation. This data is written on the sector specified by the most recent set sector operation and the head is positioned over the track specified by the most recent seek operation. As the controller writes data on the disk it is continually computing two CRC check sum bytes. After the last byte of data has been written on the diskette, the two check sum bytes are appended to the sector by the controller for later use when the sector is read back into memory. As with the read operation the controller updates its status register after the last CRC byte has been written on the diskette. These status bits are placed in the A register just before control is returned to the calling program. The carry flag is conditionally set from these bits. The details of this status information can be seen below.

"DWRITE" REGISTER A ERROR BITS

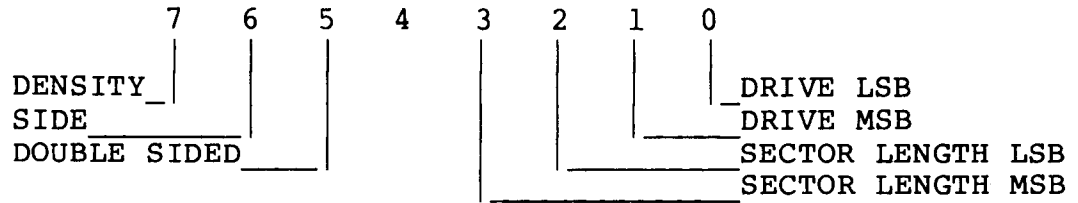


DBOOT - Branching to this routine will initiate a bootstrap load operation from the floppy disk. 128 bytes of data will be read (single density mode) into the first half of the 4th page of the Disk Jockey RAM (normally 344:000Q or E400H). The bootstrap routine terminates with a branch to the first location of this block. Typically sector 1 of track zero will contain another bootstrap program whose job it is to load a Disk Operating System (DOS) such as Disk/ATE or CP/M. If the bootstrap read is not successful, control is passed to the DSKERR utility which is described below. Before sector one is read into memory, various memory locations of the Disk Jockey RAM are initialized. Also DBOOT goes through a several second delay to insure that the system is stable. In order to effect an orderly start-up sequence, DBOOT does not require that the drive have a diskette in place when it is called. If the drive is not ready when DBOOT is called, it falls into a loop that turns on the LED at the top of the controller and slowly pulses the activity light at the front of the drive. This was done so that DBOOT could be started before a diskette was inserted in the drive. When a diskette has been inserted, the door should be closed just AFTER the activity light has been pulsed.

DMAST - This subroutine loads the B-C register pair with the current value of the DMA address recorded in the Disk Jockey RAM.

STATUS - This subroutine loads the B register with the sector number involved in the last disk transfer operation. It loads the C register with the track number the head is currently positioned over. Finally, it loads the A register with a bit pattern indicating the drive involved in the last disk transfer operation, the length of the sectors on the current track, the side specified by the last SETSID call, the density of the data during the most recent disk transfer operation, and whether the drive selected during the most recent disk operation was double sided WITH double sided media in place. The details of how this information is encoded in the A register is presented below.

A REGISTER BIT PATTERN



DRIVE MSB	DRIVE LSB	DRIVE NO.
0	0	DRIVE A
0	1	DRIVE B
1	0	DRIVE C
1	1	DRIVE D

SIDE BIT	SIDE SELECTED
0	SIDE 0
1	SIDE 1

SECTOR LENGTH MSB	SECTOR LENGTH LSB	SECTOR LENGTH	DENSITY
0	0	128	SINGLE
0	1	256	DOUBLE
1	0	512	DOUBLE
1	1	1024	DOUBLE

DENSITY BIT	
0	SINGLE
1	DOUBLE

DOUBLE SIDED = 1 Indicates double sided drive and diskette

DSKERR - Calling this routine will put the CPU into a loop which will cause the LED (Light Emitting Diode) at the top left portion of the controller board to flash on and off at intervals of about a second. This routine takes no parameters and will not return-- its primary usefulness is to indicate when a hard error has occurred during the bootstrap load operation.

RECAP OF REGISTER A ERROR BITS

"SETDMA"	7	6	5	4	3	2	1	0	BIT
DMA ADDRESS SET TO DJ I/O SPACE									

Register A Error Codes

"DREAD"	7	6	5	4	3	2	1	0	BIT
NOT READY									
ILLEGAL DMA ADDRESS									
ILLEGAL SECTOR/RECORD NOT FOUND									
CRC ERROR									
LOST DATA									
DATA REQUEST									
BAD DATA FORMAT									

"DWRITE"	7	6	5	4	3	2	1	0	BIT
NOT READY									
WRITE PROTECTED									
ILLEGAL DMA ADDR									
ILLEGAL SECTOR/RECORD NOT FOUND									
CRC ERROR									
LOST DATA									
DATA REQUEST									
BAD DATA FORMAT									

DISKETTE INITIALIZATION

Before a new diskette can be successfully used, it must be initialized. Most diskettes are sold pre-initialized. However, it is sometimes necessary to reinitialize a diskette. The process of initializing a diskette involves writing the header field of every sector of every track onto the diskette. None of the subroutines described in the section above can be used to write these header fields. This is a safety measure to ensure that an erroneous branch to the firmware EPROM cannot re-initialize a diskette, destroying all the data recorded on it. The initialization function for diskettes is typically provided by a command included in the Disk Operating System. CP/M diskettes furnished by Morrow Designs contain a command called FORMT# to allow the user to format diskettes in any of the four IBM compatible formats.

UTILIZING DISK JOCKEY FIRMWARE

Data transfers to and from the disk must be preceded by calls to certain Disk Jockey routines. The function of these routines is to set up parameters that will be used during the transfer. The following procedure is suggested:

- 1) Select the drive to be involved in the transfer. This is accomplished by calling the routine "SELDRV" with the proper drive number in register C. The drive need not be selected before every transfer. A drive once selected will remain selected until another drive is specified. For 2-headed drives, the side of a drive should be specified by calling the SETSID routine with the desired side number in the C register.
- 2) If the drive has not been accessed before, the read/write head of the drive is in an unknown position. To initialize the drive a call should be made to "TKZERO" in order to bring the head to track zero.
- 3) Set the DMA address. This involves calling the routine "SETDMA" with the correct value in the B-C register pair. It is not necessary to set the DMA address before every data transfer. If data is always being read into the same area of memory, then only one "SETDMA" call need be made.
- 4) Set the read/write head over the desired track. This involves a call to "TRKSET" with the desired track number in register C. It is only necessary to call the "TRKSET" routine when changing tracks. If the data transfer involves the same track as the previous transfer then no call to "TRKSET" should be performed.
- 5) Set the desired sector number. The sector can be set by calling "SETSEC" with the correct sector number in register C. If the sector has not changed since the previous "SETSEC" call, as with a read-modify-write sequence, then this routine may be skipped.
- 6) Read or write the desired sector. The controller can now be commanded to read or write to the disk by calling "DREAD" or "DWRITE".

The order in which these operations occur is not important with the exception that the "DREAD" or "DWRITE" routine must be called last.

Suppose sectors 5, 6, 7 and 8 of track 12, drive 1 are to be read to or from memory starting a location 7:000Q (700H). The following programs will do this:

Utilizing Disk Jockey Firmware

Example of Disk Read

001:000 061 356 346	1	READ	LXI SP,200H	set up the stack
001:003 257	2		XRA A	select drive A
001:004 117	3		MOV C,A	
001:005 315 363 341	4		CALL SELDRV	
001:010 315 362 341	5		CALL TKZERO	recalibrate the head
001:013 016 014	6		MVI C,12	seek the head to
001:015 315 313 342	7		CALL TRKSET	track 12
001:020 001 005 004	8		LXI B,4:005Q	sector count&number
001:023 305	9		PUSH B	save sector cnt&num
001:024 001 000 160	10		LXI B,7000H	set up read address
001:027 315 011 342	11	LOOP	CALL SETDMA	
001:032 301	12		POP B	restore sect to read
001:033 305	13		PUSH B	
001:034 315 166 342	14		CALL SETSEC	set up sect to read
001:037 315 042 342	15		CALL DREAD	read the sector
001:042 332 070 001	16		JC ERROR	test for error
001:045 301	17		POP B	restore sect cnt&num
001:046 005	18		DCR B	update count
001:047 312 073 001	19		JZ DONE	
001:052 014	20		INR C	update sector number
001:053 305	21		PUSH B	save count&number
001:054 315 352 341	22		CALL DMAST	dma address into B-C
001:057 041 000 001	23		LXI H,100H	add sector size to
001:062 011	24		DAD B	current address
001:063 345	25		PUSH H	new address into B-C
001:064 301	26		POP B	
001:065 303 027 001	27		JMP LOOP	continue reading
001:070 303 070 001	28	ERROR	JMP ERROR	error stop
001:073 303 073 001	29	DONE	JMP DONE	

Utilizing Disk Jockey Firmware

Example of Disk Read

0100	31 EE E6	1	READ	LXI	SP, 200H	set up the stack
0103	AF	2		XRA	A	select drive A
0104	4F	3		MOV	C, A	
0105	CD F3 E1	4		CALL	SELDRV	
0108	CD F2 E1	5		CALL	TKZERO	recalibrate the head
010B	0E 0C	6		MVI	C, 12	seek the head to
010D	CD CB E2	7		CALL	TRKSET	track 12
0110	01 05 04	8		LXI	B, 4:005Q	sector count&number
0113	C5	9		PUSH	B	save sector cnt&num
0114	01 00 70	10		LXI	B, 7000H	set up read address
0117	CD 09 E2	11	LOOP	CALL	SETDMA	
011A	C1	12		POP	B	restore sect to read
011B	C5	13		PUSH	B	
011C	CD 76 E2	14		CALL	SETSEC	set up sect to read
011F	CD 22 E2	15		CALL	DREAD	read the sector
0122	DA 38 01	16		JC	ERROR	test for error
0125	C1	17		POP	B	restore sect cnt&num
0126	05	18		DCR	B	update count
0127	CA 3B 01	19		JZ	DONE	
012A	0C	20		INR	C	update sector number
012B	C5	21		PUSH	B	save count&number
012C	CD EA E1	22		CALL	DMAST	dma address into B-C
012F	21 00 01	23		LXI	H, 100H	add sector size to
0132	09	24		DAD	B	current address
0133	E5	25		PUSH	H	new address into B-C
0134	C1	26		POP	B	
0135	C3 17 01	27		JMP	LOOP	continue reading
0138	C3 38 01	28	ERROR	JMP	ERROR	error stop
013B	C3 3B 01	29	DONE	JMP	DONE	

Utilizing Disk Jockey Firmware

WRITE:

The following program writes from memory starting at 200:000Q (8000H) onto tracks 4,5, and 6 of disk drive 1.

001:000	061	356	346	1	WRITE	LXI	SP,200H	set up the stack
001:003	257			2		XRA	A	select drive A
001:004	117			3		MOV	C,A	
001:005	315	363	341	4		CALL	SELDRV	
001:010	315	362	341	5		CALL	TKZERO	recalibrate the head
001:013	001	000	177	6		LXI	B,8000H-100H	set initial adrs.
001:016	315	011	342	7		CALL	SETDMA	
001:021	076	004		8		MVI	A,4	initial track number
001:023	062	112	001	9	TLOOP	STA	TEMP	save track number
001:026	117			10		MOV	C,A	seek to correct trk
001:027	315	313	342	11		CALL	TRKSET	
001:032	001	001	032	12		LXI	B,32:001Q	sector count&number
001:035	305			13	SLOOP	PUSH	B	save sect and count
001:036	315	352	341	14		CALL	DMAST	get current address
001:041	041	000	001	15		LXI	H,100H	update to next sect
001:044	011			16		DAD	B	
001:045	345			17		PUSH	H	move address to B-C
001:046	301			18		POP	B	
001:047	315	011	342	19		CALL	SETDMA	set up new address
001:052	301			20		POP	B	restore sect cnt&num
001:053	305			21		PUSH	B	
001:054	315	166	342	22		CALL	SETSEC	set up next sector
001:057	315	123	342	23		CALL	DWRITE	write the data
001:062	332	107	001	24		JC	ERROR	test for error
001:065	301			25		POP	B	recover sect cnt&num
001:066	014			26		INR	C	update sector
001:067	005			27		DCR	B	update count
001:070	302	035	001	28		JNZ	SLOOP	
001:073	072	112	001	29		LDA	TEMP	get current track
001:076	074			30		INR	A	update track
001:077	376	007		31		CPI	7	check if all done
001:101	302	023	001	32		JNZ	TLOOP	continue to next trk
001:104	303	104	001	33	DONE	JMP	DONE	
001:107	303	107	001	34	ERROR	JMP	ERROR	error exit
001:112	000			35	TEMP	DB	0	track storage
				36				

Utilizing Disk Jockey Firmware

WRITE:

The following program writes from memory starting at 200:000Q (8000H) onto tracks 4,5, and 6 of disk drive 1.

0100	31 EE E6	1	WRITE	LXI	SP,200H	set up the stack
0103	AF	2		XRA	A	select drive A
0104	4F	3		MOV	C,A	
0105	CD F3 E1	4		CALL	SELDRV	
0108	CD F2 E1	5		CALL	TKZERO	recalibrate the head
010B	01 00 7F	6		LXI	B,8000H-100H	set initial adrs.
010E	CD 09 E2	7		CALL	SETDMA	
0111	3E 04	8		MVI	A,4	initial track number
0113	32 4A 01	9	TLOOP	STA	TEMP	save track number
0116	4F	10		MOV	C,A	seek to correct trk
0117	CD CB E2	11		CALL	TRKSET	
011A	01 01 1A	12		LXI	B,32:001Q	sector count&number
011D	C5	13	SLOOP	PUSH	B	save sect and count
011E	CD EA E1	14		CALL	DMAST	get current address
0121	21 00 01	15		LXI	H,100H	update to next sect
0124	09	16		DAD	B	
0125	E5	17		PUSH	H	move address to B-C
0126	C1	18		POP	B	
0127	CD 09 E2	19		CALL	SETDMA	set up new address
012A	C1	20		POP	B	restore sect cnt&num
012B	C5	21		PUSH	B	
012C	CD 76 E2	22		CALL	SETSEC	set up next sector
012F	CD 53 E2	23		CALL	DWRITE	write the data
0132	DA 47 01	24		JC	ERROR	test for error
0135	C1	25		POP	B	recover sect cnt&num
0136	0C	26		INR	C	update sector
0137	05	27		DCR	B	update count
0138	C2 1D 01	28		JNZ	SLOOP	
013B	3A 4A 01	29		LDA	TEMP	get current track
013E	3C	30		INR	A	update track
013F	FE 07	31		CPI	7	check if all done
0141	C2 13 01	32		JNZ	TLOOP	continue to next trk
0144	C3 44 01	33	DONE	JMP	DONE	
0147	C3 47 01	34	ERROR	JMP	ERROR	error exit
014A	00	35	TEMP	DB	0	track storage
		36				

An assembled Disk Jockey 2D is part of a DISCUS 2 system and is also accompanied by a copy of CP/M. The supplied CP/M is tailored to the I/O on the Disk Jockey 2D controller. CP/M expects that a serial TTY/RS-232 terminal is connected to P2 (serial port) of the Disk Jockey. CP/M is supplied on a write protected diskette (notch open) which should be kept that way. DO NOT COVER THE NOTCH ON THE DISKETTE. The system is designed to self load when the disk is placed in drive A and a branch is made to 340:000Q (E000H). The CP/M diskette is accompanied by a series of manuals describing how to back-up a CP/M diskette.

Copies of CP/M which are purchased through Morrow Designs are supplied on a diskette which loads into the system through the use of the bootstrap loader DBOOT. To use DBOOT the system should be turned on and the CPU's program counter should be initialized to 340:000Q (E000H) either from the front panel of the computer or through jump-start logic either on the controller or on some other board in the system. A 2-3 second delay occurs when DBOOT is called so that the system has time to stabilize before the disk is accessed. Power should be applied to the drive(s) that are connected to the Disk Jockey controller at or before the time it is supplied to the CPU. However the system should be given time to stabilize before a diskette is inserted a drive. DBOOT always loads from drive A. If a diskette is not in place when DBOOT is started, the activity light at the front of drive A is slowly pulsed to indicate that the bootstrap loader is waiting for a diskette to be inserted in the drive and the door to be closed. The proper time to close the door is just AFTER the activity light has flashed. Shortly after the door is closed the drive signals the controller that it is ready and a loader program on sector one of track zero is read into the Disk Jockey RAM. When DBOOT is finished, it transfers control to this secondary loader.

I/O CONNECTORS P1 AND P2

Illustrated below are the details of the pin connections of P1 and P2. In both illustrations, the top of the circuit board is to the right of the drawing. The end pins of both connectors are numbered on the silk screen legend of the PC board. Note that all disk interface signals are active low.

			P1		

			50	* *	49 GND
			48	* *	47 GND
			46	* *	45 GND
		-DISK DATA	44	* *	43 GND
		-WRITE PROTECT	42	* *	41 GND
		-TRACK ZERO	40	* *	39 GND
		-WRITE GATE	38	* *	37 GND
		-WRITE DATA	36	* *	35 GND
		-STEP	34	* *	33 GND
		-DIRECTION	32	* *	31 GND
		-DRIVE SELECT 4	30	* *	29 GND
		-DRIVE SELECT 3	28	* *	27 GND
		-DRIVE SELECT 2	26	* *	25 GND
		-DRIVE SELECT 1	24	* *	23 GND
		-SECTOR	22	* *	21 GND
		-READY	20	* *	19 GND
		-INDEX	18	* *	17 GND
		-LOAD HEAD	16	* *	15 GND
		-IN USE	14	* *	13 GND
			12	* *	11 GND
		-TWO SIDED	10	* *	9 GND
			8	* *	7 GND
			6	* *	5 GND
			4	* *	3 GND
			2	* *	1 GND

P2					

RS232 GROUND	*	1			
RS232 INPUT	*	2			
RS232 OUTPUT	*	3			
TTY+ INPUT	*	4			
TTY- INPUT	*	5			
TTY+ OUTPUT	*	6			
TTY- OUTPUT	*	7			

General

This section is included for those users of the Disk Jockey 2D who have purchased a copy of CP/M Vers. 1.4 from a source OTHER than Morrow Designs. Copies of CP/M sold through Morrow Designs have the necessary I/O routines to interface CP/M to the Disk Jockey controller and to the DJ2D's serial I/O facility. These patches will help create a SINGLE DENSITY CP/M diskette--NOT a double density one. Though this may seem of marginal interest at first glance, we would point out that this section, combined with the software listings provided in the back of this manual, constitutes an excellent example of interfacing the Discus 2D to a significant disk operating system.

At the end of this section are two listings which are designed to allow the Disk Jockey to be interfaced with the Digital Research CP/M operating system. This can be done with a minimum of effort.

The first listing is the so called "cold start loader" which is used to bring CP/M in from the disk. It also has code which will allow the user easily to write a modified version of CP/M out on the disk. There is even a small routine which writes the "cold start loader" itself on sector 1 of track 0.

The second listing is CBIOS software (Custom Basic Input-Output System) which is the interface between CP/M and the Disk Jockey controller. The general idea is to key in the cold start loader, use the loader to bring CP/M in from a diskette, enter the CBIOS code and, finally, use the cold start loader to save everything out on a clean diskette.

The "Cold Start Loader"

There are three parts to the cold start loader. LOAD is at address 347:000Q (0E700H) and is designed to read CP/M into memory from location 51:000Q (2900H) to 77:377Q (3FFFH). After loading CP/M, the LOAD routine branches to location 76:000Q (3E00H) which is a routine that initializes several memory locations, prints a sign-on message, and then branches to CP/M proper.

SAVE is at location 347:111Q (0E749H) and is the reverse of LOAD. SAVE writes out on the disk starting at track 0 sector 2 all memory locations between 51:000Q (2900H) and 77:377Q (3FFFH). After performing this operation, SAVE comes to a dynamic halt at STALL 347:133Q (0E75BH).

INTLZ is a short routine which writes locations 347:000Q (0E700H) through 347:177Q (0E77FH) on sector 1 of track 0. Thus, once the cold start loader is keyed into memory, it can save itself at the right location on the disk.

*CP/M is a trademark of Digital Research

CBIOS

The standard version of CP/M is designed to run with the Intel MDS development system and floppy disk interface. Most of the CP/M system software is completely independent of the particular 8080 hardware environment in which it happens to be running. However, there is a certain part which must be tailored to the hardware of the host system. This hardware dependent software is completely contained on pages 76 and 77 of CP/M memory (assuming the standard 16K version). CP/M can be made to run on different hardware by changing the software on pages 76 (3E00H) and 77 (3F00H). The CBIOS software which is supplied with the Disk Jockey is designed to let CP/M run when an eight inch full sized floppy disk is attached to the Disk Jockey controller that is plugged into an S-100 main frame.

Patching CP/M

Before actually performing any of the steps below, the Disk Jockey should be plugged into an S-100 bus mainframe, and an 8" disk drive should be connected to the controller. Be sure to observe correct cable orientation. You should have on hand two diskettes: one with CP/M and a blank one that has been formatted. A copy of CP/M which will run on the Disk Jockey will be constructed on the blank disk before any changes are attempted on the original CP/M disk. As a precaution, the diskette with the CP/M binary should have a write protect notch and this notch should NEVER be covered during the following steps.

Step I:

Plug in the controller. Connect the disk to the controller and turn on the the CPU and the disk drive. Do NOT put a diskette in the drive at this time.

Step II:

Be sure the drive is on and the door is OPEN. Initialize the CPU's program counter to 340:000Q and start the machine. After a several second delay, the LED at the top of the controller should turn on and the activity light (if one is present) on the front of the drive should flash briefly every several seconds. Various memory locations in the Disk Jockey RAM are now initialized and the firmware is ready to perform disk transfer operations. Stop the CPU.

