# Disk Jockey 2D Model B
# Technical Reference Manual

Revision 1
December 1982

## DISCLAIMER

No representations or warranties, express or implied, are made with respect to the contents hereof, including, but not limited to, the implied warranty of merchantability or fitness for a particular purpose. Further, Morrow Designs, Inc., reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify any person of such revision.

CP/M 2.2 is an operating system developed by Digital Research; all references to this product should be so noted.

Morrow Designs
600 McCormick St.
San Leandro, CA 94577
(415) 430-1970

Revision 1-Disk Jockey 2D Model B Technical Reference Manual-12/82

## IMPORTANT WARRANTY INFORMATION

## LIMITED WARRANTY

Morrow Designs, Incorporated, warrants its products to be free from defects in workmanship and materials for the periods indicated below. This warranty is limited to the repair and replacement of parts only and liability is limited to the wholesale list price of the product.

> Limitation of Liability: The foregoing warranty is in lieu of all other warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event will Morrow Designs, Incorporated, be liable for consequential damages even if Morrow Designs, Incorporated, has been advised of the possibility of such damages.

This warranty is void if, in the sole opinion of Morrow Designs, Incorporated, the product has been subject to abuse or misuse.

Circuit boards - Parts, including the printed circuit board, purchased as factory assemblies, are warranted for a period of ninety (90) days from the original invoice/purchase date.

Electro-mechanical peripherals - Peripheral equipment such as floppy disk drives, etc., not manufactured by Morrow Designs, Incorporated, carry their own manufacturers' warranties. Please contact Morrow Designs' Customer Service Department for information regarding these peripheral devices.

Software/Firmware - Morrow Designs, Incorporated, makes no representations or warranties whatsoever with respect to software or firmware associated with its products and specifically disclaims any implied or expressed warranty of fitness for any particular purpose or compatibility with any hardware, operating system, or software/firmware. Morrow Designs, Incorporated, reserves the right to alter or update any program, publication or manual without obligation to notify any person of such changes.

# WARRANTY RETURN PROCEDURE

Should a customer experience a defect in either workmanship or materials during the warranty period, Morrow Designs, Incorporated will replace or repair the product at its expense only if the product is promptly returned to Morrow Designs, Incorporated for repair or replacement, and the following procedure for returning the product is followed:

1. Phone Morrow Design's Customer Service Department at (415) 430-1970. Inform the customer service staff of the nature of your problem and obtain a RETURN AUTHORIZATION NUMBER. No return shipment will be accepted without this number.

2. Repack the equipment and ship it to Morrow Designs, Incorporated, care of the Customer Service Department, 600 McCormick St., San Leandro, California, 94577. All freight charges must be prepaid by the customer, as well as all related charges, such as customs clearance and documentation. CODs WILL BE REFUSED. Indicate the RETURN AUTHORIZATION NUMBER on the waybill and shipping label. Include with the equipment a copy of proof of purchase showing the date the equipment was purchased. Any shipment received without proof of purchase will be billed as non-warranty repairs. Please also include a brief written description of the problem experienced with the equipment.

3. Morrow Designs, Incorporated, will repair or replace defective items and return the product to the customer via UPS surface, prepaid rates. Any other form of delivery or shipment required or requested by the customer is at the sole expense of the customer.

4. SHIPPING DAMAGE: (a) Morrow Designs, Incorporated, is not responsible for damage to goods in transit. (b) If you return a product because of shipping damage, have the product inspected by the carrier before returning it to us. Failure to do so may result in denial of your claim by the carrier. (c) Always ship the product in its original packing material. If the original packing material has been damaged or lost, new packing material can be purchased from Morrow Designs. Many shippers will not honor damage claims if the product is not adequately packaged.

5. Morrow Designs, Incorporated, is not responsible for the integrity of any data recorded on any media returned for service or repair. It is the responsibility of the user to back-up all necessary information.

Disk Jockey 2D Model B
Technical Reference Manual
Revision 1

## Table of Contents

## Table of Contents, Cont.

## Table of Contents, Cont.

## Appendices

## List Of Figures

## List of Figures, Cont.

## List of Tables

# 1. INTRODUCTION

## 1.1. INTRODUCTION

The Disk Jockey 2D Model B (DJ2D/B) disk controller, when combined with a CPU board, memory and CRT terminal presents the user with a floppy based S-100 computer system. Morrow Designs supplies the CP/M Version 2.2 operating system with systems configured for the DJ2D/B systems. Normal configurations use the serial port on the DJ2D/B as the CP/M console device.

CP/M is the industry standard operating system for 8080/Z80 based systems. The DJ2D/B reads and writes 8 inch single and double density IBM 3740 compatible diskettes allowing the user access to CP/M programs which have been recorded on other systems. Although single density allows storage of only 256 Kbytes per diskette, it is currently the only standard of compatibility in 8 inch diskettes. All diskettes recorded on your DJ2D/B system which are to be transferred to other CP/M machines should be recorded in single density format. Double density allows as much as 620K bytes per diskette.