Step III:

Enter the "cold start loader" into memory starting at location 347:000Q (0E700H). The instructions will extend from 347:000Q (E700H) to 347:177Q (0E77FH), filling most of the first half of the last page of RAM on the controller.

PATCHES FOR CP/M*

Step IV:

Set the program counter of the CPU to location 347:142Q (0E762H), but do NOT start the CPU yet.

Step V:

Insert the BLANK diskette into the drive and close the door. Be sure that the diskette is NOT write protected. (An 8" write protected diskette has a notch near the corner of the diskette diagonally opposite the labeled corner.) If this notch is missing or covered, the diskette is not write protected. Be sure the diskette is inserted right side up. On a Disk Jockey system, the label will be on the top. The diskette is inserted in the drive with the label held between the thumb and forefinger.

Step VI:

Start the computer. The drive activity light (if one is present) will come on, the head will load and step out to track 0 unless it is there already. After sixteen revolutions of the diskette, the head will unload and the activity light will go off.

Step VII:

Stop the CPU. It should be in the tight loop JMP DONE -- 303 171 347 octal (C3 79 E7 hex). The cold start loader has been written on sector 1 of track 0.

Step VIII:

Remove the diskette from the drive.

Step IX:

Change location 347:001Q (0E701H) from 000Q (00H) to 133Q (5BH) and change location 347:002 (0E702H) from 76Q (3EH) to 347Q (0E7H).

Step X:

Initialize the program counter of the CPU to 347:000Q (E700H) but do NOT start the machine.

Step XI:

Insert the CP/M diskette and be sure that the write protect notch is not covered. Close the door securely

PATCHES FOR CP/M*

Step XII:

Start the CPU. The head will load and after a second or two the head will step to track 1. Wait for the head to unload and the activity light to go off. CP/M has been loaded into memory between 51:000Q (2900H) and 77:377Q (3EFFH).

Step XIII:

Enter the CBIOS code starting at 76:000Q (3E00H). Be sure to check that the code has been entered correctly.

Step XIV:

Initialize the program counter of the CPU to 347:111Q (E749H) but do NOT start the CPU.

Step XV:

Take the diskette which has the cold start loader on track 0 sector 1 and place it in the drive. Be sure that this diskette is still write enabled (the notch should be covered).

Step XVI:

Start the CPU. The head should load, return to track 0 and write the better part of tracks 0 and 1 before it unloads. After the head unloads, remove the diskette and remove the write enable tab from the diskette. Stop the CPU. The CPU should be executing the JMP STALL instruction -- 303 133 347 octal (C3 5B E7 hex).

Step XVII:

Connect a terminal to the serial port of the Disk Jockey and adjust the baud rate, parity, stop bits, and word length of the terminal and controller so that they match.

Step XVIII:

Inspect the diskette which was removed in step XVI. Be sure that the write protect notch is NOT covered. Insert the diskette in the drive once again. Initialize the CPU's program counter to 340:000Q (E000H) and start the machine. After a few seconds the terminal should print:

16K CP/M VERS/1.4

After a few more seconds the prompt should appear:

A>

A Disk Jockey version of CP/M is now up and running. After this new version of CP/M has been tested (as documented in the CP/M manual), Steps I through XVII can be used to alter the original CP/M diskette if desired.

HARDWARE LEVEL REGISTERS

Users desiring a greater level of control over the floppy disk or serial interface may wish to refer directly to the I/O device registers on the DJ from their 8080 or Z80 program. There are fourteen one-byte registers-- five of them read only, six write only and three read/write. The registers have eight memory addresses on the S-100 bus with a different register being selected during a read operation and a write operation when the addressed register is read only or write only.

The 1791/8866 controller comprises one of the read only registers (status register), one write only register (command register), and all three of the read-write registers (track, sector, and data registers). The uses of these registers will be touched on only briefly here as there is included in the documentation a detailed data sheet describing the way in which the 1791/8866 controller functions.

The 1602 UART comprises two of the read only registers (input data and status registers) and one of the write only registers (output data). As with the 1791/8866, we do not describe these registers in great detail since a data sheet for the 1602 is also included in the documentation.

The 1791/8866 controller has a negative logic data bus. For this reason the internal bidirectional data bus of the DJ board is also negative logic. However, the bus of the 1602 UART is positive logic. This means that when references are made to the UART registers, the signal levels are opposite to what one would normally expect. In practice then, one should always invert data just before it is written into the UART output register; likewise, data read from the UART should be inverted before it is interpreted.

READABLE REGISTERS

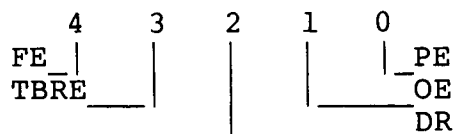
Register 0 - The inverted UART data output register
Location 343:370 (E3F8 hex) standard Disk Jockey:

Data is stored in this register by the UART after it has been assembled from the serial data input stream. When a new character is assembled and transferred to this register, the UART sets the DR (Data Ready) flag. When this register is read by the CPU, the DR flag is reset by the UART hardware.

Register 1 - The inverted UART status register
Location 343:371 (E3F9 hex) standard Disk Jockey

Only the low order five bits of this register have any significance. The meaning of these bits is presented below. The 1602 data sheet should be referred to for a more detailed discussion of these bits. We shall list these signals using their positive logic mnemonics with the understanding that the actual signals read will be the negation of these mnemonics.

INVERTED UART STATUS BITS



FE = Framing Error

TBRE = Transmitter Buffer Register Empty

DR = Data Ready

OE = Overrun Error

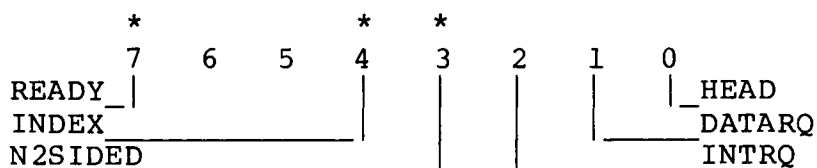
PE = Parity Error

Register 2 - Disk Jockey status register

Location 343:372 (E3FA hex) standard Disk Jockey

This register contains bits that identify the current status of the Disk Jockey and the currently selected drive. Only the six low order bits have any significance in this register. The meanings of these bits are presented below:

DISK JOCKEY STATUS REGISTER



Bits marked with an asterisk reflect the current state of the status lines from the currently selected floppy disk drive. For a detailed specification of these signals see the documentation that accompanys the floppy disk drive. If no drive is currently selected or if the head is not loaded these bits are all high.

READY - This bit is a 1 when the currently selected drive is powered up with a diskette in place and the door closed.

INDEX - This line reflects the status of the INDEX line from the floppy disk drive. It goes to a 1 once per revolution of the diskette.

N2SIDED- This line is a 0 when a double sided drive is connected to the controller AND there is a double sided diskette in place in the drive with the door closed.

HEAD - When this line is a 1 the head of the currently selected floppy disk drive is loaded.

DATARQ - When this line is a 1 the data request line from the 1791/8866 controller is high and the controller is requesting that its data register be read from or written to. When the data register is referenced, this line will change to a 0.

Hardware level registers

INTRQ - The 1791/8866 controller sets this line to a one whenever it has completed a command and is no longer busy. This line is reset by a reference to the command register or the status register of the 1791/8866 controller.

Register 3 - Not currently used
Location 343:373 (E3FB hex) standard Disk Jockey

Register 4 - 1791/8866 controller status register
Location 343:374 (E3FC hex) standard Disk Jockey

This is the status register of the 1791/8866 controller. The meaning of the bit patterns of this register varies depending upon the command that the controller is executing or has executed. See the 1791/8866 data document for a detailed discussion of this register.

WRITE ONLY REGISTERS

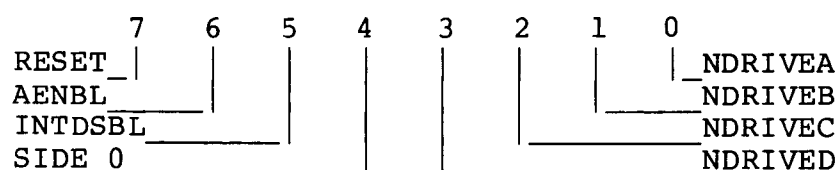
Register 0 - The inverted UART data input register
location 343:370 (E3F8 hex) standard Disk Jockey

Inverted data is stored in this register by the CPU for serial output by the UART. The UART transfers the data from this register to an internal parallel load serial output register where the start bit optional parity bit and the stop bits are appended to the data. Whenever the UART empties register 0, the TBRE status bit is raised to inform the CPU that it is possible to output more data to the UART.

Register 1 - Disk Jockey drive control register
location 343:371 (E3F9 hex) standard Disk Jockey

This is an eight bit register that is used to select one of four possible drives that can be connected to the controller, select side one or side two for double headed drives, enable or disable the interrupt control capabilities of the controller, enable or disable the stall logic of the controller during data accesses to the 1791/8866's data register, and set or clear the master reset pin of the 1791/8866 controller and the VCO oscillator. During power-up and system bus resets, this register is initialized so that it is as if ones had been written in all eight bits. The specific nature and use of the bits in this register is presented below:

DRIVE CONTROL REGISTER



Hardware level registers

- RESET - When a one is stored in this bit, the master reset pin of the 1791/8866 is active and the controller chip is in a reset condition and will not accept any commands. The Voltage Controlled Oscillator of the Phase Lock Loop is also disabled and the Phase Lock Loop will not process any data to produce data windows for the 1791/8866. This bit is used to reinitialize the 1791/8866 in the event that the micro-program in the controller chip becomes confused and gets lost trying to read bad data. When a zero is stored in this bit (after a one value) the VCO of the Phase Lock Loop will properly start and the 1791/8866 will execute a home command and place itself in a state to accept commands.
- AENBL - When the CPU references the 1791/8866's data register during a data transfer, the PREADY line (S-100 bus line 72) is brought low which puts the processor in a wait state. The CPU remains in this state until the 1791/8866 raises its DATA REQUEST line. This mode of operation dispenses with the usual status test during data transfers and makes it possible for the Disk Jockey to run at double density speeds without having to use a DMA channel. However, there are times when the CPU needs access to the data register even though the DATA REQUEST LINE is low and will stay low (just before a seek command is issued, for example). When the AENBL bit is a one, the stall logic that usually governs accesses to the 1791/8866's data register is disabled. This allows the CPU to have access to this register as if it were a normal memory location. However, before the Disk Jockey can move data to or from the floppy disk drive, this bit must be a zero so that the CPU can synchronize its data transfers to the 1791/8866 controller.
- INTDSBL - When this bit is a zero, the interrupt request line of the 1791/8866 controller is enabled to request interrupts on the S-100 system bus. When this bit is a one, no interrupts can be generated by the controller. The user should consult the 1791/8866 data sheet for a thorough understanding of the chip's interrupt request line.
- SIDE 0 - When a double headed drive is connected to the Disk Jockey, a zero in this bit will enable head 1 whenever the drive is selected. A one will enable head 0. If a single headed drive is selected, this bit has no effect on the drive.
- NDRIVED - When this bit is a zero and the head is loaded, the fourth or last drive is selected. A one written in this bit will deselect the last drive.

Hardware level registers

- NDRIVEC - This is the drive select bit for the third drive connected to the Disk Jockey. A zero selects the third drive when the head is loaded while a one deselects the third drive.
- NDRIVEB - The drive select bit for the second drive connected to the Disk Jockey. When the head is loaded, a zero in this bit will select the second drive while a one will deselect it.
- NDRIVEA - The drive select bit for the first drive connected to the Disk Jockey. A zero in this bit will select the first drive when the head is loaded and a one will deselect it.

Only one of the four low order bits of this register should ever be a zero. If more than one of these bits are zero, loading the head will select more than one drive and cause data errors during reads and possible head position errors on seeks.

Register 2 - The Disk Jockey function register

Location 343:372 (E3FA hex) standard Disk Jockey

Only the low order four bits of this register have any significance. Two bits load and unload the read/write head of the drive, one determines the density mode that the 1791/8866 controller operates at, and the last is used to turn on and off the LED at the top of the PC board. During power-up and system bus reset, this register is initialized so that it is as if ones had been written in all four bits. The specific function of the various bits in this register is detailed below:

DISK JOCKEY FUNCTION REGISTER

3	2	1	0
LEDOFF_			SINGLE
HD1	_____		_____ HD0

- LEDOFF - When a one is stored in this bit, the LED at the top of the circuit board is turned off. A zero will turn the LED on.
- SINGLE - When this bit is a one, the DJ board will read and write data to and from the disk in single density. When this bit is a zero, reads and writes are performed in double density.
- HD0,HD1 - These two bits control the loading of the read/write head. Their functional character is detailed in the table below.

Hardware level registers

HD1	HD0	Read/write head function
0	0	head is loaded
0	1	not allowed
1	0	1791/8866 may unload head
1	1	head is unloaded

Register 3 - Not currently used

Location 343:373 (E3FB hex) standard Disk Jockey

Register 4 - 1791/8866 controller command register

Location 343:374 (E3FC hex) standard Disk Jockey

This is the command register of the 1791/8866 controller. There are four different classes of commands and within each class there are a number of separate commands that the controller can execute. See the 1791/8866 data document for a detailed discussion of this register and its use.

READ-WRITE REGISTERS

Register 5 - 1791/8866 track register

Location 343:375 (E3FD hex) standard Disk Jockey

The 1791/8866 controller uses this register as a reference to where the read/write head of the disk drive is positioned. Extreme care should be exercised when writing in this register. If care is not exercised, seek errors may likely occur. See the 1791/8866 data document for a more detailed discussion.

Register 6 - 1791/8866 sector register

Location 343:376 (E3FE hex) standard Disk Jockey

This is the sector register of the 1791/8866 controller. Only one of the commands will cause the 1791/8866 to write in this register. Generally the 1791/8866 uses this register to determine which sector is to be read or written. See the 1791/8866 data document for a more detailed discussion.

Register 7 - 1791/8866 data register

Location 343:377 (E3FF hex) standard Disk Jockey

This is the data register of the 1791/8866 controller. Data is written into this register when the controller is writing to the disk. Data is read from this register when the controller is reading from the disk. The desired track number is also written in this register when seek commands are issued to the controller. As before the 1791/8866 data document should be referred to for a more complete discussion

Hardware level registers

FINAL NOTE

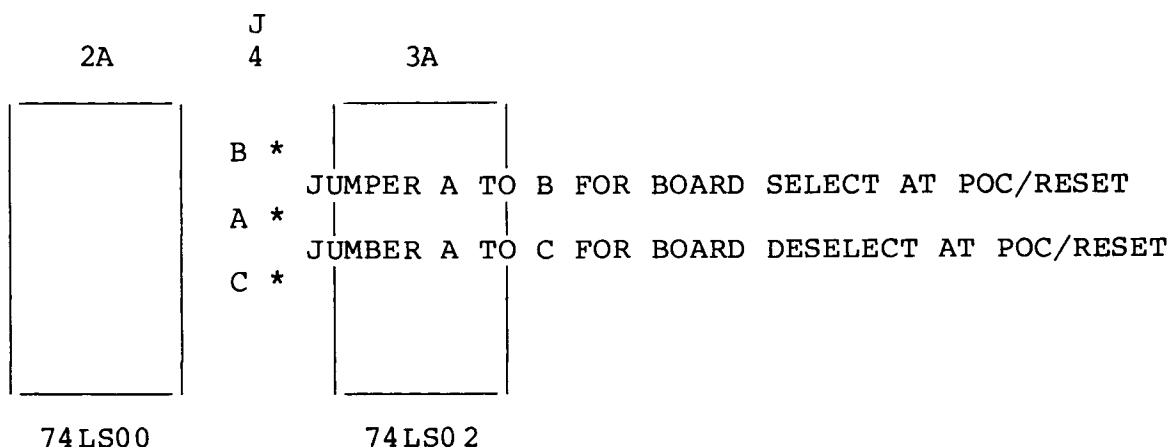
The Disk Jockey firmware contains numerous examples illustrating the use of the hardware registers listed above. A comprehensive study of the two Western Digital data documents along with a careful examination of the Disk Jockey firmware will equip the interested user with enough knowledge to control the disk drive at the hardware level.

BOARD SELECT BY I/O ADDRESS -- BANK SELECTION

The purpose of "Bank Selection" is to allow more memory in a system than the CPU can normally address. This is accomplished by assigning a board not only a memory address somewhere within the 64K range of addressable memory, but also a bit position within a special dedicated I/O port - port 40H (100Q). Port 40H is called the "Bank Select Port" and is used by a wide variety of S-100 hardware manufacturers exclusively for this purpose. With this scheme, it is possible to have as much as 524,288 bytes of memory on the S-100 bus without addressing conflicts.

System software and user programs are growing larger each day and it is clear that memory mapped devices such as the Disk Jockey must exercise care in the way that they use S-100 bus memory space. To make way for the increased need for memory space, the Disk Jockey now implements the bank select port, port 40H, so that the 2K of memory space that the board uses can be assigned to any of eight banks within the extended address space on the bus. Another feature of the board is its ability to select or de-select itself during power-on clear or bus resets.

To implement the bank select logic on the board, the user must decide which bit within port 40H will be used to select and de-select the board. This bit is selected by installing a jumper on the board. A decision must also be made as to whether the board should select or de-select itself when POC* (bus line 99) or PRESET* (bus line 75) is active. This decision is made by the installation of another jumper. The details of these two jumper options are presented below:



Factory assembled boards will be shipped with a jumper installed between A and B so that the board will select itself during POC* or PRESET*. If for some reason this choice is not acceptable to the user, it is easy to remove the jumper and install it between A and C. It is necessary that one of the two jumpers always be installed, even if the board is not to be used in a bank select environment. If the bank select logic is not to be used, the jumper should be between A and B. A final note - both jumpers should never be installed simultaneously.

Bank Select Logic - Bit Selection

The bank select scheme will provide for eight banks of memory each having 64K bytes. These banks are numbered 0 through 7 which correspond to the bit positions in the illustration at the right. The pad just above J3A below should be jumpered to exactly one of the pads to the right. The bit number to the right of the pad will determine the memory bank that the Disk Jockey will reside in. Once this choice is made, the Disk Jockey will be enabled or disabled when the CPU executes an OUT 40H instruction. The pattern in the A register will determine whether the board is selected or not. Suppose, for example, that J3A is connected to bit 7. Then the Disk Jockey will be enabled when the CPU executes an OUT 40H instruction and the A register has a pattern such that bit 7 is a one. The values of the other bits have no influence on whether the board will be selected or not. If bit 7 is a zero, the board will be deselected. Again, the values of the other bits have no influence. However, for the bank select scheme to work properly, when an OUT 40H instruction is executed, usually only one of the bits in the A register should be a one. In this way, only one bank of memory will be selected at one time.

The bank select logic on the Disk Jockey board can be disabled by removing the 25LS2521 IC from position 11C.

11C		12C
	* 0	BIT 0
	* 1	BIT 1
	* 2	BIT 2
	* 3	BIT 3
	* 4	BIT 4
	* 5	BIT 5
	* 6	BIT 6
	* 7	BIT 7
25LS2521	*	74LS273
	J	
	3	
	A	

INTERRUPT LOGIC

Whenever the 1791/8866 disk controller chip finishes an operation such as read sector, seek to a track, seek to track 0, etc., it raises an internal interrupt request flag which is brought to the outside world on pin 39 of the device. This flag can be used to inform external hardware that the chip is ready to execute new tasks. The present version of the Disk Jockey controller buffers this signal and makes provision for the user to connect it to any of the nine different interrupt lines available on the S-100 bus.

Interrupt Logic

Presently there is not a great deal of interrupt driven software available for microcomputer systems. However, this will probably change as the user demand for increased system speed and performance begins to be felt by software vendors. It is also fair to say that interrupt driven operating systems are somewhat more complex and require a great deal more thought to implement than operating systems which are not interrupt driven. Operating systems such as UNIX have been designed with interrupts in mind while operating systems such as CP/M were designed before people seriously considered using classic interrupt techniques in a microcomputing environment.

The Disk Jockey interrupt logic is implemented by installing a jumper at the lower left hand area of the circuit board. The jumper should originate at the open pad just to the left of J1A and should connect to ONLY ONE of the pads below the symbols VI0, VI1, VI2, VI3, VI4, VI5, VI6, VI7, or PINT. Unless there is a vectored interrupt controller on the bus or on the system's CPU board, the jumper connection should be made to PINT. After the interrupt jumper is installed, interrupts from the 1791/8866 can be enabled or disabled by writing a 0 or 1 in bit 5 of the Disk Jockey drive control register (write only register #1). For the details please refer to the section on Hardware Level Registers. The jumper pad layout for installing interrupts on the DJ board are shown below:

```

* J1A

      P
V  V  V  V  V  V  V  V  I
I  I  I  I  I  I  I  I  N
0  1  2  3  4  5  6  7  T
*  *  *  *  *  *  *  *  *
```

BOOT LED

Just to the left of P1, the right angle header connector for the disk drive, is the boot LED. This LED (light emitting diode) will slowly flash on and off if the DBOOT routine cannot load the bootstrap from the diskette. Since the boot routine does not use any of the terminal I/O logic, this LED is helpful in determining whether a go/no-go attempt at bringing up an operating system is due to faulty I/O hardware and/or drivers or due to some other cause-- memory, diskette media, controller, CPU, etc.

BOOTING WITHOUT A DISKETTE

If no diskette has been placed in Drive A and a boot is attempted (as is often the case during a power-on-jump when the system is first powered up), the red activity light at the front of the Drive A will flash on briefly about once every second and the boot LED will turn on without flashing. It is possible to execute a bootstrap load in this mode. Insert a system diskette into Drive A. Do not lower the door, but push the diskette into the drive far enough so that it locks into place (the higher the drive door, the easier for the diskette to lock into place). Wait for the activity light at the front of the drive to flash on and off and, when it goes off, close the drive door. The system will boot the next time the drive activity light goes on.

POWER STABILIZATION

Whenever the bootstrap load DBOOT routine is called, the head on Drive A will not load (as evidenced by the drive activity LED at the front of the drive) for a second or two. There is a built in delay in DBOOT to make sure that all components of the system are stable and have finished any reset processes that may occur when the system encounters an active POC* (negative logic power-on-clear) or PRESET* (negative logic bus reset) signal. This delay precaution is especially important when power is first applied to a system which does a power-on-jump to the controller.

PHANTOM LOGIC

The DJ will respond to the PHANTOM* line (S-100 pin 67) if paddle 6 of switch 1 is placed in the 'on' position. This paddle is the third from the top of the LEFT switch which is at position 5D on the circuit board. The Disk Jockey controller will become de-selected when the PHANTOM* is active (logic zero) if this paddle is on. If this paddle is placed in the 'off' position, the DJ controller will ignore the PHANTOM* line. In order for the Power-on Jump feature of the controller to work on a SOL computer, the PHANTOM* switch must be on.

The DJ can also generate PHANTOM* whenever the prom or ram on the DJ is accessed. This feature can be used to disable other memory boards in the system which may conflict with the memory address of the DJ. To enable this feature install the jumper J2 on the circuit board. With jumper J2 installed the DJ will drive the PHANTOM* line low (active state) whenever the address on the S-100 bus matches the addresses occupied by the DJ. Note that if jumper J2 is installed AND the PHANTOM* enable switch is on the DJ will never become selected. Only one of the PHANTOM* options of the DJ can be used at a time.

4 MHz OPERATION

The Disk Jockey controller has been designed to work at all three of the most common S-100 bus speeds: 2 MHz, 4 MHz, and 5 MHz. However, at bus speeds in excess of 2 MHz, the 2708 EPROM on the board may not function properly unless a wait state is inserted during fetches to this part. The DJ has been designed to automatically insert ONE wait state in bus cycles which read data or instructions from the 2708 EPROM if paddle 7 of switch 1 is in the 'on' position. If this paddle is in the 'off' position no wait states will be generated during fetches from the 2708 EPROM.

Whenever the Disk Jockey is operating in a system that has a CPU clock speed faster than 2 MHz, paddle 7 of switch 1 MUST be in the 'on' position. If the Disk Jockey is operating with a CPU that is running a 2MHz or slower, paddle 7 of switch 1 MUST be in the 'off' position. This paddle is the second from the top of the LEFT switch at location 5D on the circuit board.

The Disk Jockey controller has the ability to generate addresses on the system S-100 bus when power is first applied or when a system reset is active. This address generating ability will force the CPU to branch to the DBOOT routine on the DJ board so that the system will boot an operating system into memory. There are six paddles on switch 1 at board position 5D which control the power-on jump logic of the controller. Paddle 8, at the top of the switch enables or disables the power-on jump circuitry. The logic is enabled if the paddle is in the 'on' position and disabled if the paddle is in the 'off' position. If the logic is disabled, the settings of the other five paddles are not important. If the logic is enabled, the settings of the rest of the paddles informs the CPU of the starting address of the Disk Jockey controller within a 64K region of memory. Since the controller uses 2K of address space which starts on a 2K boundary, it is necessary to specify the 5 high order address bits to affect a branch to the controller. The remaining 5 paddles on switch 1 program these 5 high order address bits. These switches are arranged in ascending order:

Paddle 5	programs	address bit 11	- on for low, off for high
Paddle 4	programs	address bit 12	- on for low, off for high
Paddle 3	programs	address bit 13	- on for low, off for high
Paddle 2	programs	address bit 14	- on for low, off for high
Paddle 1	programs	address bit 15	- on for low, off for high

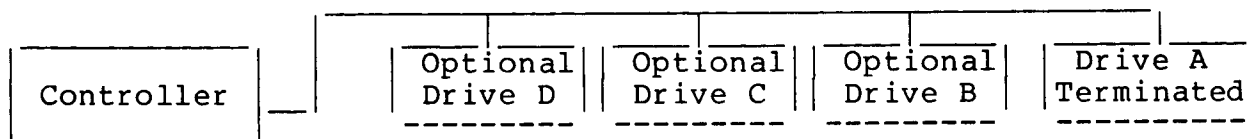
These paddles occupy the lowest five positions on switch 1 at board position 5D. For a standard DJ board located at E000H (340:000Q), paddles 1, 2, and 3 should be off while paddles 4 and 5 should be on. Below a complete table of switch settings is detailed.

POWER-ON JUMP TABLE

JUMP ADDRESS		SWITCH SETTING				
Octal	Hex	SW1-1 (A15)	SW1-2 (A14)	SW1-3 (A13)	SW1-4 (A12)	SW1-5 (A11)
000:000	0000	on	on	on	on	on
010:000	0800	on	on	on	on	off
020:000	1000	on	on	on	off	on
030:000	1800	on	on	on	off	off
040:000	2000	on	on	off	on	on
050:000	2800	on	on	off	on	off
060:000	3000	on	on	off	off	on
070:000	3800	on	on	off	off	off
100:000	4000	on	off	on	on	on
110:000	4800	on	off	on	on	off
120:000	5000	on	off	on	off	on
130:000	5800	on	off	on	off	off
140:000	6000	on	off	off	on	on
150:000	6800	on	off	off	on	off
160:000	7000	on	off	off	off	on
170:000	7800	on	off	off	off	off
200:000	8000	off	on	on	on	on
210:000	8800	off	on	on	on	off
220:000	9000	off	on	on	off	on
230:000	9800	off	on	on	off	off
240:000	A000	off	on	off	on	on
250:000	A800	off	on	off	on	off
260:000	B000	off	on	off	off	on
270:000	B800	off	on	off	off	off
300:000	C000	off	off	on	on	on
310:000	C800	off	off	on	on	off
320:000	D000	off	off	on	off	on
330:000	D800	off	off	on	off	off
340:000	E000	off	off	off	on	on
350:000	E800	off	off	off	on	off
360:000	F000	off	off	off	off	on
370:000	F800	off	off	off	off	off

CABLE CONNECTIONS

Drives on Discus systems are connected in daisy chain fashion to the controller board, as illustrated below.



As can be seen from the above figure, Drive A is located at one end of the cable and is the only terminated drive on the cable. The location of any additional drives on the cable is not important as long as they are not at the end of the cable. Again, extra drives are not terminated.

Aside from termination, the only physical difference between an "A" and a "B" drive, or between any two differently addressed drives, is the jumper strapping on the PC board of the drives. Strapping a drive for termination and drive selection is documented in the manual which accompanies the drive.

Four different daisy chain cables are available for one, two, three or four drive systems. A daisy chain cable is simply a parallel cable. Not all available connectors on a multiple drive cable need be filled for the system to function. Also, a dual system with drives addressed, say, as "A" and "C" would work fine as long as the operator remembered to refer to the second drive as "C" rather than "B". In other words, the absence of a "B" drive in no way "locks out" the "C" and "D" drives.

The following rule applies to all cable configurations supplied by Morrow Designs:

The 50 pin flat ribbon cable provided with the Discus system should be connected to the Disk Jockey controller board so that the cable extends out over the solder side of the PC board-- not the component side.

Whichever end of the 50 pin flat ribbon cable is chosen to plug into the controller board, that side of the cable which is on the LEFT (closer to the heat sink) as it connects to the controller should be UP as it connects to each and every drive on the system. Thus, P1 pin 50 on the DJ controller board should come in to each disk drive via the top part of the male 50 pin connector attached to the cabinet of each drive. If the LED on the front of the drive comes on upon power up, the cable is on backwards and should be reversed. The LED on the front of the drive should light up only when a command has been issued to load the head.

Any visual "key" such as an arrow or triangle on a connector should be used solely as an aid in implementing the connection scheme described above.

SERIAL I/O SWITCH SETTINGS

BAUD RATE SELECTION

Paddles 1 to 4 of Switch 2 at the right side of the DJ control the baud rate for the 1602 UART. Sixteen separate baud rates, ranging from 50 to 19,200, are available. The following table lists all possible switch settings for baud rate selection.

BAUD RATE SWITCH SETTINGS

SW2-1	SW2-2	SW2-3	SW2-4	BAUD RATE
on	on	on	on	50
on	on	on	off	75
on	on	off	on	110
on	on	off	off	134.5
on	off	on	on	150
on	off	on	off	300
on	off	off	on	600
on	off	off	off	1200
off	on	on	on	1800
off	on	on	off	2000
off	on	off	on	2400
off	on	off	off	3600
off	off	on	on	4800
off	off	on	off	7200
off	off	off	on	9600
off	off	off	off	19200

WORD LENGTH

Paddle 5 of Switch 2 controls data word length selection for the 1602 UART. Placing paddle 5 in the "on" position sets the word length to 7 bits, while "off" fixes the word length to 8 bits. The table below gives the word length selection settings for the DJ.

WORD LENGTH SELECTION

SW2-5	WORD LENGTH
"on"	7 BITS
"off"	8 BITS

STOP BIT COUNT

SW2-6 controls the number of stop bits, either one or two, which the UART sends after each data word. The "off" position will set the device to two stop bits, and the "on" position to one.

Serial I/O Switch Settings

Most devices are extremely tolerant concerning stop bit setting. As a general rule, if a device fails to communicate with the Disk Jockey, it is not because the stop bit setting is incorrect.

STOP BIT COUNT SELECTION

SW2-6	STOP BIT COUNT
"on"	1 STOP BIT
"off"	2 STOP BITS

PARITY

If paddle 8 of switch 2 is in the "off" position, the UART will not generate any parity bits at the end of the serial data word. If the paddle is in the "on" position, refer to the table below for the proper parity setting via paddle 7.

PARITY SWITCH SETTING

SW2-7	PARITY
"on"	ODD PARITY
"off"	EVEN PARITY

FAST REFERENCE FOR DJ2D DIP SWITCHES

Power-on-jump Switch

on	off	
	8	-"on" enables POJ
	7	-"on" for 4 MHz
	6	-"on" for PHANTOM
SW1	5	-ADDR 11 "on" for 0 address bits
	4	-ADDR 12
	3	-ADDR 13 -
	2	-ADDR 14 "off" for 1 address bits
	1	-ADDR 15

5D

UART Switch

on	off	
	8	-"on"= parity/"off"=no
	7	-"on"= odd/"off"=even
	6	-"on"= 1 stop bits
SW2	5	-"on"= 7 bits/"off"=8
	4	-A low order bit
	3	-B Baud Rate Selection
	2	-C "on" = 0 bit
	1	-D high order bit

13C

DJ/2D MODEL B PARTS LIST

[]	1	5" x 10" printed circuit board w/solder mask & legend	
[]	1	180 Ohm 1/4 watt 5% resistor	brown-grey-brown
[]	2	240 Ohm 1/4 watt 5% resistor	red-yellow-brown
[]	1	330 Ohm 1/4 watt 5% resistor	orange-orange-brown
[]	2	470 Ohm 1/4 watt 5% resistors	yellow-purple-brown
[]	2	560 Ohm 1/4 watt 5% resistors	green-blue-brown
[]	1	750 Ohm 1/2 watt 5% resistor	purple-green-brown
[]	8	1k Ohm 1/4 watt 5% resistors	brown-black-red
		NOTE: On early versions of the silk screened legend on the circuit board, a 3.3k Ohm resistor is shown just to the right of IC 6300 at board position 8C. This is an error. This should be a 1k Ohm resistor.	
[]	1	1.5k Ohm 1/4 watt 5% resistor	brown-green-red
[]	5	3.3k Ohm 1/4 watt 5% resistors	orange-orange-red
[]	3	4.7k Ohm 1/4 watt 5% resistors	yellow-purple-red
[]	2	6.19k Ohm 1/8 watt 1% resistors	blue-brown white-brown
[]	2	10k Ohm 1/4 watt 5% resistors	brown-black-orange
[]	1	18.2k Ohm 1/8 watt 1% resistor	brown-grey-red-red
[]	1	20.5k Ohm 1/8 watt 1% resistors	red-black-green-red
[]	2	27k Ohm 1/4 watt 5% resistors	red-purple-orange
[]	1	47k Ohm 1/4 watt 5% resistor	yellow-purple-orange
[]	1	54.9k Ohm 1/8 watt 1% resistor	green-yellow-white-red
[]	1	86.6k Ohm 1/8 watt 1% resistor	white-blue-blue-red
[]	4	1M Ohm 1/4 watt 5% resistors	brown-black-green
[]	1	180 Ohm 1/8 watt 5% 9 resistor SIP array	SIP3
[]	1	1k Ohm 1/8 wattt 5% 9 resistor SIP array	SIP1
[]	2	3.3k Ohm 1/8 watt 5% 9 resistor SIP array	SIP2,SIP4
[]	3	33 picofarad 5% silver mica capicators	

DJ/2D MODEL B PARTS LIST

[]	2	47 picofarad 2% silver mica capicator	
[]	2	112 picofarad 2% silver mica capicator	
[]	1	470 picofarad 5% silver mica capicator	
[]	1	.001 microfarad ceramic disk capicator	
[]	1	.01 microfarad mylar capicator	
[]	3	1.0 - 2.0 microfarad dipped tantalum capicator	
[]	6	1.0 - 4.7 microfarad axial lead tantalum capicators	
[]	2	39 microfarad axial lead tantalum capicators	
[]	16	ceramic disk capicators - may vary in value from .01 to .1 microfarads depending on current supplies	
[]	1	Dual-in-line 50 conductor right angle header	P1
[]	1	Single-in-line 7 conductor right angle header	P2
[]	1	3-pin header	
[]	1	2-pin header	
[]	1	Heat sink for the 7805 regulator at bottom of board	
[]	4	6-32 5/16 pan head machine screws	
[]	4	6-32 1/4" hex machine nuts	
[]	1	5.0688 MHz HU/18 Crystal	
[]	1	10.0000 MHz HU/18 Crystal	
[]	2	8 position DIP switch arrays	5D,13C
[]	1	1N751A 5.1 volt Zener diode	
[]	8	1N914/4820-0201 signal diodes	
		NOTE: The silk screened legend on the circuit board shows a group of four diodes just above the 1791/8866 controller at position 14C on the circuit board. These parts are not to be installed and are not furnished with the kit. These parts go with a version of the 1791 controller that Western Digital is not presently making.	
[]	1	RL209 light emitting diode	
[]	2	2N3904 transistor	

DJ/2D MODEL B PARTS LIST

[]	2	2N3906 transistor	
[]	1	8 pin low-profile socket	
[]	15	14 pin low-profile sockets	
[]	13	16 pin low-profile sockets	
[]	3	18 pin low-profile sockets	
[]	7	20 pin low-profile sockets	
[]	1	24 pin low-profile socket	
[]	2	40 pin low-profile sockets	
[]	2	74LS00 quad 2-input NAND gate	2A, 3B
[]	1	74LS02 quad 2-input NOR gate	3A
[]	1	74LS04/LS14 hex inverter	5C
[]	1	7404 hex inverter	2C
[]	1	74LS08 quad 2-input AND gate	7B
[]	1	74LS10 triple 3-input NAND gate	7A
[]	1	74LS30 8-input NAND gate	7C
[]	1	74LS32 quad 2-input OR gate	4C
[]	1	7438/LS38 quad 2-input NAND buffer	8B
[]	5	74LS74 dual D type flip-flop	4A, 5A, 6A, 8A, 2B
[]	1	74LS155 dual 1 of 4 decoder	6B
[]	1	74160/LS160/74161/LS161 4 bit counter	6C
[]	1	74175/LS175 4 bit dual rail register	9B
[]	1	74LS221 dual monostable	2D
[]	1	74LS240 octal tri-state inverting buffer	10D
[]	2	74LS244 octal tri-state buffer	6D, 8D
[]	1	74273/LS273 octal latch	12C
[]	1	74367/LS367 hex tri-state buffer	13B
[]	4	74368/LS368 tri-state inverting buffer	10B, 11B, 4D, 12B

DJ/2D MODEL B PARTS LIST

[]	1	74LS373 octal tri-state buffer/latch	7D
[]	1	74390/LS390 dual decade counter	3C
[]	1	81LS96/LS98 octal tri-state inverting buffer	9D
[]	1	25LS2521 octal comparator	11C
[]	1	96LS02 dual monostable	4B
[]	1	MMI6300/6301/82S129/74S287 4 x 256 PROM	8C
[]	1	MMI6331/82S123/74S288 8 x 32 PROM	3D
[]	1	2708 8 x 1k EPROM	11D
[]	2	2114-3L 4 x 1k low power 300NS static RAM	9C,10C
[]	1	BR1941/2941/COM5016 dual baud rate generator	13D
[]	1	TR1602/TR1868/MB8866 Uart	14D
[]	1	FD1791/8866 dual density floppy disk controller	14B
[]	1	1448/4558 dual operational amplifier	1C
[]	1	7812/78M12 monolithic 12 volt .5 amp regulator	
[]	1	79L05 monolithic -5 volt 100 ma regulator	
[]	1	7912/79M12 monolithic -12 volt regulator	

ASSEMBLY INSTRUCTIONS

WARNING! IMPROPER ASSEMBLY OF THIS KIT WILL VOID THE WARRANTY. READ THESE INSTRUCTIONS CAREFULLY BEFORE ATTEMPTING TO CONSTRUCT THIS KIT

INVENTORY

Make sure that all parts listed in the Parts List have been included. Notify Morrow Designs immediately if any are missing. Also, quickly return all extra parts.

USE BENDING BOARD

With the exception of the axial tantalum capacitors, the 1N751A zener diode, one of the 1/4 watt 240 Ohm resistors, and the 1/2 watt 750 Ohm resistors, all the resistor and diode leads should be bent to .4 inches. The leads of the 750 Ohm resistors should have a spacing of .55 inches. The axial lead tantalum capacitors should be bent to .7 inches. Use of a bending block will give your finished kit a more professional look.

USE SOCKETS

Sockets are provided for every IC on the Disk Jockey.

NO REPAIR WORK WILL BE ATTEMPTED ON ANY RETURNED BOARD WITH ANY IC SOLDERED DIRECTLY TO THE CARD

ORIENTATION

When this manual refers to the bottom of the circuit board it means the side with the gold S-100 edge connectors. Right and left assume a view from the component side of the board which has the silk screen legend.

All IC sockets will either have their pins numbered, have a 45 degree angle across the corner of pin one, or have a deep groove at the top of the socket. On the Disk Jockey, all sockets and all IC's have pin 1 closest to the top left corner of the board.

Assembly Instructions

The tantalum capacitors are polarized. The dipped tantalum cap has a red dot at its positive lead. This lead should be inserted at the side of the oval legend where the "+" sign is located. The 1.0 microfarad capacitor's positive lead is identified by a circular "tit" where it enters the body of the housing. The positive end of the 39 microfarad capacitors is identified by a red band. The silk screen identifies the positive lead of these axial parts with a "+" sign. The by-pass caps, identified on the silk screened legend by an asterisk "*" enclosed by a box, are not polarized. The .01 mylar cap and the

The two DIP switch arrays are to be positioned so that switch paddle number 1 is toward the bottom of the board.

The SIP resistor packs, historically prone to being inserted backwards, should have their white dot nearest the white dot on their respective legends. For SIP2 and SIP4 this means that the white dot should point toward the top of the board. For SIP1, the white dot should point to the left and for SIP3, the dot will point to the right.

The crystals included in this kit have a piece of foam pad attached to their PC board side. When these parts are installed, the protective paper on the back of the pad should be peeled off just before the leads are inserted through the circuit board at the position indicated on the parts legend. The foam pad has an adhesive on it which will hold the crystal to the circuit board. The pad and the adhesive are insulators so that no short circuit can occur when the crystal is installed.

The orientation of the transistors is indicated on the silk screen legend of the circuit board, as is their type number. A very common cause of smoke on power-up is a 2N3906 correctly oriented in the place of a 2N3904 and vice versa.

The black band at one end of the diodes marks the cathode and should correspond to the white arrow point on the legend of the circuit board.

Placing the 50 pin flat cable connector, P1, upside down is a disaster. The angled pins should go through the circuit board. Only the longer straight pins are long enough to accept the ribbon cable to the disk drive. The I/O connector, P2, should be positioned so that the longer angled pins point toward the top of the board while the shorter straight pins go through the circuit board.

Assembly Instructions

EXAMINE THE BOARD

Visually examine the circuit board for any trace opens or shorts. A concentrated five minute scrutiny will uncover most trace defects. Several hours of scattered, unconcentrated scrutiny generally won't reveal anything. Take special care that no shorts or opens exist on those areas of the circuit board that will be covered by IC sockets. Ohm out any suspicious looking traces for either shorts or discontinuity as appropriate. Return immediately any bare board found to be flawed. Such boards will be replaced under warranty.

SOLDERING AND SOLDER IRONS

The most desirable soldering tool for complex electronic kits is a constant temperature iron with an element regulated at 650 degrees F. The tip should be fine so that it can be brought into close contact with the pads of the circuit board. Such irons are available from Weller and Unger and should be part of any electronics shop.

There are three important soldering requirements for building this kit:

1. Do not use an iron that is too cold (less than 600 degrees F) or too hot (more than 750 degrees F).
2. Do not hold the iron against a pad for more than about six seconds.
3. Do not apply excessive amounts of solder.

The recommended procedure for soldering components to the circuit board is as follows:

1. Bring the iron in contact with BOTH the component lead AND the pad.
2. Apply a SMALL amount of solder at the point where the iron, component lead, and pad ALL make contact.
3. After the initial application of solder has been accomplished with the solder flowing to the pad and component lead, the heat of the iron will have transferred to BOTH the pad AND lead. Apply a small amount of additional solder to cover the joint between the pad and the lead.