## 1.2. MANUAL ORGANIZATION

This manual was written as a technical reference guide for persons installing the DJ2D/B in their computer systems. Section 2 provides an overview of the DJ2D/B features; hardware, software and general features and capabilities are outlined. Section 3 covers installation of the DJ2D/B in a "standard environment". Section 4 discusses system start-up in the standard environment, and procedures for system power-down. Section 5 provides information to enable operation in a "non-standard" environment. Section 6 details the DJ2D/B hardware registers, Section 7 covers the DJ2D/B software functions, and Section 8 incorporates some of the unique hardware and software functions in "Hardware and Software Notes".

Appendices provide information on power-on jump logic, drive select registers, pin-outs for the DJ 2D/B I/O and serial ports, a troubleshooting guide, schematics, and software listings. Schematics, the DJ2D/B parts list and a detailed subject index are also provided.

# 2. SYSTEM OVERVIEW

## 2.1. HARDWARE FEATURES

The DJ controller board features four distinct hardware features:

1. The controller can be connected to any 8 inch floppy disk drive that is plug compatible with the Shugart 800/850 series of floppy disk drives. A 1791 (or equivalent) floppy disk controller reads and writes data in either single density FM code or double density MFM code with write precompensation.

2. A serial interface (1602 UART) allows communication with a terminal device at TTY 20ma current loop or RS-232 levels.

3. On-board software and hardware makes first-time power up easy.

4. Bank Select logic allows the board to be enabled or disabled under software control. This logic also can be programmed to force the board to be enabled or disabled during power-on/reset sequences.

## 2.2. SOFTWARE FEATURES

The Disk Jockey software is configured so that all that is necessary to get the system running is a central processing unit (CPU), memory board(s) and a terminal. All the "tools" needed to customize the system software are also included. Each Morrow Designs system diskette (included with the controller) includes:

* CP/M 2.2 operating system

* Source to CP/M BIOS

* CP/M transient commands

* Disk formatting program (with source file)

## 2.3. GENERAL INFORMATION

The DJ2D/B plugs into an S-100 bus slot in a system with an 8080, 8085, or Z80 (1.7MHz - 6MHz) CPU (see Section 3, Seating the DJ2D/B Board). 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 DJ2D/B is provided for attaching a terminal device (see Section 3, I/O Connectors).

# 3. INSTALLING THE DJ2D/B - STANDARD ENVIRONMENT

## 3.1. INSTALLATION OVERVIEW

The DJ2D/B has been configured at the factory to work in the following operating environment:

* CPU does its own power-on jump.

* Memory board responds to PHANTOM signal.

* Communication rate for the serial port is set at 9600 baud; protocol is 8 data bits, 2 stop bits, parity inhibited.

* Board is not set for bank select or interrupt drivers.

NOTE: If your system does not meet these requirements, refer to Section 5, OPERATION IN A NON-STANDARD ENVIRONMENT.

## 3.2. INSTALLATION REQUIREMENTS

Installation of the DJ2D/B controller board requires the following:

1. Setting the two 8-position DIP switches.

2. Installing jumper options.

3. Seating the board in the system's S-100 bus slot.

4. Connecting the cable from the DJ2D/B to the disk drive.

5. Connecting the cable to the DJ2D/B serial port connector and to your terminal.

Each step is covered separately within this section.

WARNING: NEVER INSERT OR REMOVE A BOARD WITH SYSTEM POWER ON! THIS WILL DAMAGE THE BOARD.

## 3.3. EXAMINE THE BOARD

Before installing the DJ controller board, examine the board for shipping damage. If shipping damage exists, note the condition on the waybill, notify the carrier, and check the Warranty Return Procedure at the front of this manual for instructions on returning the board to Morrow Designs.

## 3.4. SETTING THE SWITCHES

Each DJ2D/B controller board contains two (2) 8-position DIP switches, labelled SW1 and SW2. The location of these switches on the board are 5D and 13C, respectively.