DO NOT PILE SOLDER ON THE JOINT! EXCESSIVE HEAT AND SOLDER CAUSE PADS AND LEADS TO LIFT FROM THE CIRCUIT BOARD. EXCESSIVE SOLDER IS THE PRIMARY CAUSE FOR BOARD SHORTS AND BRIDGED CONNECTIONS.

PARTS INSTALLATION

[] Install and solder the eight signal diodes (1N914 or equivalent) and clip the excess leads from the parts. Be sure that the black bands of the diodes are positioned to match the arrow points of the white legend of the circuit board.

[] Install, solder, and trim the 1N751A zener diode.

PROTECT YOUR EYES WHEN YOU CLIP COMPONENT LEADS AFTER SOLDERING

[] Install and solder all the 1/4 watt resistors in place. Do this in sections so that the leads can be conveniently clipped.

[] Install, solder, and trim the leads of the 1% precision resistors.

[] Next, install, solder and trim the leads of the 750 Ohm 1/2 watt resistor.

[] Install and solder the 40 pin sockets first, then the 24, 20, 18, 16, and 14 pin sockets in that order. Finally install and solder the 8 pin socket. By installing the sockets in this order, a smaller sized socket will never be placed in a larger sized position.

[] Install and solder the SIP resistor pack arrays. The top pack at the left should have its dot to the left. The top pack at the right should have its dot to the right. The two packs at the center and at the bottom of the board should have their dots pointing toward the top of the board.

[] Install and solder the 6 axial lead 1.0 microfarad capacitors. The top two have their "+" leads to the left. The next pair have their "+" leads to the right and the final two will have their "+" leads pointing to the left again. Clip the excess leads from the parts.

[] Install, solder, and clip the leads of the two 39 microfarad caps. The red band of these parts must point to the left.

[] Bend the leads of the 7812, 7912, and one of the 7805 regulators. Set the other 7805 aside for now. Install the top three regulators at the left hand side of the board by placing a nut on top of the regulator, insert a screw from the bottom of the circuit board through the hole of the board and through the hole of the regulator. Hand tighten the nut. Solder the leads. Tighten the screws firmly.

[] After bending the leads 90 degrees, install and solder the two crystals in place. Clip the excess leads. Fix them to the circuit board by peeling the protective paper off their foam pad and pressing the pad against the board. Be sure to solder the crystals into place so that their padded side will fall into the area outlined on the silk screened legend.

Parts Installation

[] Install and solder the two connectors P1 and P2. Be sure to reread the orientation section before installing these parts.

[] Install and solder the light emitting diode at the top of the board just to the left of P1. One of the leads of this diode is longer than the other. The longer lead is the anode and must be to the left when the part is inserted. Clip the excess leads after soldering.

[] Install, solder and clip the leads of the 1.5 dipped tantalum capacitors. A total of three are to be installed. One is just to the right of the 7805 regulator in upper left corner of the board. The red dot of this device is to point to the left. The rest have their dots pointing toward the top of the board. There is one to the right of the 1791/8866 IC at position 14B, and another to the left of the 1602 IC at position 14D.

[] Install, solder and clip the 33 picofarad silver mica cap just to the right of the 10 Meg crystal in the left side of the board.

[] Install, solder and clip the leads of the 47 and 112 picofarad silver mica caps just to the left of the 74LS123 IC at location 2D.

[] Install, solder and clip the two 33 picofarad silver mica caps-- one between the 74LS10 IC at 7A and the 74LS74 IC at 8A and the other between the 6631 IC at 3D and the 74LS367 IC at 4D.

[] Install, solder and clip the 470 picofarad mica cap at the upper left of the 7404 IC at location 2C.

[] Install, solder and clip the .001 microfarad disk cap to the left of the 10 MHz crystal.

[] Install, solder and clip the .01 microfarad mylar cap to the left of the .001 disk cap just installed.

[] Install, solder and clip the leads of the three transistors just to the right of the regulator area carefully observing the placement and orientation information silk screened on the circuit board.

[] Install and solder the two DIP switch arrays. Switch 1 of each DIP should be positioned toward the bottom of the board.

[] Install, solder, and clip the leads of the 16 by-pass capacitors whose positions are identified by rectangular boxes each with asterisk "*" in the middle.

Parts Installation

[] Bend the leads of the remaining 7805 regulator and insert it in the circuit board. Place a separate, finned heat sink between the regulator and the board, work a screw from the back of the board through the board, heat sink, and regulator and hand tighten into the nut on top of the regulator. Solder the leads and adjust the wings of the separate heat sink and, finally, tighten the screw.

CLEAN AND EXAMINE THE BOARD

Use flux cleaner to remove solder rosin residue. Examine the circuit board carefully for shorts, solder bridges, or missed pins.

HOW TO FIND WHERE TO PLACE PARTS

For parts placement, please see the silk screened legend on the printed circuit board.

IC's may vary from those marked on the silk screened legend if they are listed as alternate IC's (following a slash) in the Parts List.

DO NOT INSERT ANY IC'S IN THEIR SOCKETS AT THIS TIME
--

INITIAL CHECK-OUT AND POWER-UP

Before inserting any IC's in their sockets perform the following check-out procedure:

1. Re-check the back of the board for solder shorts and bridged connections and for pins of IC sockets that have not been soldered. These unsoldered pins can cause aggravating intermittent problems during check-out.

2. Re-check components for orientation and make sure all components to be soldered have been soldered.

3. With an ohm meter, check for shorts between all regulated voltages (+5V,-5V,+12V,-12V) and ground and between any two regulator outputs (all regulator output pins are on the right side of the regulator, towards the bottom of the circuit board in this case). Check for shorts between S-100 supply voltages (+8V,+16V, -16V) and ground. S-100 pins 1 and 51 hold 8 volts, pin 2 holds +16 volts, and pin 52 -16 volts. Ground is on S-100 pins 50 and 100. Check these voltages for shorts among each other.

Parts Installation

4. Place the board WITHOUT IC's into an empty system bus slot and power up. In case of smoke, power down immediately and investigate.

5. With a VOM or scope, check the regulators for +5V (both of the 7805's), +12V, and -12V. The bottom pin of all four regulators is the output. Check for Vcc and ground on all IC's. Check for +12V on the 1791/8866 controller, the 2941 baud rate generator, and the 1458/4558 op amp. Check for -12V on the 1602 UART and the 1458/4558 op amp. Finally, check for -5V on the 2941 baud rate generator. If everything is OK, power down and proceed to the next step.

IC INSERTION

If an IC insertion tool is not available, IC leads should be straightened a ROW at a time, not by the individual PIN. The edge of a straight sided table is an excellent device for this operation. Hold the IC by the plastic case, place one row of legs against a flat surface and push very slightly. Repeat with the opposite row. Continue this procedure until the legs of the IC can be inserted with minimum effort into its socket.

When inserting an IC into its socket, take care that you DO NOT BEND THE IC'S LEGS UNDERNEATH ITS PLASTIC PACK. This is an extremely common error and can escape even a fairly careful visual inspection.

If IC pins become bent under during insertion, use a long nose pliers to straighten them and try again. When removing an IC from its socket, use an IC remover, an IC test clip (another must for any electronics shop) or a miniature screw driver. DO NOT ATTEMPT TO REMOVE AN IC WITH YOUR FINGERS. You will bleed on severely bent pins.

Once all IC's have been inserted, re-check for bent pins. Then check twice for proper orientation. Upside down IC's are generally destroyed upon power up.

IF FOR ANY REASON IT BECOMES NECESSARY TO REMOVE A COMPONENT WHICH HAS BEEN SOLDERED TO THE CIRCUIT BOARD, CLIP ALL LEADS BEFORE REMOVING. THIS WILL REDUCE THE CHANCE OF LIFTING PADS OFF TRACES.
--

Parts Installation

POWER UP

If all previous checks have been performed, you are ready to put power to your fully populated board. In an empty system with power off, insert the Disk Jockey and power up. If the board smokes, power down and investigate. If not, measure the regulated voltages again.

If any voltages have been lost since powering up the bare board, power down and check for upside down IC's. Isolate the possible faulty chip or chips by powering down, removing a section of IC's, and powering up again. Continue this sequence until the faulty IC or IC's are found.

BE SURE NEVER TO INSERT OR REMOVE A BOARD WITH POWER ON! THIS MAY DAMAGE THE BOARD

This completes the initial check-out of your Disk Jockey. If there are any problems or questions regarding the operation of your Disk Jockey contact the service department of Morrow Designs, (415) 524-2104.

DJ/2D MODEL B MEMORY MAP

HEX ADDRESS	FUNCTION		OCTAL ADDRESS
E000-E3F7	ROM FIRMWARE		340:000-343:367

	I/O REGISTERS		
	WHEN READ	WHEN WRITTEN	
-----		-----	
E3F8	UART INVERTED DATA INPUT	UART INVERTED DATA OUTPUT	343:370
E3F9	UART INVERTED STATUS	DISK JOCKEY FUNCTION	343:371
E3FA	DISK JOCKEY STATUS	DRIVE CONTROL REGISTER	343:372
E3FB	NOT USED	NOT USED	343:373
E3FC	1791 CONTROLLER STATUS	1791 CONTROLLER COMMAND	343:374

E3FD	1791 TRACK REGISTER		343:375
E3FE	1791 SECTOR REGISTER		343:376
E3FF	1791 DATA REGISTER		343:377

E400-E7FF	RAM		344:000-347:377

SOFTWARE LISTINGS

Morrow Designs makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Futher, Morrow Designs reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Morrow Designs to notify any person of such revision or changes.

```

*****
*
* Boot loader program for cp/m. The following code is
* loaded by the boot program on the Disk Jockey 2D. The
* 2D loads sector one of track zero into memory at
* ORIGIN+300H (the last page of ram on the controller)
* then jumps there. It is the responsibility of this code
* to load in the rest of cp/m.
*
*****

2900 =      CPMORG EQU      2900H      ;CPM STARTING ADDRESS
E000 =      ORIGIN EQU      0E000H    ;Disk Jockey starting address
E400 =      RAM EQU      ORIGIN+400H    ;ram starting address (of 2D)
E640 =      STACK EQU      RAM+240H    ;stack pointer starting address within ram
E009 =      TKZERO EQU      ORIGIN+11Q  ;track zero seek entry point
E00C =      TRKSET EQU      ORIGIN+14Q  ;entry for track seek
E00F =      SETSEC EQU      ORIGIN+17Q  ;entry point for sector set
E012 =      SETDMA EQU      ORIGIN+22Q  ;entry address for read/write beginning address
E015 =      DREAD EQU      ORIGIN+25Q  ;disk read entry point
E018 =      DWRITE EQU      ORIGIN+30Q  ;disk write routine address
E024 =      DMAST EQU      ORIGIN+44Q  ;disk read/write status routine

E700          ORG      ORIGIN+700H

*****
*
* load: load in all the rest of cp/m and the cbios. There
* are only two ways to exit this code: 1) If an
* error occurs, a jump is made to the loader on the
* Disk Jockey 2D. 2) If everything works, a jump is
* made to the starting location of the cold boot in
* the cbios.
*
*****

E700 21003E      LOAD      LXI      H,CPMORG+1500H    ;starting location for cbios
E703 3140E6      LXI      SP,STACK                    ;initialize the stack
E706 E5          PUSH     H                          ;save jump address for return later
E707 01022E      STADDR   LXI      B,2E02H            ;reg B=sector count, reg C=starting sector
E70A C5          PUSH     B                          ;save sector and count
E70B CD0FE0      CALL     SETSEC                      ;set the sector to read
E70E CD09E0      CALL     TKZERO                      ;home the drive
E711 210029      LXI      H,CPMORG                    ;starting location for load
E714 44          LDLOOP   MOV     B,H                  ;put starting address in B&C
E715 4D          MOV     C,L
E716 CD12E0      CALL     SETDMA                      ;set up starting load address
E719 060A        MVI      B,10                        ;retry counter
E71B C5          RDLOOP   PUSH    B                    ;save retry count
E71C CD15E0      CALL     DREAD                      ;read in the sector
E71F C1          POP     B                          ;fetch retry count
E720 D22AE7      JNC      RDGOOD                      ;take jump if read is ok.
E723 05          DCR      B                          ;update retry counter
E724 C21BE7      JNZ      RDLOOP                    ;try again if not ten errors
E727 C300E0      EXIT     JMP     ORIGIN              ;start all over from the beginning
E72A C1          RDGOOD   POP     B                    ;refetch sector count and #
E72B 05          DCR      B                          ;update the count
E72C C8          RZ          ;GO TO CPM IF DONE
E72D 0C          INR      C                          ;COMPUTE NEW SECTOR (MOD 26)
E72E 3E1B        MVI      A,27                      ;test if over 26
E730 B9          CMP     C
E731 C236E7      JNZ      OK                          ;take jump if sector < 27
E734 0E01        MVI      C,1                        ;start with sector 1 of next track
E736 C5          OK       PUSH    B                    ;save count and sector
E737 CC0CE0      CZ          TRKSET                    ;conditionally set new track
E73A C1          POP     B                          ;restore count and sector #
E73B C5          PUSH    B                          ;save it again
E73C CD0FE0      CALL     SETSEC                      ;set new sector
E73F CD24E0      CALL     DMAST                      ;get load address
E742 218000      LXI      H,200Q                    ;update the load address
E745 09          DAD     B
E746 C314E7      JMP     LDLOOP                      ;read next sector

```

```

*****
*
* save: write all of cpm and the cbios onto the disk.
* If an error occurs, the status returned by the
* 2D controller will be in location STACK-1.
*
*****

E749 2118E0 SAVE LXI H,DWRITE ;change load to write instead of read
E74C 221DE7 SHLD RDLOOP+2
E74F 215EE7 LXI H,ERROR ;change error return address
E752 2228E7 SHLD EXIT+1
E755 215BE7 LXI H,STALL ;get return address
E758 C303E7 JMP LOAD+3 ;go and do the write
E75B C35BE7 STALL JMP STALL ;stop here if everything ok !
E75E F5 ERROR PUSH PSW ;save status and flags
E75F C35FE7 ERROR1 JMP ERROR1 ;stop here on error.

*****
*
* intlz: write this cold boot loader program out to the
* disk.
*
*****

E762 3140E6 INTLZ LXI SP,STACK ;set up stack
E765 CD09E0 CALL TKZERO ;home the drive
E768 0100E7 LXI B,RAM+300H ;get starting address of this program
E76B CD12E0 CALL SETDMA ;set the write address
E76E 0E01 MVI C,1 ;set the sector to write
E770 CD0FE0 CALL SETSEC
E773 CD18E0 CALL DWRITE ;write this program out
E776 DA5EE7 JC ERROR
E779 C379E7 DONE JMP DONE ;stop here

```

```

*****
*
* CBIOS DRIVERS FOR CPM
*
* Currently the cbios is set up for a 16K cpm, to make a
* larger system, change the value of CPM.
*
*****

2900 = CPM EQU 2900H ;cp/m beginning load address
3106 = ENTRY EQU CPM+806H ;cp/m entrance point
0004 = CDISK EQU 4 ;current disk storage location
0003 = IOBYTE EQU 3H ;iobyte storage location

*****
*
* iobyte allows selection of different I/O devices. It
* can be initialized in any way by changing the equate
* below.
* Initial iobyte is currently defined as :
* console = tty
* reader = tty
* punch = tty
* list = tty
*
*****

0000 = INTIOBY EQU 0 ;initial iobyte,

*****
*
* The following equates reference the disk jockey/2d
* controller board. If your controller is non-standard
* then all the equates can be changed by re-assigning the
* value of ORIGIN to be the starting address of your
* controller.
*
*****

E000 = ORIGIN EQU 0E000H ;disk jockey/2d beginning address
E003 = INPUT EQU ORIGIN+3 ;serial input routine
E006 = OUTPUT EQU ORIGIN+6 ;serial output routine
E009 = TKZERO EQU ORIGIN+9H ;track zero seek routine
E00C = SEEK EQU ORIGIN+0CH ;regular track seek routine
E00F = SECTOR EQU ORIGIN+0FH ;set sector routine
E012 = DMA EQU ORIGIN+12H ;read/write beginning address set
E015 = DISKR EQU ORIGIN+15H ;disk read routine
E018 = DISKW EQU ORIGIN+18H ;disk write routine
E01B = SELECT EQU ORIGIN+1BH ;disk selection routine
E021 = TSTAT EQU ORIGIN+21H ;serial device status routine
E640 = STACK EQU ORIGIN+640H ;disk jockey/2d ram area for boot only
0099 = SEKERR EQU 99H ;seek error bit mask
00FF = RWERR EQU 0FFH ;read/write error bit mask
000D = ACR EQU 0DH ;carriage return
000A = ALF EQU 0AH ;line feed
E006 = COTTY EQU OUTPUT ;default character output
E003 = CITTY EQU INPUT ;default character input

*****
*
* The jump table immediately below must not be altered.
* It is ok to make the jumps to other address, but the
* function performed must be the same.
*
*****

3E00 ORG CPM+1500H

3E00 C32D3E START JMP BOOT ;cold boot
3E03 C3603E JMP WBOOT ;warm boot
3E06 C3C03E JMP CONST ;console status
3E09 C3CC3E JMP CONIN ;console input
3E0C C3DE3E CROUT JMP CONOUT ;console output
3E0F C3F93E JMP LIST ;list output
3E12 C3EE3E JMP PUNCH ;punch output
3E15 C3E43E JMP READER ;reader input
3E18 C3713E JMP HOME ;track zero home
3E1B C31BE0 JMP SELECT ;disk selection
3E1E C39B3E JMP SETTRK ;track seek
3E21 C30FE0 JMP SECTOR ;sector select
3E24 C312E0 JMP DMA ;read/write address select
3E27 C3A13E JMP READ ;disk read
3E2A C3BA3E JMP WRITE ;disk write

```

```

*****
*
* boot: load in all of cpm and then
* jump there. Initialize lobyte.
*
*****

3E2D 3140E6  BOOT    LXI    SP,STACK    ;initial stack
3E30 3E00    MVI    A,INTIOBY    ;initialize lobyte
3E32 320300  STA    IOBYTE
3E35 21643F  LXI    H,PROMPT    ;print signon message
3E38 CD8E3E  CALL   MESSG
3E3B AF      XRA    A    ;select disk A
3E3C 320400  STA    CDISK
3E3F 018000  GOCPM   LXI    B,80H    ;set up default disk buffer
3E42 CD12E0  CALL   DMA
3E45 3EC3    MVI    A,0C3H    ;put jump instruction to warm boot at 0
3E47 320000  STA    0
3E4A 21033E  LXI    H,START+3
3E4D 220100  SHLD   1
3E50 320500  STA    5    ;put jump to cpm entry at 5
3E53 210631  LXI    H,ENTRY
3E56 220600  SHLD   6
3E59 3A0400  LDA    CDISK    ;jump to cpm with current disk in C
3E5C 4F      MOV    C,A
3E5D C30029  JMP    CPM

*****
*
* warm boot: load in all of cpm except the cbios. Then
* enter cpm.
*
*****

3E60 3140E6  WBOOT   LXI    SP,STACK    ;initialize the stack
3E63 AF      XRA    A    ;select drive A
3E64 4F      MOV    C,A
3E65 CD18E0  CALL   SELECT
3E68 01022A  LXI    B,2A02H    ;sector count and beginning sector
3E6B CD0AE7  CALL   ORIGIN+70AH    ;call the cold start loader
3E6E C33F3E  JMP    GOCPM    ;now enter cpm

*****
*
* Home: move the head to track zero.
*
*****

3E71 CD09E0  HOME    CALL   TKZERO    ;call the disk jockey/2d
3E74 0E99    SEEK1   MVI    C,SEKERR    ;non relevent error mask

*****
*
* doerrs: returns if no error. Otherwise prints an appro-
* priate error messgae, and returns to cpm with an error
* indication.
*
*****

3E76 DA7B3E  DOERRS  JC      DOERR1    ;test if error
3E79 AF      RWOK    XRA    A    ;return if ok
3E7A C9      RET
3E7B A1      DOERR1  ANA    C    ;strip off unwanted errors
3E7C 0E08    MVI    C,8    ;error counter
3E7E 217A3F  LXI    H,MSGTBL    ;beginning address of messages
3E81 5E      DOLOOP  MOV    E,M'    ;get error address in D&E
3E82 23      INX    H
3E83 56      MOV    D,M
3E84 23      INX    H
3E85 1F      RAR        ;check if this bit is the error
3E86 DA8D3E  JC      MESSGA    ;yes, exit after printig error
3E89 0D      DCR    C    ;no error, update the count down
3E8A F2B13E  JP      DOLOOP    ;continue if not found

*
* if fall through then unknown error
*

3E8D EB      MESSGA  XCHG        ;put message address into H&L

```

```

*****
*
* messg: print the messgae pointed to by H&L and termin-
* ated by a 0FFH byte.
*
*****

3E8E 7E      MESSG  MOV    A,M      ;get character
3E8F A7      ANA     A          ;test for end
3E90 F8      RM
3E91 E5      PUSH    H          ;save address
3E92 4F      MOV     C,A        ;prep for console output
3E93 CD0C3E  CALL    CPOUT      ;output it
3E96 E1      POP     H          ;restore pointer
3E97 23      INX     H          ;bump to next character
3E98 C38E3E  JMP     MESSG      ;continue until end

*****
*
* settrk: call the disk jockey/2d to seek then exit by
* testing for errors.
*
*****

3E9B CD0CE0  SETTRK  CALL    SEEK
3E9E C3743E  JMP     SEEK1

*****
*
* read: read one sector from the disk. Try ten times on
* errors, before returning an error condition.
*
*****

3EA1 2115E0  READ     LXI     H,DISKR ;put disk read address into repeat loop
3EA4 22AB3E  RDWR    SHLD    RW+1
3EA7 060A     MVI     B,10      ;retry counter
3EA9 C5      RDWRL   PUSH    B
3EAA CD0000  RW       CALL    0      ;actually call disk read/write
3EAD C1      POP     B
3EAE D2793E  JNC     RWOK      ;exit if succesful
3EB1 05      DCR     B      ;test error count
3EB2 C2A93E  JNZ     RDWRL     ;continue if not zero
3EB5 0EFF     MVI     C,RWERR ;read/write error bit mask
3EB7 C3763E  JMP     DOERRS    ;print the appropriate error message

*****
*
* write: write data onto the disk, also try ten times
* before reporting an error.
*
*****

3EBA 2118E0  WRITE    LXI     H,DISKW
3EBD C3A43E  JMP     RDWR

*****
*
* const: get the status for the currently assigned console
* device. The console device can be gotten from
* iobyte, then a jump to the correct console status
* routine is performed.
*
*****

3EC0 212C3F  CONST    LXI     H,CSTBLE ;beginning of jump table
3EC3 C3CF3E  JMP     CONIN1  ;select correct jump

*****
*
* csreader: if the console is assigned to the reader then
* a jump will be made here, where another jump
* will occur to the correct reader status.
*
*****

3EC6 21343F  CSREADR  LXI     H,CSRTBLE ;beginning of reader status table
3EC9 C3E73E  JMP     READERA

```

```

*****
*
* conin: take the correct jump for the console input
* routine. The jump is based on the two least sig-
* nificant bits of iobyte.
*
*****

3ECC 21043F  CONIN  LXI      H,CITBLE      ;beginning of character input table
*
* entry at conin1 will decode the two least significant bits
* of iobyte. This is used by conin,conout, and const.
*

3ECF 3A0300  CONIN1  LDA      IOBYTE
3ED2 17      RAL

*
* entry at seldev will form an offset into the table pointed
* to by H&L and then pick up the address and jump there.
*

3ED3 E606    SELDEV  ANI      6H      ;strip off unwanted bits
3ED5 1600    MVI      D,0      ;form affset
3ED7 5F      MOV      E,A
3ED8 19      DAD      D      ;add offset
3ED9 7E      MOV      A,M      ;pick up high byte
3EDA 23      INX      H
3EDB 66      MOV      H,M      ;pick up low byte
3EDC 6F      MOV      L,A      ;form address
3EDD E9      PCHL      ;go there !

*****
*
* conout: take the proper branch address based on the two
* least significant bits of iobyte.
*
*****

3EDE 210C3F  CONOUT  LXI      H,COTBLE      ;beginning of the character out table
3EE1 C3CF3E  JMP      CONIN1 ;do the decode

*****
*
* reader: select the correct reader device for input. The
* reader is selected from bits 2 and 3 of iobyte.
*
*****

3EE4 21243F  READER  LXI      H,RTBLE ;beginning of reader input table
*
* entry at readera will decode bits 2 & 3 of iobyte, used
* by csreader.
*

3EE7 3A0300  READERA  LDA      IOBYTE
*
* entry at reader1 will shift the bits into position, used
* by list and punch.
*

3EEA 1F      READR1  RAR
3EEB C3D33E  JMP      SELDEV

*****
*
* punch: select the correct punch device. The seection
* comes from bits 4&5 of iobyte.
*
*****

3EEE 211C3F  PUNCH   LXI      H,PTBLE ;beginning of punch table
3EF1 3A0300  LDA      IOBYTE
*
* entry at pnchl rotates bits a little more in prep for
* seldev, used by list.
*

3EF4 1F      PNCH1   RAR
3EF5 1F      RAR
3EF6 C3EA3E  JMP      READR1

```



```

*****
*
* list: select a list device based on bits 6&7 of iobyte
*
*****

3EF9 21143F LIST LXI H,LTBLE ;beginning of the list device routines
3EFC 3A0300 LDA IOBYTE
3EFF 1F RAR
3F00 1F RAR
3F01 C3F43E JMP PNCH1

*****
*
* If customizing I/O routines is being performed, the
* table below should be modified to reflect the changes.
* all I/O devices are decoded out of iobyte and the jump
* is taken from the following tables.
*
*****

*
* console input table
*

3F04 03E0 CITBLE DW CTTY ;input from tty (currently assigned by intioby,input from 2d)
3F06 473F DW CICRT ;input from crt (currently SWITCHBOARD serial port 1)
3F08 E43E DW READER ;input from reader (depends on reader selection)
3F0A 473F DW CIUC1 ;input from user console 1 (currently SWITCHBOARD serial port 1)

*
* console output table
*

3F0C 06E0 COTBLE DW CTTY ;output to tty (currently assigned by intioby,output to 2d)
3F0E 3C3F DW COCRT ;output to crt (currently SWITCHBOARD serial port 1)
3F10 F93E DW LIST ;output to list device (depends on bits 6&7 of iobyte)
3F12 3C3F DW COUC1 ;output to user console 1 (currently SWITCHBOARD serial port 1)

*
* list device table
*

3F14 06E0 LTBLE DW CTTY ;output to tty (currently assigned by intioby,output to 2d)
3F16 3C3F DW COCRT ;output to crt (currently SWITCHBOARD serial port 1)
3F18 3C3F DW COLPT ;output to line printer (currently SWITCHBOARD serial port 1)
3F1A 3C3F DW COUL1 ;output to user line printer 1 (currently SWITCHBOARD serial port 1)

*
* punch device table
*

3F1C 06E0 PTBLE DW CTTY ;output to the tty (currently assigned by intioby,output to 2d)
3F1E 3C3F DW COPTP ;output to paper tape punch (currently SWITCHBOARD serial port 1)
3F20 3C3F DW COUP1 ;output to user punch 1 (currently SWITCHBOARD serial port 1)
3F22 3C3F DW COUP2 ;output to user punch 2 (currntly SWITCHBOARD serial port 1)

*
* reader device input table
*

3F24 03E0 RTBLE DW CTTY ;input from tty (currently assigned by intioby, input from 2d)
3F26 473F DW CIPTP ;input from paper tape reader (currently SWITCHBOARD serial port 1)
3F28 473F DW CIUR1 ;input from user reader 1 (currently SWITCHBOARD serial port 1)
3F2A 473F DW CIUR2 ;input from user reader 2 (currently SWITCHBOARD serial port 1)

*
* console status table
*

3F2C 533F CSTBLE DW CSTTY ;status of tty (currently assigned by intioby, status from 2d)
3F2E 5B3F DW CSCRT ;status from crt (currently SWITCHBOARD serial port 1)
3F30 C63E DW CSREADR ;status from reader (depends on reader device)
3F32 5B3F DW CSUC1 ;status from user console 1 (currently SWITCHBOARD serial port 1)

*
* status from reader device
*

3F34 533F CSRTBLE DW CSTTY ;status from tty (currently assigned by intioby, status of 2d)
3F36 5B3F DW CSPTR ;status from paper tape reader (currently SWITCHBOARD serial port 1)
3F38 5B3F DW CSUR1 ;status from user reader 1 (currently SWITCHBOARD serial port 1)
3F3A 5B3F DW CSUR2 ;status of user reader 2 (currently SWITCHBOARD serial port 1)

```

```

*****
*
* The following equates set output device to output to
* the SWITCHBOARD serial port 1.
*
*****

3F3C = COCRT EQU $ ;output from crt
3F3C = COUC1 EQU $ ;output from user console 1
3F3C = COUL1 EQU $ ;output from user line printer 1
3F3C = COPTP EQU $ ;output from paper tape punch
3F3C = COUP1 EQU $ ;output from user punch 1
3F3C = COUP2 EQU $ ;output from user punch 2
3F3C DB02 COLPT IN 2 ;output from line printer, get status
3F3E E680 ANI 80H ;wait until ok to send
3F40 CA3C3F JZ COLPT
3F43 79 MOV A,C ;output the character
3F44 D301 OUT 1
3F46 C9 RET

*****
*
* The following equates set the input from the devices to
* come from the SWITCHBOARD serial port 1
*
*****

3F47 = CIUC1 EQU $ ;input from user console 1
3F47 = CICRT EQU $ ;input from crt
3F47 = CIUR1 EQU $ ;input from user reader 1
3F47 = CIUR2 EQU $ ;input from user reader 2
3F47 DB02 CIPTR IN 2 ;input from paper tape reader, get status
3F49 E640 ANI 40H ;wait for character
3F4B CA473F JZ CIPTR
3F4E DB01 IN 1
3F50 E67F ANI 7FH ;strip off the parity
3F52 C9 RET

*****
*
* console status routines, test if a character has arrived
*
*****

3F53 CD21E0 CSTTY CALL TSTAT ;status from disk jockey 2d
3F56 3E00 STAT MVI A,0 ;prep for zero return
3F58 C0 RNZ ;nothing found
3F59 3D DCR A ;return with 0FFH
3F5A C9 RET

*****
*
* The following equates cause the devices to get status
* from the SWITCHBOARD serial port 1.
*
*****

3F5B = CSUR1 EQU $ ;status of user reader 1
3F5B = CSUR2 EQU $ ;status of user reader 2
3F5B = CSPTR EQU $ ;status of paper tape reader
3F5B = CSUC1 EQU $ ;status of user console 1
3F5B DB02 CSCRT IN 2 ;status from crt, get status
3F5D E640 ANI 40H ;strip of data ready bit
3F5F EE40 XRI 40H ;make correct polarity
3F61 C3563F JMP STAT ;return proper indication

*****
*
* The following messages could be put out by the cbios.
*
*****

3F64 0D0A PROMPT DB ACR,ALF ;prompt message - "16K CP/M VERS 1.4"
3F66 31364B20 DB '16K '
3F6A 43502F4D DB 'CP/M'
3F6E 20564552 DB 'VER'
3F72 5320312E DB 'S 1.'
3F76 34 DB '4'
3F77 0D0A DB ACR,ALF
3F79 FF DB 0FFH

```

*
* error message table
*

3F7A 8C3F	MSGTBL	DW	ILLDATA ;illegal data
3F7C 983F		DW	DATAREQ ;data request
3F7E A33F		DW	DATALOS ;data lost
3F80 AF3F		DW	CRCERR ;crc error
3F82 8B3F		DW	ILLSEC ;illegal sector
3F84 CF3F		DW	ILLDMA ;illegal dma
3F86 DA3F		DW	WRITPRO ;write protected
3F88 E53F		DW	NOTRDY ;not ready
3F8A F13F		DW	UNKNOWN ;unknown error

3F8C 0D0A	ILLDATA	DB	ACR,ALF
3F8E 494C474C20		DB	'ILGL DATA'
3F97 FF		DB	0FFH
3F98 0D0A	DATAREQ	DB	ACR,ALF
3F9A 4441544120		DB	'DATA REQ'
3FA2 FF		DB	0FFH
3FA3 0D0A	DATALOS	DB	ACR,ALF
3FA5 4441544120		DB	'DATA LOST'
3FAE FF		DB	0FFH
3FAF 0D0A	CRCERR	DB	ACR,ALF
3FB1 4352432045		DB	'CRC ERROR'
3FBA FF		DB	0FFH
3FBB 0D0A	ILLSEC	DB	ACR,ALF
3FBD 494C474C20		DB	'ILGL SECTOR/TRACK'
3FCE FF		DB	0FFH
3FCF 0D0A	ILLDMA	DB	ACR,ALF
3FD1 494C474C20		DB	'ILGL DMA'
3FD9 FF		DB	0FFH
3FDA 0D0A	WRITPRO	DB	ACR,ALF
3FDC 5752542050		DB	'WRT PROT'
3FE4 FF		DB	0FFH
3FE5 0D0A	NOTRDY	DB	ACR,ALF
3FE7 4E4F542052		DB	'NOT READY'
3FF0 FF		DB	0FFH
3FF1 0D0A	UNKNOWN	DB	ACR,ALF
3FF3 554E4B4F57		DB	'UNKOWN ERROR'
3FFF FF		DB	0FFH

340:000		1	AORG	OE000H
		2		
340:000	340:000	3	ORIGIN	EQU 340:000Q
		4		
340:000	343:336	5	BEGINS	EQU ORIGIN+3:336Q
340:000	344:000	6	RAM	EQU ORIGIN+4:000Q
340:000	343:370	7	IO	EQU ORIGIN+3:370Q
340:000	343:370	8	UDATA	EQU IO
340:000	343:371	9	DREG	EQU IO+1
340:000	343:371	10	USTAT	EQU IO+1
340:000	343:372	11	DCMD	EQU IO+2
340:000	343:372	12	DSTAT	EQU IO+2
340:000	343:373	13	CSTALL	EQU IO+3
340:000	343:374	14	CMDREG	EQU IO+4
340:000	343:374	15	CSTAT	EQU IO+4
340:000	343:375	16	TRKREG	EQU IO+5
340:000	343:376	17	SECREG	EQU IO+6
340:000	343:377	18	DATREG	EQU IO+7
		19		
340:000	000:001	20	LIGHT	EQU 1
340:000	000:001	21	HEAD	EQU 1
340:000	000:001	22	DENSITY	EQU 1
340:000	000:004	23	ISTAT	EQU 4
340:000	000:004	24	INTRQ	EQU 4
340:000	000:004	25	TZERO	EQU 4
340:000	000:004	26	LOAD	EQU 4
340:000	000:006	27	ULOAD	EQU 6
340:000	000:010	28	OSTAT	EQU 10Q
340:000	000:010	29	DSIDE	EQU 10Q
340:000	000:011	30	NOLITE	EQU 11Q
340:000	000:011	31	DCRINT	EQU 11Q
340:000	000:011	32	HCMD	EQU 11Q
340:000	000:020	33	INDEX	EQU 20Q
340:000	000:022	34	WINDXD	EQU 22Q
340:000	000:030	35	SKCMD	EQU 30Q
340:000	000:032	36	RINDXD	EQU 32Q
340:000	000:035	37	SVCMD	EQU 35Q
340:000	000:100	38	WPROT	EQU 100Q
340:000	000:100	39	ACCESS	EQU 100Q
340:000	000:200	40	RSTBIT	EQU 200Q
340:000	000:200	41	READY	EQU 200Q
340:000	000:210	42	RDCMD	EQU 210Q
340:000	000:250	43	WRCMD	EQU 250Q
340:000	000:300	44	STBITS	EQU 300Q
340:000	000:304	45	RACMD	EQU 304Q
340:000	000:320	46	CLRCMD	EQU 320Q
		47		
		48	*NP	

340:000	303	151	340	49	DBOOT	JMP	BOOT	
340:003	303	351	340	50	TERMIN	JMP	CIN	
340:006	303	332	340	51	TRMOUT	JMP	COUT	
340:011	303	132	341	52	TKZERO	JMP	HOME	
340:014	303	213	341	53	TRKSET	JMP	SEEK	
340:017	303	201	341	54	SETSEC	JMP	SECSET	
340:022	303	103	341	55	SETDMA	JMP	DMA	
340:025	303	335	341	56	DREAD	JMP	READ	
340:030	303	274	341	57	DWRITE	JMP	WRITE	
340:033	303	074	341	58	SELDRV	JMP	DRIVE	
340:036	303	370	340	59	TPANIC	JMP	CPAN	
340:041	303	003	341	60	TSTAT	JMP	TMSTAT	
340:044	303	064	341	61	DMAST	JMP	DMSTAT	
340:047	303	011	341	62	STATUS	JMP	DISKST	
340:052	303	305	340	63	DSKERR	JMP	LERROR	
340:055	303	263	343	64	SETDEN	JMP	DENFIX	
340:060	303	345	343	65	SETSID	JMP	SIDEXF	
				66				
340:063	000	066		67		DS	66Q	
				68				
340:151				69	BOOT			
340:151	061	372	347	70	LXI	SP,TRACK+1	initialize SP	
340:154	315	322	343	71	CALL	TIMOUT	poc/reset timeout	
340:157	041	001	000	72	LXI	H,1		
340:162	345			73	PUSH	H	track 0, sector 1	
340:163	056	011		74	MVI	L,DCRINT	set up the	
340:165	345			75	PUSH	H	-side select	
340:166	046	377		76	MVI	H,377Q	-and initial	
340:170	345			77	PUSH	H	-drive	
340:171	345			78	PUSH	H	-parameters	
340:172	345			79	PUSH	H		
340:173	345			80	PUSH	H		
340:174	041	010	000	81	LXI	H,10Q	initialize	
340:177	345			82	PUSH	H	-tzflag & cdisk	
340:200	056	176		83	MVI	L,176Q	initialize	
340:202	345			84	PUSH	H	-disk & drvsel	
340:203	056	010		85	MVI	L,10Q	initialize	
340:205	345			86	PUSH	H	-hdflag & dsflag	
340:206	046	030		87	MVI	H,30Q	initialize	
340:210	345			88	PUSH	H	-timer constant	
340:211	076	177		89	MVI	A,177Q	start 1791	
340:213	062	371	343	90	STA	DREG		
340:216	076	320		91	MVI	A,CLRCMD	1791 reset	
340:220	062	374	343	92	STA	CMDREG		
340:223				93	LDHEAD			
340:223	257			94	XRA	A	load the head	
340:224	315	033	343	95	CALL	HDCHK	-and test for	
340:227	322	245	340	96	JNC	DOOROK	-drive ready	
340:232	076	001		97	MVI	A,LIGHT	turn on the	
340:234	062	366	347	98	STA	DCREG	-error LED	
340:237	315	322	343	99	CALL	TIMOUT	timeout to	
340:242	303	223	340	100	JMP	LDHEAD	-close drive door	
				101	*NP			

340:245		102	DOOROK		
340:245 076 011		103		MVI A,NOLITE	turn off the
340:247 062 366 347		104		STA DCREG	-error LED
340:252 315 226 343		105		CALL MEASUR	head load time
340:255 301		106		POP B	adjust the stack
340:256 001 000 347		107		LXI B, RAM+300H	DMA addr
340:261 305		108		PUSH B	initialize
340:262 325		109		PUSH D	-dmaadr & timer
340:263 041 000 000		110		LXI H,0	initialize
340:266 345		111		PUSH H	-error counts
340:267 000		112		NOP .	debug instruction
340:270 305		113		PUSH B	boot address
340:271 006 014		114		MVI B,12	number of retrys
340:273		115	LDLOOP		
340:273 305		116		PUSH B	save the retry no.
340:274 315 335 341		117		CALL READ	read boot sector
340:277 301		118		POP B	restor retry no.
340:300 320		119		RNC .	successful read?
340:301 005		120		DCR B	no! - count down
340:302 302 273 340		121		JNZ LDLOOP	-and try again
340:305		122	LERROR		
340:305 016 011		123		MVI C,11Q	
340:307 021 303 242		124		LXI D,242:303Q	
340:312		125	LELOOP		
340:312 033		126		DCX D	
340:313 172		127		MOV A,D	
340:314 263		128		ORA E	
340:315 302 312 340		129		JNZ LELOOP	
340:320 076 010		130		MVI A,10Q	blink
340:322 251		131		XRA C	-the LED at
340:323 117		132		MOV C,A	-top of the
340:324 062 372 343		133		STA DCMD	-circuit board
340:327 303 307 340		134		JMP LERROR+2	
		135			
340:332		136	COUT		
340:332 072 371 343		137		LDA USTAT	get UART status
340:335 346 010		138		ANI OSTAT	output ready mask
340:337 302 332 340		139		JNZ COUT	test buffer empty
340:342 171		140		MOV A,C	character data
340:343 057		141		CMA .	negative logic bus
340:344 062 370 343		142		STA UDATA	send data to UART
340:347 057		143		CMA .	make positive
340:350 311		144		RET	
		145			
340:351		146	CIN		
340:351 072 371 343		147		LDA USTAT	get UART status
340:354 346 004		148		ANI ISTAT	input ready mask
340:356 302 351 340		149		JNZ CIN	wait for input
340:361 072 370 343		150		LDA UDATA	get the character
340:364 057		151		CMA .	adjust for negative bus
340:365 346 177		152		ANI 177Q	trim to 7 bits
340:367 311		153		RET	
		154			
340:370		155	CPAN		
340:370 072 371 343		156		LDA USTAT	get UART status
340:373 346 004		157		ANI ISTAT	input ready mask
340:375 300		158		RNZ .	test for data
340:376 315 351 340		159		CALL CIN	get character
341:001 271		160		CMP C	test for panic chtr
341:002 311		161		RET	
		162	*NP		

341:003			163	TMSTAT		
341:003	072	371	343	164	LDA	USTAT get UART status
341:006	346	004		165	ANI	ISTAT input ready mask
341:010	311			166	RET	
			167			
341:011			168	DISKST		
341:011	041	375	343	169	LXI	H,TRKREG most recent
341:014	116			170	MOV	C,M -track to C
341:015	043			171	INX	H most recent
341:016	106			172	MOV	B,M -sector to B
341:017	072	366	347	173	LDA	DCREG get current
341:022	057			174	CMA	. -density in
341:023	346	001		175	ANI	1 -the msb
341:025	017			176	RRC	. -position
341:026	127			177	MOV	D,A save in D
341:027	072	367	347	178	LDA	SIDE put the
341:032	007			179	RLC	. -most recent
341:033	007			180	RLC	. -side select
341:034	007			181	RLC	. -in bit positin
341:035	262			182	ORA	D -6 and merge
341:036	127			183	MOV	D,A save in D
341:037	072	350	347	184	LDA	DSFLAG get the
341:042	356	010		185	XRI	DSIDE -most recent
341:044	027			186	RAL	. -double sided
341:045	027			187	RAL	. -status and place
341:046	202			188	ADD	D -in bit position
341:047	127			189	MOV	D,A -5 and merge
341:050	072	375	347	190	LDA	SECLN get the
341:053	027			191	RAL	. -sector length
341:054	027			192	RAL	. -code bits in
341:055	262			193	ORA	D -positions 2 & 3
341:056	127			194	MOV	D,A -and merge
341:057	072	354	347	195	LDA	CDISK get the current
341:062	202			196	ADD	D -disk no. in bit
341:063	311			197	RET	. -positions 0 & 1
			198			
341:064			199	DMSTAT		
341:064	345			200	PUSH	H save the HL pair
341:065	052	346	347	201	LHLD	DMAADR move the
341:070	104			202	MOV	B,H -DMA address to
341:071	115			203	MOV	C,L -the BC pair
341:072	341			204	POP	H recover HL
341:073	311			205	RET	
			206			
341:074			207	DRIVE		
341:074	171			208	MOV	A,C drive select
341:075	346	003		209	ANI	3 -values must be
341:077	062	353	347	210	STA	DISK -between zero
341:102	311			211	RET	. -and three
			212	*NP		

341:103		213	DMA		
341:103	041 000 034	214		LXI H,-RAM	test the
341:106	011	215		DAD B	-DMA address
341:107	332 124 341	216		JC DMASET	-for conflict
341:112	041 010 040	217		LXI H,8-ORIGIN	
341:115	011	218		DAD B	-with the I/O
341:116	322 124 341	219		JNC DMASET	-on the DJ/2D
341:121	076 020	220		MVI A,20Q	-controller
341:123	311	221		RET	
341:124		222	DMASET		
341:124	140	223		MOV H,B	store the
341:125	151	224		MOV L,C	-BC pair
341:126	042 346 347	225		SHLD DMAADR	
341:131	311	226		RET	
		227			
341:132		228	HOME		
341:132	315 343 342	229		CALL HDLOAD	load the head
341:135	330	230		RC .	not ready error
341:136	315 160 341	231		CALL HENTRY	move the head
341:141	365	232		PUSH PSW	save status
341:142	237	233		SBB A	update the
341:143	062 371 347	234		STA TRACK	-track
341:146	062 375 343	235		STA TRKREG	-registers
341:151	257	236		XRA A	set the not
341:152	062 355 347	237		STA TZFLAG	-verified flag
341:155	303 043 342	238		JMP LEAVE+1	unload the head
341:160		239	HENTRY		
341:160	257	240		XRA A	set the force
341:161	062 351 347	241		STA HDFLAG	-verify flag
341:164	041 000 000	242		LXI H,0	timeout constant
341:167	076 011	243		MVI A,HCMD	move the head
341:171	315 142 343	244		CALL CENTRY	to track 0
341:174	346 004	245		ANI TZERO	track zero bit
341:176	300	246		RNZ	
341:177	067	247		STC .	error flag
341:200	311	248		RET	
		249			
341:201		250	SECSET		
341:201	257	251		XRA A	test for
341:202	261	252		ORA C	-zero value
341:203	067	253		STC .	error flag
341:204	310	254		RZ .	error return
341:205	346 037	255		ANI 37Q	trim & clear cry
341:207	062 370 347	256		STA SECTOR	
341:212	311	257		RET	
		258			
341:213		259	SEEK		
341:213	171	260		MOV A,C	test for
341:214	376 115	261		CPI 77	-track
341:216	077	262		CMC .	-too large
341:217	330	263		RC	
341:220	062 371 347	264		STA TRACK	
341:223	311	265		RET	
		266	*NP		

341:224		267	ISSUE	
341:224 062 343 347		268	STA ECOUNT+1	update count
341:227 315 226 343		269	CALL MEASUR	find the index
341:232 016 001		270	MVI C,1	start w/sector 1
341:234		271	ISLOOP	
341:234 171		272	MOV A,C	initialize the
341:235 062 376 343		273	STA SECREG	-sector register
341:240 072 370 347		274	LDA SECTOR	test for
341:243 271		275	CMP C	-target sector
341:244 310		276	RZ	
341:245 076 210		277	MVI A,RDCMD	do a fake
341:247 315 135 343		278	CALL COMAND	-read command
341:252 332 040 342		279	JC PLEAVE	abort on error
341:255 014		280	INR C	increment sector no.
341:256 303 234 341		281	JMP ISLOOP	
		282		
341:261		283	COMNDP	
341:261 062 374 343		284	STA CMDREG	start the operation
341:264 110		285	MOV C,B	initialize block count
341:265 021 377 343		286	LXI D,DATREG	data register
341:270 052 346 347		287	LHLD DMAADR	transfer address
341:273 311		288	RET	
		289		
341:274		290	WRITE	
341:274 315 063 342		291	CALL PREP	prepare for write
341:277 332 042 342		292	JC LEAVE	abort operation
341:302		293	WREENTRY	
341:302 076 250		294	MVI A,WRCMD	start a write
341:304 315 261 341		295	CALL COMNDP	
341:307		296	WRLOOP	
341:307 176		297	MOV A,M	load 1st byte of data
341:310 043		298	INX H	advance pointer
341:311 022		299	STAX D	write 1st byte of data
341:312 176		300	MOV A,M	load 2nd byte of data
341:313 043		301	INX H	advance pointer
341:314 022		302	STAX D	write 2nd byte of data
341:315 176		303	MOV A,M	load 3rd byte of data
341:316 043		304	INX H	advance pointer
341:317 022		305	STAX D	write 3rd byte of data
341:320 015		306	DCR C	reduce block count
341:321 176		307	MOV A,M	load 4th byte of data
341:322 043		308	INX H	advance pointer
341:323 022		309	STAX D	write 4th byte of data
341:324 302 307 341		310	JNZ WRLOOP	write next 4 bytes
341:327 041 302 341		311	LXI H,WREENTRY	return entry addr
341:332 303 373 341		312	JMP CBUSY	
		313	*NP	

341:335		314	READ		
341:335	315 063 342	315		CALL PREP	prepare for read
341:340	332 042 342	316		JC LEAVE	abort operation
341:343		317	RDENTRY		
341:343	076 210	318		MVI A,RDCMD	start a read
341:345	315 261 341	319		CALL COMNDP	
341:350		320	RDLOOP		
341:350	032	321		LDAX D	read 1st byte
341:351	167	322		MOV M,A	store 1st byte
341:352	043	323		INX H	advance pointer
341:353	032	324		LDAX D	read 2nd byte
341:354	167	325		MOV M,A	store 2nd byte
341:355	043	326		INX H	advance pointer
341:356	032	327		LDAX D	read 3rd byte
341:357	167	328		MOV M,A	store 3rd byte
341:360	043	329		INX H	advance pointer
341:361	015	330		DCR C	reduce block count
341:362	032	331		LDAX D	read 4th byte
341:363	167	332		MOV M,A	store 4th byte
341:364	043	333		INX H	advance pointer
341:365	302 350 341	334		JNZ RDLOOP	read next 4 bytes
341:370	041 343 341	335		LXI H,RDENTRY	return entry addr
		336			
341:373		337	CBUSY		
341:373	345	338		PUSH H	save return
341:374	041 374 343	339		LXI H,CSTAT	wait for 1791
341:377	315 154 343	340		CALL BUSY	-to finish command
342:002	346 137	341		ANI 137Q	error bit mask
342:004	312 041 342	342		JZ LEAVE-1	no error
342:007	376 020	343		CPI 20Q	premature interrupt
342:011	302 040 342	344		JNZ PLEAVE	other error type
342:014	072 342 347	345		LDA ECOUNT	decrement error
342:017	075	346		DCR A	-count number 1
342:020	372 027 342	347		JM STEST	hard interrupt error
342:023	062 342 347	348		STA ECOUNT	update count
342:026	311	349		RET .	do operation over
342:027		350	STEST		
342:027	072 343 347	351		LDA ECOUNT+1	decrement error
342:032	075	352		DCR A	-count number 2
342:033	362 224 341	353		JP ISSUE	issue a command
342:036	076 020	354		MVI A,20Q	irrecoverable error!
342:040		355	PLEAVE		
342:040	067	356		STC .	error flag
342:041	341	357		POP H	adjust the stack
342:042		358	LEAVE		
342:042	365	359		PUSH PSW	save the status
342:043	072 366 347	360		LDA DCREG	control bits
342:046	356 004	361		XRI LOAD	toggle the
342:050	062 372 343	362		STA DCMD	-head load bit
342:053	072 352 347	363		LDA DRVSEL	enable access to
342:056	062 371 343	364		STA DREG	-the data register
342:061	361	365		POP PSW	recover the status
342:062	311	366		RET	
		367	*NP		

342:063			368	PREP		
342:063	315	343	342	369	CALL	HDLOAD load the head
342:066	330			370	RC .	test for drive ready
342:067	072	375	343	371	LDA	TRKREG get old track
342:072	074			372	INR	A test for head
342:073	314	160	341	373	CZ	HENTRY -not calibrated
342:076	330			374	RC .	seek error?
342:077	041	375	343	375	LXI	H,TRKREG old track
342:102	072	371	347	376	LDA	TRACK new track
342:105	276			377	CMP	M test for head motion
342:106	043			378	INX	H advance to the
342:107	043			379	INX	H -data register
342:110	167			380	MOV	M,A save new track
342:111	171			381	MOV	A,C turn off data reg
342:112	062	371	343	382	STA	DREG -access control bit
342:115	312	152	342	383	JZ	TVERIFY test for seek
342:120	257			384	XRA	A force a read
342:121	062	351	347	385	STA	HDFLAG -header operation
342:124	072	372	343	386	LDA	DSTAT get the double
342:127	346	010		387	ANI	DSIDE -sided flag
342:131	062	350	347	388	STA	DSFLAG save for status
342:134	037			389	RAR	. shift for
342:135	037			390	RAR	. -3/6 ms step
342:136	037			391	RAR	. -rate constant
342:137	306	030		392	ADI	SKCMD do a
342:141	041	000	000	393	LXI	H,0 -seek
342:144	315	142	343	394	CALL	CENTRY -operation
342:147	332	216	342	395	JC	SERROR seek error?
				396		
342:152				397	TVERIFY	
342:152	072	351	347	398	LDA	HDFLAG get the force
342:155	267			399	ORA	A -verify hdr flag
342:156	302	271	342	400	JNZ	CHKSEC no seek & head OK
342:161	006	002		401	MVI	B,2 verify retry count
342:163				402	SLOOP	
342:163	076	035		403	MVI	A,SVCMD do a verify
342:165	315	135	343	404	CALL	COMAND -command
342:170	346	231		405	ANI	231Q error bit mask
342:172	127			406	MOV	D,A save
342:173	312	225	342	407	JZ	RDHDR no error!
342:176	072	366	347	408	LDA	DCREG denisty control
342:201	356	001		409	XRI	DENSITY flip the density
342:203	062	366	347	410	STA	DCREG update and
342:206	062	372	343	411	STA	DCMD -change density
342:211	005			412	DCR	B decrement retry
342:212	302	163	342	413	JNZ	SLOOP -count & test
342:215	172			414	MOV	A,D restore error bits
342:216				415	SERROR	
342:216	067			416	STC	. error flag
342:217	365			417	PUSH	PSW save errors
342:220	315	160	341	418	CALL	HENTRY seek to trk 0
342:223	361			419	POP	PSW recover errors
342:224	311			420	RET	
				421	*NP	

342:225		422	RDHDR		
342:225	006 012	423		MVI B,12Q	number of retrys
342:227		424	RHLOOP		
342:227	021 377 343	425		LXI D,DATREG	data register
342:232	041 372 347	426		LXI H,TRACK+1	data pointer
342:235	076 304	427		MVI A,RACMD	start a read
342:237	062 374 343	428		STA CMDREG	-header operation
342:242		429	RHL1		
342:242	032	430		LDAX D	get disk data
342:243	167	431		MOV M,A	store in mem
342:244	054	432		INR L	advance pointer
342:245	302 242 342	433		JNZ RHL1	test end of page
342:250	041 374 343	434		LXI H,CSTAT	wait for 1791
342:253	315 154 343	435		CALL BUSY	-to finish cmd
342:256	267	436		ORA A	test for errors
342:257	312 271 342	437		JZ CHKSEC	transfer OK?
342:262	005	438		DCR B	no! - test for
342:263	302 227 342	439		JNZ RHLOOP	-hard error
342:266	303 216 342	440		JMP SERROR	recalibrate
342:271		441	CHKSEC		
342:271	072 375 347	442		LDA SECLN	get the sector
342:274	117	443		MOV C,A	-size and setup
342:275	006 000	444		MVI B,0	-the table offset
342:277	041 337 342	445		LXI H,STABLE	sector table
342:302	011	446		DAD B	sector size pntr
342:303	072 370 347	447		LDA SECTOR	get the sector
342:306	107	448		MOV B,A	-and save in B
342:307	206	449		ADD M	compare w/table
342:310	076 020	450		MVI A,20Q	error flag
342:312	330	451		RC	error return
342:313	170	452		MOV A,B	initialize 1791
342:314	062 376 343	453		STA SECREG	-sector register
342:317	076 040	454		MVI A,40Q	128 byte sector
342:321	041 005 005	455		LXI H,5:005Q	initialize
342:324	042 342 347	456		SHLD ECOUNT	-error counts
		457			
342:327		458	SZLOOP		
342:327	015	459		DCR C	reduce size count
342:330	107	460		MOV B,A	sector size to B
342:331	370	461		RM	return on minus
342:332	027	462		RAL	double the count
342:333	267	463		ORA A	clear the carry
342:334	303 327 342	464		JMP SZLOOP	
		465			
342:337		466	STABLE		
342:337	345	467		DB 345Q	26 sector diskettes
342:340	345	468		DB 345Q	26 sector diskettes
342:341	360	469		DB 360Q	15 sector diskettes
342:342	367	470		DB 367Q	8 sector diskettes
		471	*NP		

342:343		472	HDLOAD		
342:343 041 353 347		473	LXI	H,DISK	new drv ptr
342:346 116		474	MOV	C,M	save new drv in C
342:347 043		475	INX	H	current drv ptr
342:350 136		476	MOV	E,M	save old drv in E
342:351 161		477	MOV	M,C	update current drv
342:352 043		478	INX	H	home cmd flag
342:353 173		479	MOV	A,E	test for
342:354 271		480	CMP	C	-drive change
342:355 176		481	MOV	A,M	head load mask
342:356 066 001		482	MVI	M,HEAD	update the mask
342:360 312 033 343		483	JZ	HDCHK	no drive change?
342:363 043		484	INX	H	addr of drive table
342:364 345		485	PUSH	H	save table addr
342:365 026 000		486	MVI	D,O	set up the
342:367 102		487	MOV	B,D	-offset address
342:370 031		488	DAD	D	calculate the
342:371 031		489	DAD	D	-parameter addr
342:372 072 366 347		490	LDA	DCREG	save the
342:375 167		491	MOV	M,A	density status
342:376 043		492	INX	H	track pointer
342:377 021 375 343		493	LXI	D,TRKREG	1791 trk reg
343:002 032		494	LDAX	D	get current track
343:003 167		495	MOV	M,A	save in the table
343:004 341		496	POP	H	beginning of table
343:005 011		497	DAD	B	new drive
343:006 011		498	DAD	B	-table pointer
343:007 176		499	MOV	A,M	get density status
343:010 062 366 347		500	STA	DCREG	update DCREG
343:013 043		501	INX	H	get the old
343:014 176		502	MOV	A,M	-track number
343:015 022		503	STAX	D	-and update 1791
343:016 076 177		504	MVI	A,177Q	drive select bits
343:020		505	DSROT		
343:020 007		506	RLC	.	rotate to
343:021 015		507	DCR	C	-select the
343:022 362 020 343		508	JP	DSROT	-proper drive
343:025 346 177		509	ANI	177Q	set the run bit
343:027 062 352 347		510	STA	DRVSEL	save in drv reg
343:032 257		511	XRA	A	force a head load
		512	*NP		

343:033			513	HDCHK		
343:033	041	372	343	514	LXI	H,DSTAT test for
343:036	246			515	ANA	M -head loaded
343:037	062	351	347	516	STA	HDFLAG save the head
343:042	365			517	PUSH	PSW -loaded status
343:043	072	352	347	518	LDA	DRVSEL get current drive
343:046	117			519	MOV	C,A save
343:047	072	367	347	520	LDA	SIDE get current side
343:052	057			521	CMA	. -and merge
343:053	241			522	ANA	C -with drive select
343:054	062	371	343	523	STA	DREG select drive & side
343:057	356	100		524	XRI	ACCESS toggle access bit
343:061	117			525	MOV	C,A save for PREP routine
343:062	072	366	347	526	LDA	DCREG den & head cntl bits
343:065	107			527	MOV	B,A save
343:066	072	371	347	528	LDA	TRACK get the new track
343:071	326	001		529	SUI	1 force single
343:073	237			530	SBB	A -density
343:074	075			531	DCR	A -if track = 0
343:075	057			532	CMA	. compliment
343:076	260			533	ORA	B merge w/control bits
343:077	167			534	MOV	M,A load head & set density
343:100	361			535	POP	PSW head load status
343:101	302	117	343	536	JNZ	RDYCHK conditionally
343:104	345			537	PUSH	H -wait for head
343:105	052	344	347	538	LHLD	TIMER -load time out
343:110				539		
343:110	053			540	DCX	H count down
343:111	174			541	MOV	A,H -40 ms for
343:112	265			542	ORA	L -head load
343:113	302	110	343	543	JNZ	TLOOP -time out
343:116	341			544	POP	H
343:117				545		
343:117	176			546	RDYCHK	
343:120	346	200		547	MOV	A,M test for
343:122	300			548	ANI	READY -drive ready
343:123				549	RNZ	
343:123	072	366	347	550	UNLOAD	
343:126	366	006		551	LDA	DCREG force a
343:130	167			552	ORI	ULOAD -head
343:131	076	200		553	MOV	M,A -unload
343:133	067			554	MVI	A,READY set drive
343:134	311			555	STC	. -not ready
				556	RET	. -error flag
343:135				557		
343:135	052	344	347	558	COMAND	
343:140	051			559	LHLD	TIMER get index count
343:141	051			560	DAD	H -and multiply
343:142				561	DAD	H -by four
343:142	353			562	CENTRY	
343:143	041	374	343	563	XCHG	. save in D-E pair
343:146	167			564	LXI	H,CSTAT issue command
343:147				565	MOV	M,A -to the 1791
343:147	176			566	NBUSY	
343:150	037			567	MOV	A,M wait
343:151	322	147	343	568	RAR	. -for the
				569	JNC	NBUSY -busy flag
					*NP	

343:154		570	BUSY		
343:154 176		571		MOV A,M	test for
343:155 037		572		RAR .	-device busy
343:156 176		573		MOV A,M	restore status
343:157 320		574		RNC .	return if not busy
343:160 303 166 343		575		JMP PATCH+3	jump around patch
343:163		576	PATCH		
343:163 303 343 342		577		JMP HDLOAD	patch for old ATE
343:166 033		578		DCX D	test for
343:167 172		579		MOV A,D	-two disk
343:170 263		580		ORA E	-revolutions
343:171 302 154 343		581		JNZ BUSY	47 machine cycles
343:174 136		582		MOV E,M	get error code
343:175 345		583		PUSH H	save cmd address
343:176 043		584		INX H	track register
343:177 126		585		MOV D,M	save present track
343:200 072 352 347		586		LDA DRVSEL	control bits
343:203 356 200		587		XRI RSTBIT	reset the 1791
343:205 062 371 343		588		STA DREG	-controller to
343:210 356 300		589		XRI STBITS	-clear the
343:212 343		590		XTHL .	-command busy
343:213 062 371 343		591		STA DREG	-fault condition
343:216 066 320		592		MVI M,CLRCMD	force interrupt
343:220 343		593		XTHL .	restore the
343:221 162		594		MOV M,D	-the track reg
343:222 341		595		POP H	restore the stack
343:223 173		596		MOV A,E	error code to A
343:224 067		597		STC .	-error flag
343:225 311		598		RET	
		599			
343:226		600	MEASUR		
343:226 021 000 000		601		LXI D,O	initialize count
343:231 041 372 343		602		LXI H,DSTAT	status port
343:234 016 020		603		MVI C,INDEX	index bit flag
343:236		604	INDXLO		
343:236 176		605		MOV A,M	wait for
343:237 241		606		ANA C	-index
343:240 312 236 343		607		JZ INDXLO	-pulse high
343:243		608	INDXHI		
343:243 176		609		MOV A,M	wait for
343:244 241		610		ANA C	-index
343:245 302 243 343		611		JNZ INDXHI	-pulse low
343:250		612	INDXCT		
343:250 023		613		INX D	advance count
343:251 343		614		XTHL .	four dummy
343:252 343		615		XTHL .	-instructions
343:253 343		616		XTHL .	-to lengthen
343:254 343		617		XTHL .	-the delay
343:255 176		618		MOV A,M	wait for
343:256 241		619		ANA C	-the index
343:257 312 250 343		620		JZ INDXCT	-to go high
343:262 311		621		RET .	98 machine cycles
		622	*NP		

343:263		623	DENFIX		
343:263 171		624	MOV	A,C	trim the
343:264 346 001		625	ANI	1	-excess bits
343:266 057		626	CMA	.	compliment and
343:267 107		627	MOV	B,A	-save in B
343:270 041 353 347		628	LXI	H,DISK	new disk ptr
343:273 136		629	MOV	E,M	get disk no.
343:274 026 000		630	MVI	D,O	offset addr
343:276 043		631	INX	H	current disk ptr
343:277 176		632	MOV	A,M	move to ACC
343:300 253		633	XRA	E	cmpr old w/new
343:301 365		634	PUSH	PSW	save status
343:302 043		635	INX	H	disk table
343:303 043		636	INX	H	-address
343:304 031		637	DAD	D	add the
343:305 031		638	DAD	D	-offset
343:306 176		639	MOV	A,M	get parameters
343:307 366 001		640	ORI	1	mask off density
343:311 240		641	ANA	B	set new density
343:312 167		642	MOV	M,A	update parameters
343:313 361		643	POP	PSW	test new=old?
343:314 300		644	RNZ		
343:315 176		645	MOV	A,M	updata CDISK
343:316 062 366 347		646	STA	DCREG	-also
343:321 311		647	RET		
		648			
343:322		649	TIMOUT		
343:322 041 000 000		650	LXI	H,O	time-out delay
343:325		651	TILOOP		
343:325 053		652	DCX	H	decrement count
343:326 174		653	MOV	A,H	test for delay
343:327 265		654	ORA	L	-count equal zero
343:330 343		655	XTHL	.	long NOP
343:331 343		656	XTHL	.	-instruction
343:332 302 325 343		657	JNZ	TILOOP	
343:335 311		658	RET		
		659			
343:336		660	SBEGIN		
343:336 345		661	PUSH	H	
343:337 041 342 343		662	LXI	H,DSTALL	
343:342		663	DSTALL		
343:342 351		664	PCHL		
343:343 341		665	POP	H	
343:344 311		666	RET		
		667			
343:345		668	SIDEXF		
343:345 171		669	MOV	A,C	get the side bit
343:346 346 001		670	ANI	1	trim the excess
343:350 027		671	RAL	.	move the bit
343:351 027		672	RAL	.	-to the side
343:352 027		673	RAL	.	-select bit
343:353 027		674	RAL	.	-position
343:354 062 367 347		675	STA	SIDE	save side bit
343:357 311		676	RET		
		677	*NP		

343:360		678	PWRJMP		
343:360 000		679	NOP	.	power-on
343:361 000		680	NOP	.	-jump
343:362 000		681	NOP	.	-sequence
343:363 000		682	NOP	.	-with NOP
343:364 000		683	NOP	.	-padding
343:365 303 000 340		684	JMP	DBOOT	
		685			
343:370 000:010		686	DS	10Q	I/O locations
		687			
347:311		688	AORG	RAM+3:311Q	
		689			
347:311 000:031		690	STACK	DS	31Q
		691			
347:342 000 000		692	ECOUNT	DW	0 error count cells
347:344 000 030		693	TIMER	DW	30:000Q head load time out
347:346 000 347		694	DMAADR	DW	RAM+300H dma address
347:350 010		695	DSFLAG	DB	10Q
347:351 000		696	HDFLAG	DB	0 read header flag
347:352 176		697	DRVSEL	DB	176Q drive select constant
347:353 000		698	DISK	DB	0 new drive
347:354 010		699	CDISK	DB	10Q current disk
347:355 000		700	TZFLAG	DB	0 home cmd indicator
347:356 011		701	DOPRAM	DB	11Q drive 0 parameters
347:357 377		702	DOTRK	DB	377Q drive 0 track no
347:360 011		703	D1PRAM	DB	11Q drive 1 parameters
347:361 377		704	D1TRK	DB	377Q drive 1 track no
347:362 011		705	D2PRAM	DB	11Q drive 2 parameters
347:363 377		706	D2TRK	DB	377Q drive 2 track no
347:364 011		707	D3PRAM	DB	11Q drive 3 parameters
347:365 377		708	D3TRK	DB	377Q drive 3 track no
347:366 011		709	DCREG	DB	11Q current parameters
347:367 000		710	SIDE	DB	0 new side
347:370 001		711	SECTOR	DB	1 new sector
347:371 000		712	TRACK	DB	0 new track
347:372 000		713	TRKNO	DB	0 disk
347:373 000		714	SIDENO	DB	0 -sector
347:374 000		715	SECTNO	DB	0 -header
347:375 000		716	SECLen	DB	0 -data
347:376 000		717	CRCLO	DB	0 -buffer
347:377 000		718	CRCHI	DB	0

E000		1		AORG	OE000H
		2			
E000	E000	3	ORIGIN	EQU	340:000Q
		4			
E000	E3DE	5	BEGINS	EQU	ORIGIN+3:336Q
E000	E400	6	RAM	EQU	ORIGIN+4:000Q
E000	E3F8	7	IO	EQU	ORIGIN+3:370Q
E000	E3F8	8	UDATA	EQU	IO
E000	E3F9	9	DREG	EQU	IO+1
E000	E3F9	10	USTAT	EQU	IO+1
E000	E3FA	11	DCMD	EQU	IO+2
E000	E3FA	12	DSTAT	EQU	IO+2
E000	E3FB	13	CSTALL	EQU	IO+3
E000	E3FC	14	CMDREG	EQU	IO+4
E000	E3FC	15	CSTAT	EQU	IO+4
E000	E3FD	16	TRKREG	EQU	IO+5
E000	E3FE	17	SECREG	EQU	IO+6
E000	E3FF	18	DATREG	EQU	IO+7
		19			
E000	0001	20	LIGHT	EQU	1
E000	0001	21	HEAD	EQU	1
E000	0001	22	DENSITY	EQU	1
E000	0004	23	ISTAT	EQU	4
E000	0004	24	INTRQ	EQU	4
E000	0004	25	TZERO	EQU	4
E000	0004	26	LOAD	EQU	4
E000	0006	27	ULOAD	EQU	6
E000	0008	28	OSTAT	EQU	10Q
E000	0008	29	DSIDE	EQU	10Q
E000	0009	30	NOLITE	EQU	11Q
E000	0009	31	DCRINT	EQU	11Q
E000	0009	32	HCMD	EQU	11Q
E000	0010	33	INDEX	EQU	20Q
E000	0012	34	WINDXD	EQU	22Q
E000	0018	35	SKCMD	EQU	30Q
E000	001A	36	RINDXD	EQU	32Q
E000	001D	37	SVCMD	EQU	35Q
E000	0040	38	WPROT	EQU	100Q
E000	0040	39	ACCESS	EQU	100Q
E000	0080	40	RSTBIT	EQU	200Q
E000	0080	41	READY	EQU	200Q
E000	0088	42	RDCMD	EQU	210Q
E000	00A8	43	WRCMD	EQU	250Q
E000	00C0	44	STBITS	EQU	300Q
E000	00C4	45	RACMD	EQU	304Q
E000	00D0	46	CLRCMD	EQU	320Q
		47			
		48	*NP		

E000	C3 69 E0	49	DBOOT	JMP	BOOT
E003	C3 E9 E0	50	TERMIN	JMP	CIN
E006	C3 DA E0	51	TRMOUT	JMP	COUT
E009	C3 5A E1	52	TKZERO	JMP	HOME
E00C	C3 8B E1	53	TRKSET	JMP	SEEK
E00F	C3 81 E1	54	SETSEC	JMP	SECSET
E012	C3 43 E1	55	SETDMA	JMP	DMA
E015	C3 DD E1	56	DREAD	JMP	READ
E018	C3 BC E1	57	DWRITE	JMP	WRITE
E01B	C3 3C E1	58	SELDRV	JMP	DRIVE
E01E	C3 F8 E0	59	TPANIC	JMP	CPAN
E021	C3 03 E1	60	TSTAT	JMP	TMSTAT
E024	C3 34 E1	61	DMAST	JMP	DMSTAT
E027	C3 09 E1	62	STATUS	JMP	DISKST
E02A	C3 C5 E0	63	DSKERR	JMP	LERROR
E02D	C3 B3 E3	64	SETDEN	JMP	DENFIX
E030	C3 E5 E3	65	SETSID	JMP	SIDEX
		66			
E033	0036	67		DS	66Q
		68			
E069		69	BOOT		
E069	31 FA E7	70		LXI	SP,TRACK+1 initialize SP
E06C	CD D2 E3	71		CALL	TIMOUT poc/reset timeout
E06F	21 01 00	72		LXI	H,1
E072	E5	73		PUSH	H track 0, sector 1
E073	2E 09	74		MVI	L,DCRINT set up the
E075	E5	75		PUSH	H -side select
E076	26 FF	76		MVI	H,377Q -and initial
E078	E5	77		PUSH	H -drive
E079	E5	78		PUSH	H -parameters
E07A	E5	79		PUSH	H
E07B	E5	80		PUSH	H
E07C	21 08 00	81		LXI	H,10Q initialize
E07F	E5	82		PUSH	H -tzflag & cdisk
E080	2E 7E	83		MVI	L,176Q initialize
E082	E5	84		PUSH	H -disk & drvsel
E083	2E 08	85		MVI	L,10Q initialize
E085	E5	86		PUSH	H -hdflag & dsflag
E086	26 18	87		MVI	H,30Q initialize
E088	E5	88		PUSH	H -timer constant
E089	3E 7F	89		MVI	A,177Q start 1791
E08B	32 F9 E3	90		STA	DREG
E08E	3E D0	91		MVI	A,CLRCMD 1791 reset
E090	32 FC E3	92		STA	CMDREG
E093		93	LDHEAD		
E093	AF	94		XRA	A load the head
E094	CD 1B E3	95		CALL	HDCHK -and test for
E097	D2 A5 E0	96		JNC	DOOROK -drive ready
E09A	3E 01	97		MVI	A,LIGHT turn on the
E09C	32 F6 E7	98		STA	DCREG -error LED
E09F	CD D2 E3	99		CALL	TIMOUT timeout to
EOA2	C3 93 E0	100		JMP	LDHEAD -close drive door
		101	*NP		

EOA5		102	DOOROK		
EOA5	3E 09	103		MVI A,NOLITE	turn off the
EOA7	32 F6 E7	104		STA DCREG	-error LED
EOAA	CD 96 E3	105		CALL MEASUR	head load time
EOD	C1	106		POP B	adjust the stack
EOAE	01 00 E7	107		LXI B, RAM+300H	DMA addr
EOB1	C5	108		PUSH B	initialize
EOB2	D5	109		PUSH D	-dmaadr & timer
EOB3	21 00 00	110		LXI H,0	initialize
EOB6	E5	111		PUSH H	-error counts
EOB7	00	112		NOP .	debug instruction
EOB8	C5	113		PUSH B	boot address
EOB9	06 0C	114		MVI B,12	number of retrys
EOBB		115	LDLOOP		
EOBB	C5	116		PUSH B	save the retry no.
EOBC	CD DD E1	117		CALL READ	read boot sector
EOBF	C1	118		POP B	restor retry no.
EOC0	D0	119		RNC .	successful read?
EOC1	05	120		DCR B	no! - count down
EOC2	C2 BB E0	121		JNZ LDLOOP	-and try again
EOC5		122	LERROR		
EOC5	0E 09	123		MVI C,11Q	
EOC7	11 C3 A2	124		LXI D,242:303Q	
EOCA		125	LELOOP		
EOCA	1B	126		DCX D	
EOCB	7A	127		MOV A,D	
EOCC	B3	128		ORA E	
EOCD	C2 CA E0	129		JNZ LELOOP	
EOD0	3E 08	130		MVI A,10Q	blink
EOD2	A9	131		XRA C	-the LED at
EOD3	4F	132		MOV C,A	-top of the
EOD4	32 FA E3	133		STA DCMD	-circuit board
EOD7	C3 C7 E0	134		JMP LERROR+2	
		135			
EODA		136	COUT		
EODA	3A F9 E3	137		LDA USTAT	get UART status
EODD	E6 08	138		ANI OSTAT	output ready mask
EODF	C2 DA E0	139		JNZ COUT	test buffer empty
EOE2	79	140		MOV A,C	character data
EOE3	2F	141		CMA .	negative logic bus
EOE4	32 F8 E3	142		STA UDATA	send data to UART
EOE7	2F	143		CMA .	make positive
EOE8	C9	144		RET	
		145			
EOE9		146	CIN		
EOE9	3A F9 E3	147		LDA USTAT	get UART status
EOEC	E6 04	148		ANI ISTAT	input ready mask
EOEE	C2 E9 E0	149		JNZ CIN	wait for input
EOF1	3A F8 E3	150		LDA UDATA	get the character
EOF4	2F	151		CMA .	adjust for negative bus
EOF5	E6 7F	152		ANI 177Q	trim to 7 bits
EOF7	C9	153		RET	
		154			
EOF8		155	CPAN		
EOF8	3A F9 E3	156		LDA USTAT	get UART status
EOFB	E6 04	157		ANI ISTAT	input ready mask
EOFD	C0	158		RNZ .	test for data
EOFE	CD E9 E0	159		CALL CIN	get character
E101	B9	160		CMP C	test for panic chtr
E102	C9	161		RET	
		162	*NP		

E103		163	TMSTAT		
E103	3A F9 E3	164		LDA	USTAT get UART status
E106	E6 04	165		ANI	ISTAT input ready mask
E108	C9	166		RET	
		167			
E109		168	DISKST		
E109	21 FD E3	169		LXI	H,TRKREG most recent
E10C	4E	170		MOV	C,M -track to C
E10D	23	171		INX	H most recent
E10E	46	172		MOV	B,M -sector to B
E10F	3A F6 E7	173		LDA	DCREG get current
E112	2F	174		CMA	. -density in
E113	E6 01	175		ANI	1 -the msb
E115	0F	176		RRC	. -position
E116	57	177		MOV	D,A save in D
E117	3A F7 E7	178		LDA	SIDE put the
E11A	07	179		RLC	. -most recent
E11B	07	180		RLC	. -side select
E11C	07	181		RLC	. -in bit positin
E11D	B2	182		ORA	D -6 and merge
E11E	57	183		MOV	D,A save in D
E11F	3A E8 E7	184		LDA	DSFLAG get the
E122	EE 08	185		XRI	DSIDE -most recent
E124	17	186		RAL	. -double sided
E125	17	187		RAL	. -status and place
E126	82	188		ADD	D -in bit position
E127	57	189		MOV	D,A -5 and merge
E128	3A FD E7	190		LDA	SECLen get the
E12B	17	191		RAL	. -sector length
E12C	17	192		RAL	. -code bits in
E12D	B2	193		ORA	D -positions 2 & 3
E12E	57	194		MOV	D,A -and merge
E12F	3A EC E7	195		LDA	CDISK get the current
E132	82	196		ADD	D -disk no. in bit
E133	C9	197		RET	. -positions 0 & 1
		198			
E134		199	DMSTAT		
E134	E5	200		PUSH	H save the HL pair
E135	2A E6 E7	201		LHLD	DMAADR move the
E138	44	202		MOV	B,H -DMA address to
E139	4D	203		MOV	C,L -the BC pair
E13A	E1	204		POP	H recover HL
E13B	C9	205		RET	
		206			
E13C		207	DRIVE		
E13C	79	208		MOV	A,C drive select
E13D	E6 03	209		ANI	3 -values must be
E13F	32 EB E7	210		STA	DISK -between zero
E142	C9	211		RET	-and three
		212	*NP		

E143		213	DMA		
E143	21 00 1C	214		LXI H,-RAM	test the
E146	09	215		DAD B	-DMA address
E147	DA 54 E1	216		JC DMASET	-for conflict
E14A	21 08 20	217		LXI H,8-ORIGIN	
E14D	09	218		DAD B	-with the I/O
E14E	D2 54 E1	219		JNC DMASET	-on the DJ/2D
E151	3E 10	220		MVI A,20Q	-controller
E153	C9	221		RET	
E154		222	DMASET		
E154	60	223		MOV H,B	store the
E155	69	224		MOV L,C	-BC pair
E156	22 E6 E7	225		SHLD DMAADR	
E159	C9	226		RET	
		227			
E15A		228	HOME		
E15A	CD E3 E2	229		CALL HDLOAD	load the head
E15D	D8	230		RC .	not ready error
E15E	CD 70 E1	231		CALL HENTRY	move the head
E161	F5	232		PUSH PSW	save status
E162	9F	233		SBB A	update the
E163	32 F9 E7	234		STA TRACK	-track
E166	32 FD E3	235		STA TRKREG	-registers
E169	AF	236		XRA A	set the not
E16A	32 ED E7	237		STA TZFLAG	-verified flag
E16D	C3 23 E2	238		JMP LEAVE+1	unload the head
E170		239	HENTRY		
E170	AF	240		XRA A	set the force
E171	32 E9 E7	241		STA HDFLAG	-verify flag
E174	21 00 00	242		LXI H,0	timeout constant
E177	3E 09	243		MVI A,HCMD	move the head
E179	CD 62 E3	244		CALL CENTRY	to track 0
E17C	E6 04	245		ANI TZERO	track zero bit
E17E	C0	246		RNZ	
E17F	37	247		STC .	error flag
E180	C9	248		RET	
		249			
E181		250	SECSET		
E181	AF	251		XRA A	test for
E182	B1	252		ORA C	-zero value
E183	37	253		STC .	error flag
E184	C8	254		RZ .	error return
E185	E6 1F	255		ANI 37Q	trim & clear cry
E187	32 F8 E7	256		STA SECTOR	
E18A	C9	257		RET	
		258			
E18B		259	SEEK		
E18B	79	260		MOV A,C	test for
E18C	FE 4D	261		CPI 77	-track
E18E	3F	262		CMC .	-too large
E18F	D8	263		RC	
E190	32 F9 E7	264		STA TRACK	
E193	C9	265		RET	
		266	*NP		

E194		267	ISSUE		
E194	32 E3 E7	268		STA	ECOUNT+1 update count
E197	CD 96 E3	269		CALL	MEASUR find the index
E19A	OE 01	270		MVI	C,1 start w/sector 1
E19C		271	ISLOOP		
E19C	79	272		MOV	A,C initialize the
E19D	32 FE E3	273		STA	SECREG -sector register
E1A0	3A F8 E7	274		LDA	SECTOR test for
E1A3	B9	275		CMP	C -target sector
E1A4	C8	276		RZ	
E1A5	3E 88	277		MVI	A,RDCMD do a fake
E1A7	CD 5D E3	278		CALL	COMAND -read command
E1AA	DA 20 E2	279		JC	PLEAVE abort on error
E1AD	OC	280		INR	C increment sector no.
E1AE	C3 9C E1	281		JMP	ISLOOP
		282			
E1B1		283	COMNDP		
E1B1	32 FC E3	284		STA	CMDREG start the operation
E1B4	48	285		MOV	C,B initialize block count
E1B5	11 FF E3	286		LXI	D,DATREG data register
E1B8	2A E6 E7	287		LHLD	DMAADR transfer address
E1BB	C9	288		RET	
		289			
E1BC		290	WRITE		
E1BC	CD 33 E2	291		CALL	PREP prepare for write
E1BF	DA 22 E2	292		JC	LEAVE abort operation
E1C2		293	WREENTRY		
E1C2	3E A8	294		MVI	A,WRCMD start a write
E1C4	CD B1 E1	295		CALL	COMNDP
E1C7		296	WRLOOP		
E1C7	7E	297		MOV	A,M load 1st byte of data
E1C8	23	298		INX	H advance pointer
E1C9	12	299		STAX	D write 1st byte of data
E1CA	7E	300		MOV	A,M load 2nd byte of data
E1CB	23	301		INX	H advance pointer
E1CC	12	302		STAX	D write 2nd byte of data
E1CD	7E	303		MOV	A,M load 3rd byte of data
E1CE	23	304		INX	H advance pointer
E1CF	12	305		STAX	D write 3rd byte of data
E1D0	OD	306		DCR	C reduce block count
E1D1	7E	307		MOV	A,M load 4th byte of data
E1D2	23	308		INX	H advance pointer
E1D3	12	309		STAX	D write 4th byte of data
E1D4	C2 C7 E1	310		JNZ	WRLOOP write next 4 bytes
E1D7	21 C2 E1	311		LXI	H,WREENTRY return entry addr
E1DA	C3 FB E1	312		JMP	CBUSY
		313	*NP		

E1DD		314	READ		
E1DD	CD 33 E2	315		CALL PREP	prepare for read
E1E0	DA 22 E2	316		JC LEAVE	abort operation
E1E3		317	RDENTRY		
E1E3	3E 88	318		MVI A,RDCMD	start a read
E1E5	CD B1 E1	319		CALL COMNDP	
E1E8		320	RDLOOP		
E1E8	1A	321		LDAX D	read 1st byte
E1E9	77	322		MOV M,A	store 1st byte
E1EA	23	323		INX H	advance pointer
E1EB	1A	324		LDAX D	read 2nd byte
E1EC	77	325		MOV M,A	store 2nd byte
E1ED	23	326		INX H	advance pointer
E1EE	1A	327		LDAX D	read 3rd byte
E1EF	77	328		MOV M,A	store 3rd byte
E1F0	23	329		INX H	advance pointer
E1F1	0D	330		DCR C	reduce block count
E1F2	1A	331		LDAX D	read 4th byte
E1F3	77	332		MOV M,A	store 4th byte
E1F4	23	333		INX H	advance pointer
E1F5	C2 E8 E1	334		JNZ RDLOOP	read next 4 bytes
E1F8	21 E3 E1	335		LXI H,RDENTRY	return entry addr
		336			
E1FB		337	CBUSY		
E1FB	E5	338		PUSH H	save return
E1FC	21 FC E3	339		LXI H,CSTAT	wait for 1791
E1FF	CD 6C E3	340		CALL BUSY	-to finish command
E202	E6 5F	341		ANI 137Q	error bit mask
E204	CA 21 E2	342		JZ LEAVE-1	no error
E207	FE 10	343		CPI 20Q	premature interrupt
E209	C2 20 E2	344		JNZ PLEAVE	other error type
E20C	3A E2 E7	345		LDA ECOUNT	decrement error
E20F	3D	346		DCR A	-count number 1
E210	FA 17 E2	347		JM STEST	hard interrupt error
E213	32 E2 E7	348		STA ECOUNT	update count
E216	C9	349		RET .	do operation over
E217		350	STEST		
E217	3A E3 E7	351		LDA ECOUNT+1	decrement error
E21A	3D	352		DCR A	-count number 2
E21B	F2 94 E1	353		JP ISSUE	issue a command
E21E	3E 10	354		MVI A,20Q	irrecoverable error!
E220		355	PLEAVE		
E220	37	356		STC .	error flag
E221	E1	357		POP H	adjust the stack
E222		358	LEAVE		
E222	F5	359		PUSH PSW	save the status
E223	3A F6 E7	360		LDA DCREG	control bits
E226	EE 04	361		XRI LOAD	toggle the
E228	32 FA E3	362		STA DCMD	-head load bit
E22B	3A EA E7	363		LDA DRVSEL	enable access to
E22E	32 F9 E3	364		STA DREG	-the data register
E231	F1	365		POP PSW	recover the status
E232	C9	366		RET	
		367	*NP		

E233		368	PREP		
E233	CD E3 E2	369		CALL HDLOAD	load the head
E236	D8	370		RC .	test for drive ready
E237	3A FD E3	371		LDA TRKREG	get old track
E23A	3C	372		INR A	test for head
E23B	CC 70 E1	373		CZ HENTRY	-not calibrated
E23E	D8	374		RC .	seek error?
E23F	21 FD E3	375		LXI H,TRKREG	old track
E242	3A F9 E7	376		LDA TRACK	new track
E245	BE	377		CMP M	test for head motion
E246	23	378		INX H	advance to the
E247	23	379		INX H	-data register
E248	77	380		MOV M,A	save new track
E249	79	381		MOV A,C	turn off data reg
E24A	32 F9 E3	382		STA DREG	-access control bit
E24D	CA 6A E2	383		JZ TVERFY	test for seek
E250	AF	384		XRA A	force a read
E251	32 E9 E7	385		STA HDFLAG	-header operation
E254	3A FA E3	386		LDA DSTAT	get the double
E257	E6 08	387		ANI DSIDE	-sided flag
E259	32 E8 E7	388		STA DSFLAG	save for status
E25C	1F	389		RAR .	shift for
E25D	1F	390		RAR .	-3/6 ms step
E25E	1F	391		RAR .	-rate constant
E25F	C6 18	392		ADI SKCMD	do a
E261	21 00 00	393		LXI H,0	-seek
E264	CD 62 E3	394		CALL CENTRY	-operation
E267	DA 8E E2	395		JC SERROR	seek error?
		396			
E26A		397	TVERFY		
E26A	3A E9 E7	398		LDA HDFLAG	get the force
E26D	B7	399		ORA A	-verify hdr flag
E26E	C2 B9 E2	400		JNZ CHKSEC	no seek & head OK
E271	06 02	401		MVI B,2	verify retry count
E273		402	SLOOP		
E273	3E 1D	403		MVI A,SVCMD	do a verify
E275	CD 5D E3	404		CALL COMAND	-command
E278	E6 99	405		ANI 231Q	error bit mask
E27A	57	406		MOV D,A	save
E27B	CA 95 E2	407		JZ RDHDR	no error!
E27E	3A F6 E7	408		LDA DCREG	denisty control
E281	EE 01	409		XRI DENSITY	flip the density
E283	32 F6 E7	410		STA DCREG	update and
E286	32 FA E3	411		STA DCMD	-change density
E289	05	412		DCR B	decrement retry
E28A	C2 73 E2	413		JNZ SLOOP	-count & test
E28D	7A	414		MOV A,D	restore error bits
E28E		415	SERROR		
E28E	37	416		STC .	error flag
E28F	F5	417		PUSH PSW	save errors
E290	CD 70 E1	418		CALL HENTRY	seek to trk 0
E293	F1	419		POP PSW	recover errors
E294	C9	420		RET	
		421	*NP		

E295		422	RDHDR		
E295	06 0A	423		MVI B,12Q	number of retrys
E297		424	RHLOOP		
E297	11 FF E3	425		LXI D,DATREG	data register
E29A	21 FA E7	426		LXI H,TRACK+1	data pointer
E29D	3E C4	427		MVI A,RACMD	start a read
E29F	32 FC E3	428		STA CMDREG	-header operation
E2A2		429	RHL1		
E2A2	1A	430		LDAX D	get disk data
E2A3	77	431		MOV M,A	store in mem
E2A4	2C	432		INR L	advance pointer
E2A5	C2 A2 E2	433		JNZ RHL1	test end of page
E2A8	21 FC E3	434		LXI H,CSTAT	wait for 1791
E2AB	CD 6C E3	435		CALL BUSY	-to finish cmd
E2AE	B7	436		ORA A	test for errors
E2AF	CA B9 E2	437		JZ CHKSEC	transfer OK?
E2B2	05	438		DCR B	no! - test for
E2B3	C2 97 E2	439		JNZ RHLOOP	-hard error
E2B6	C3 8E E2	440		JMP SERROR	recalibrate
E2B9		441	CHKSEC		
E2B9	3A FD E7	442		LDA SECLN	get the sector
E2BC	4F	443		MOV C,A	-size and setup
E2BD	06 00	444		MVI B,0	-the table offset
E2BF	21 DF E2	445		LXI H,STABLE	sector table
E2C2	09	446		DAD B	sector size pntr
E2C3	3A F8 E7	447		LDA SECTOR	get the sector
E2C6	47	448		MOV B,A	-and save in B
E2C7	86	449		ADD M	compare w/table
E2C8	3E 10	450		MVI A,20Q	error flag
E2CA	D8	451		RC .	error return
E2CB	78	452		MOV A,B	initialize 1791
E2CC	32 FE E3	453		STA SECREG	-sector register
E2CF	3E 20	454		MVI A,40Q	128 byte sector
E2D1	21 05 05	455		LXI H,5:005Q	initialize
E2D4	22 E2 E7	456		SHLD ECOUNT	-error counts
		457			
E2D7		458	SZLOOP		
E2D7	0D	459		DCR C	reduce size count
E2D8	47	460		MOV B,A	sector size to B
E2D9	F8	461		RM .	return on minus
E2DA	17	462		RAL .	double the count
E2DB	B7	463		ORA A	clear the carry
E2DC	C3 D7 E2	464		JMP SZLOOP	
		465			
E2DF		466	STABLE		
E2DF	E5	467		DB 345Q	26 sector diskettes
E2E0	E5	468		DB 345Q	26 sector diskettes
E2E1	F0	469		DB 360Q	15 sector diskettes
E2E2	F7	470		DB 367Q	8 sector diskettes
		471	*NP		

E2E3		472	HDLOAD		
E2E3	21 EB E7	473		LXI	H,DISK new drv ptr
E2E6	4E	474		MOV	C,M save new drv in C
E2E7	23	475		INX	H current drv ptr
E2E8	5E	476		MOV	E,M save old drv in E
E2E9	71	477		MOV	M,C update current drv
E2EA	23	478		INX	H home cmd flag
E2EB	7B	479		MOV	A,E test for
E2EC	B9	480		CMP	C -drive change
E2ED	7E	481		MOV	A,M head load mask
E2EE	36 01	482		MVI	M,HEAD update the mask
E2F0	CA 1B E3	483		JZ	HDCHK no drive change?
E2F3	23	484		INX	H addr of drive table
E2F4	E5	485		PUSH	H save table addr
E2F5	16 00	486		MVI	D,O set up the
E2F7	42	487		MOV	B,D -offset address
E2F8	19	488		DAD	D calculate the
E2F9	19	489		DAD	D -parameter addr
E2FA	3A F6 E7	490		LDA	DCREG save the
E2FD	77	491		MOV	M,A density status
E2FE	23	492		INX	H track pointer
E2FF	11 FD E3	493		LXI	D,TRKREG 1791 trk reg
E302	1A	494		LDAX	D get current track
E303	77	495		MOV	M,A save in the table
E304	E1	496		POP	H beginning of table
E305	09	497		DAD	B new drive
E306	09	498		DAD	B -table pointer
E307	7E	499		MOV	A,M get density status
E308	32 F6 E7	500		STA	DCREG update DCREG
E30B	23	501		INX	H get the old
E30C	7E	502		MOV	A,M -track number
E30D	12	503		STAX	D -and update 1791
E30E	3E 7F	504		MVI	A,177Q drive select bits
E310		505	DSROT		
E310	07	506		RLC	. rotate to
E311	0D	507		DCR	C -select the
E312	F2 10 E3	508		JP	DSROT -proper drive
E315	E6 7F	509		ANI	177Q set the run bit
E317	32 EA E7	510		STA	DRVSEL save in drv reg
E31A	AF	511		XRA	A force a head load
		512	*NP		

E31B		513	HDCHK		
E31B	21 FA E3	514		LXI H,DSTAT	test for
E31E	A6	515		ANA M	-head loaded
E31F	32 E9 E7	516		STA HDFLAG	save the head
E322	F5	517		PUSH PSW	-loaded status
E323	3A EA E7	518		LDA DRVSEL	get current drive
E326	4F	519		MOV C,A	save
E327	3A F7 E7	520		LDA SIDE	get current side
E32A	2F	521		CMA .	-and merge
E32B	A1	522		ANA C	-with drive select
E32C	32 F9 E3	523		STA DREG	select drive & side
E32F	EE 40	524		XRI ACCESS	toggle access bit
E331	4F	525		MOV C,A	save for PREP routine
E332	3A F6 E7	526		LDA DCREG	den & head cntl bits
E335	47	527		MOV B,A	save
E336	3A F9 E7	528		LDA TRACK	get the new track
E339	D6 01	529		SUI 1	force single
E33B	9F	530		SBB A	-density
E33C	3D	531		DCR A	-if track = 0
E33D	2F	532		CMA .	compliment
E33E	B0	533		ORA B	merge w/control bits
E33F	77	534		MOV M,A	load head & set density
E340	F1	535		POP PSW	head load status
E341	C2 4F E3	536		JNZ RDYCHK	conditionally
E344	E5	537		PUSH H	-wait for head
E345	2A E4 E7	538		LHLD TIMER	-load time out
E348		539	TLOOP		
E348	2B	540		DCX H	count down
E349	7C	541		MOV A,H	-40 ms for
E34A	B5	542		ORA L	-head load
E34B	C2 48 E3	543		JNZ TLOOP	-time out
E34E	E1	544		POP H	
E34F		545	RDYCHK		
E34F	7E	546		MOV A,M	test for
E350	E6 80	547		ANI READY	-drive ready
E352	C0	548		RNZ	
E353		549	UNLOAD		
E353	3A F6 E7	550		LDA DCREG	force a
E356	F6 06	551		ORI ULOAD	-head
E358	77	552		MOV M,A	-unload
E359	3E 80	553		MVI A,READY	set drive
E35B	37	554		STC .	-not ready
E35C	C9	555		RET .	-error flag
		556			
E35D		557	COMAND		
E35D	2A E4 E7	558		LHLD TIMER	get index count
E360	29	559		DAD H	-and multiply
E361	29	560		DAD H	-by four
E362		561	CENTRY		
E362	EB	562		XCHG .	save in D-E pair
E363	21 FC E3	563		LXI H,CSTAT	issue command
E366	77	564		MOV M,A	-to the 1791
E367		565	NBUSY		
E367	7E	566		MOV A,M	wait
E368	1F	567		RAR .	-for the
E369	D2 67 E3	568		JNC NBUSY	-busy flag
		569	*NP		

E36C		570	BUSY		
E36C	7E	571		MOV A,M	test for
E36D	1F	572		RAR .	-device busy
E36E	7E	573		MOV A,M	restore status
E36F	DO	574		RNC .	return if not busy
E370	C3 76 E3	575		JMP PATCH+3	jump around patch
E373		576	PATCH		
E373	C3 E3 E2	577		JMP HDLOAD	patch for old ATE
E376	1B	578		DCX D	test for
E377	7A	579		MOV A,D	-two disk
E378	B3	580		ORA E	-revolutions
E379	C2 6C E3	581		JNZ BUSY	47 machine cycles
E37C	5E	582		MOV E,M	get error code
E37D	E5	583		PUSH H	save cmd address
E37E	23	584		INX H	track register
E37F	56	585		MOV D,M	save present track
E380	3A EA E7	586		LDA DRVSEL	control bits
E383	EE 80	587		XRI RSTBIT	reset the 1791
E385	32 F9 E3	588		STA DREG	-controller to
E388	EE CO	589		XRI STBITS	-clear the
E38A	E3	590		XTHL .	-command busy
E38B	32 F9 E3	591		STA DREG	-fault condition
E38E	36 DO	592		MVI M,CLRCMD	force interrupt
E390	E3	593		XTHL .	restore the
E391	72	594		MOV M,D	-the track reg
E392	E1	595		POP H	restore the stack
E393	7B	596		MOV A,E	error code to A
E394	37	597		STC .	-error flag
E395	C9	598		RET	
		599			
E396		600	MEASUR		
E396	11 00 00	601		LXI D,0	initialize count
E399	21 FA E3	602		LXI H,DSTAT	status port
E39C	0E 10	603		MVI C,INDEX	index bit flag
E39E		604	INDXLO		
E39E	7E	605		MOV A,M	wait for
E39F	A1	606		ANA C	-index
E3A0	CA 9E E3	607		JZ INDXLO	-pulse high
E3A3		608	INDXHI		
E3A3	7E	609		MOV A,M	wait for
E3A4	A1	610		ANA C	-index
E3A5	C2 A3 E3	611		JNZ INDXHI	-pulse low
E3A8		612	INDXCT		
E3A8	13	613		INX D	advance count
E3A9	E3	614		XTHL .	four dummy
E3AA	E3	615		XTHL .	-instructions
E3AB	E3	616		XTHL .	-to lengthen
E3AC	E3	617		XTHL .	-the delay
E3AD	7E	618		MOV A,M	wait for
E3AE	A1	619		ANA C	-the index
E3AF	CA A8 E3	620		JZ INDXCT	-to go high
E3B2	C9	621		RET .	98 machine cycles
		622	*NP		

E3B3		623	DENFIX		
E3B3	79	624		MOV A,C	trim the
E3B4	E6 01	625		ANI 1	-excess bits
E3B6	2F	626		CMA .	compliment and
E3B7	47	627		MOV B,A	-save in B
E3B8	21 EB E7	628		LXI H,DISK	new disk ptr
E3BB	5E	629		MOV E,M	get disk no.
E3BC	16 00	630		MVI D,O	offset addr
E3BE	23	631		INX H	current disk ptr
E3BF	7E	632		MOV A,M	move to ACC
E3C0	AB	633		XRA E	cmpr old w/new
E3C1	F5	634		PUSH PSW	save status
E3C2	23	635		INX H	disk table
E3C3	23	636		INX H	-address
E3C4	19	637		DAD D	add the
E3C5	19	638		DAD D	-offset
E3C6	7E	639		MOV A,M	get parameters
E3C7	F6 01	640		ORI 1	mask off density
E3C9	A0	641		ANA B	set new density
E3CA	77	642		MOV M,A	update parameters
E3CB	F1	643		POP PSW	test new=old?
E3CC	C0	644		RNZ	
E3CD	7E	645		MOV A,M	update CDISK
E3CE	32 F6 E7	646		STA DCREG	-also
E3D1	C9	647		RET	
		648			
E3D2		649	TIMOUT		
E3D2	21 00 00	650		LXI H,O	time-out delay
E3D5		651	TILOOP		
E3D5	2B	652		DCX H	decrement count
E3D6	7C	653		MOV A,H	test for delay
E3D7	B5	654		ORA L	-count equal zero
E3D8	E3	655		XTHL .	long NOP
E3D9	E3	656		XTHL .	-instruction
E3DA	C2 D5 E3	657		JNZ TILOOP	
E3DD	C9	658		RET	
		659			
E3DE		660	SBEGIN		
E3DE	E5	661		PUSH H	
E3DF	21 E2 E3	662		LXI H,DSTALL	
E3E2		663	DSTALL		
E3E2	E9	664		PCHL	
E3E3	E1	665		POP H	
E3E4	C9	666		RET	
		667			
E3E5		668	SIDEXF		
E3E5	79	669		MOV A,C	get the side bit
E3E6	E6 01	670		ANI 1	trim the excess
E3E8	17	671		RAL .	move the bit
E3E9	17	672		RAL .	-to the side
E3EA	17	673		RAL .	-select bit
E3EB	17	674		RAL .	-position
E3EC	32 F7 E7	675		STA SIDE	save side bit
E3EF	C9	676		RET	
		677	*NP		

E3F0		678	PWRJMP			
E3F0	00	679	NOP	.		power-on
E3F1	00	680	NOP	.		-jump
E3F2	00	681	NOP	.		-sequence
E3F3	00	682	NOP	.		-with NOP
E3F4	00	683	NOP	.		-padding
E3F5	C3 00 E0	684	JMP	DBOOT		
		685				
E3F8	0008	686	DS	10Q		I/O locations
		687				
E7C9		688	AORG	RAM+3:311Q		
		689				
E7C9	0019	690	STACK	DS	31Q	
		691				
E7E2	00 00	692	ECOUNT	DW	0	error count cells
E7E4	00 18	693	TIMER	DW	30:000Q	head load time out
E7E6	00 E7	694	DMAADR	DW	RAM+300H	dma address
E7E8	08	695	DSFLAG	DB	10Q	
E7E9	00	696	HDFLAG	DB	0	read header flag
E7EA	7E	697	DRVSEL	DB	176Q	drive select constant
E7EB	00	698	DISK	DB	0	new drive
E7EC	08	699	CDISK	DB	10Q	current disk
E7ED	00	700	TZFLAG	DB	0	home cmd indicator
E7EE	09	701	DOPRAM	DB	11Q	drive 0 parameters
E7EF	FF	702	DOTRK	DB	377Q	drive 0 track no
E7F0	09	703	D1PRAM	DB	11Q	drive 1 parameters
E7F1	FF	704	D1TRK	DB	377Q	drive 1 track no
E7F2	09	705	D2PRAM	DB	11Q	drive 2 parameters
E7F3	FF	706	D2TRK	DB	377Q	drive 2 track no
E7F4	09	707	D3PRAM	DB	11Q	drive 3 parameters
E7F5	FF	708	D3TRK	DB	377Q	drive 3 track no
E7F6	09	709	DCREG	DB	11Q	current parameters
E7F7	00	710	SIDE	DB	0	new side
E7F8	01	711	SECTOR	DB	1	new sector
E7F9	00	712	TRACK	DB	0	new track
E7FA	00	713	TRKNO	DB	0	disk
E7FB	00	714	SIDENO	DB	0	-sector
E7FC	00	715	SECTNO	DB	0	-header
E7FD	00	716	SECTNO	DB	0	-data
E7FE	00	717	CRCLO	DB	0	-buffer
E7FF	00	718	CRCHI	DB	0	

Morrow Designs, Inc.


**Thinker
Toys™**

5221 Central Avenue, Richmond, CA 94804 (415) 524-2101

LIMITED WARRANTY

Morrow Designs Inc. warrants its products to be free from defects in workmanship and material for the period indicated. This warranty is limited to the repair or replacement of parts only and liability is limited to the purchase price of the product. The warranty is void if, in the sole opinion of Morrow Designs Inc., the product has been subject to abuse, misuse, unauthorized modification, improper assembly, non-conformance to assembly directions, or if the unit is used in any other manner than intended.

KITS - Parts, including the printed circuit boards, purchased in kit form are warranted for a period of ninety (90) days from the invoice/purchase date. If a board, which was purchased in kit form, is returned for testing or repair, a minimum service charge of \$35. will be assessed.

ASSEMBLED BOARDS - Parts, including the printed circuit boards, purchased as factory assemblies, are warranted for a period of six (6) months from the invoice/purchase date. Out-of-Warranty boards returned for testing of repair will be assessed a minimum of \$35. service charge. If the charge to repair will exceed \$35., the customer will be notified prior to the actual repair.

ELECTROMECHANICAL PERIPHERALS - Peripheral equipment, such as floppy disk drives, hard disk drives, etc., not manufactured by Morrow Designs Inc. have warranties which vary according to the manufacturer. In most cases, Morrow Designs Inc. provides a warranty equal to or greater than the original manufacturer. Please contact the factory for individual warranty information. Warranty information for each device is included with the equipment when it is shipped.

RETURN PROCEDURE - A COPY OF THE INVOICE OR PROOF OF ORIGINAL PURCHASE IS REQUIRED AND MUST ACCOMPANY THE ITEM FOR IN-WARRANTY SERVICE. Items returned without proof of original purchase will be sent back, shipping charges collect. A description of the problem must accompany the returned item. Shipment must be made prepaid to Morrow Designs Inc. Repaired items will be shipped via U.P.S. surface. Shipment by air requires payment of the additional charges. Morrow Designs Inc. is not responsible for any consequential damages or for damage incurred in transit.

The foregoing warranty is in lieu of all other warranties either expressed or implied and, in any event, is limited to product repair or replacement.

Effective February 1, 1980

Specifications, terms, and pricing are subject to change without notice.

LIMITED WARRANTY

DISCUS 1 and DISCUS 2D Systems

This addendum to Morrow Designs Inc. Limited Warranty applies to the Shugart Associates Model 800/801 Floppy Disk Drives as used in the DISCUS 1 and 2D Disk systems.

Parts and labor for a floppy disk drive purchased from Morrow Designs Inc. are warranted for a period of forty-five (45) days from the invoice/purchase date. For a period of one (1) year from the invoice/purchase date, parts are warranted. A fixed fee of \$55. will be charged for labor. After one (1) year current rates for parts and labor will be charged.

LIMITED WARRANTY

DISCUS 2+2 Systems

This addendum to Morrow Designs Inc. Limited Warranty applies to the EX-CELL-0 Corporation Remex Model RFD4000 Floppy Disk Drives as used in the DISCUS 2+2 System.

Parts and labor for a floppy disk drive purchased from Morrow Designs Inc. are warranted for a period of six (6) months from the invoice/purchase date. After six (6) months current rates for parts and labor will be charged.