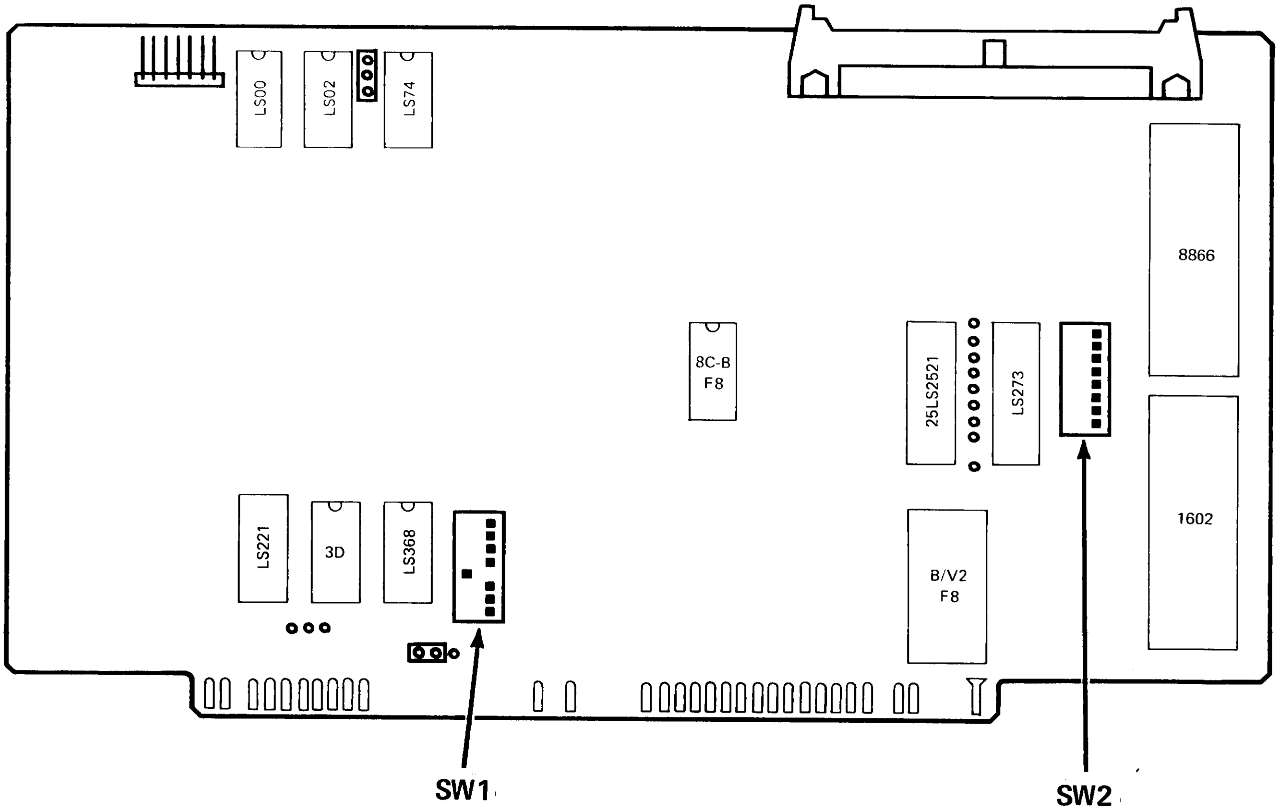


**Figure 3-1: DJ2D/B DIP Switch Locations**

Each switch has eight paddles. The setting of each paddle, and the functions they perform, are described and illustrated in this section.

### 3.4.1. Switch 1

Switch 1 controls the DJ2D/B Power-On Jump logic, Phantom logic, and operation of the board at various bus speeds. Standard location for the DJ2D/B controller is F800H.

- Paddles 1 through 5 control the Power-On Jump (POJ) address.
- Paddle 8 enables/disables POJ circuitry.
- Paddle 6 affects the PHANTOM* line.
- Paddle 7 is set according to system bus speed.

Most computer systems have some means of performing power-on jump to specific memory locations. In this case, the POJ circuitry of the DJ2D/B is not required, and all Switch 1 paddles may be left in the OFF position. If the CPU cannot perform POJ (to F800H), refer to Section 5, OPERATING IN A NON-STANDARD ENVIRONMENT. Refer also to your CPU manual for instructions on setting the POJ address.

The figure below illustrates the settings for Switch 1 as shipped from the factory:

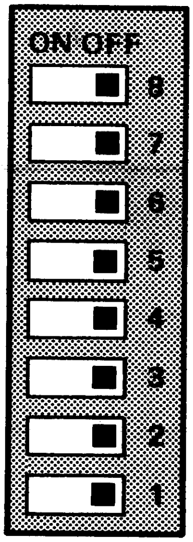


Figure 3-2: Factory Settings SW1 (Located at 5D)

### 3.4.2. Switch 2

Switch 2 controls the console serial port.

- Paddles 1 through 4 set the baud rate.
- Paddle 5 selects word length.
- Paddle 6 controls the number of stop bits.
- Paddles 7 and 8 affect parity checking.

The serial port can communicate with devices at speeds from 50 baud to 19.2K baud (selectable by switches at board location 13C). In addition to baud rate selection, the user can select the number of data bits, stop bits and parity via these switches. The vast majority of users will connect their terminal to this port for the CP/M console.

The figure below illustrates the settings for Switch 2 as shipped from the factory:

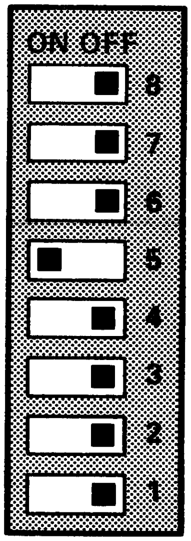


Figure 3-3: Factory Settings SW 2 (located at 13C)

Note: User's wishing to run their terminals at 19,200 baud need simply to turn paddle 4 OFF.

## 3.5. INSTALLING JUMPER OPTIONS

The DJ2D/B has four jumper options:

J4 - controls whether the DJ2D/B powers up active or inactive.

J2 - determines whether the DJ2D/B generates PHANTOM signal (Revision 3 or later only).

J1A - determines interrupt upon completion of commands.

J3A - controls bank select logic.

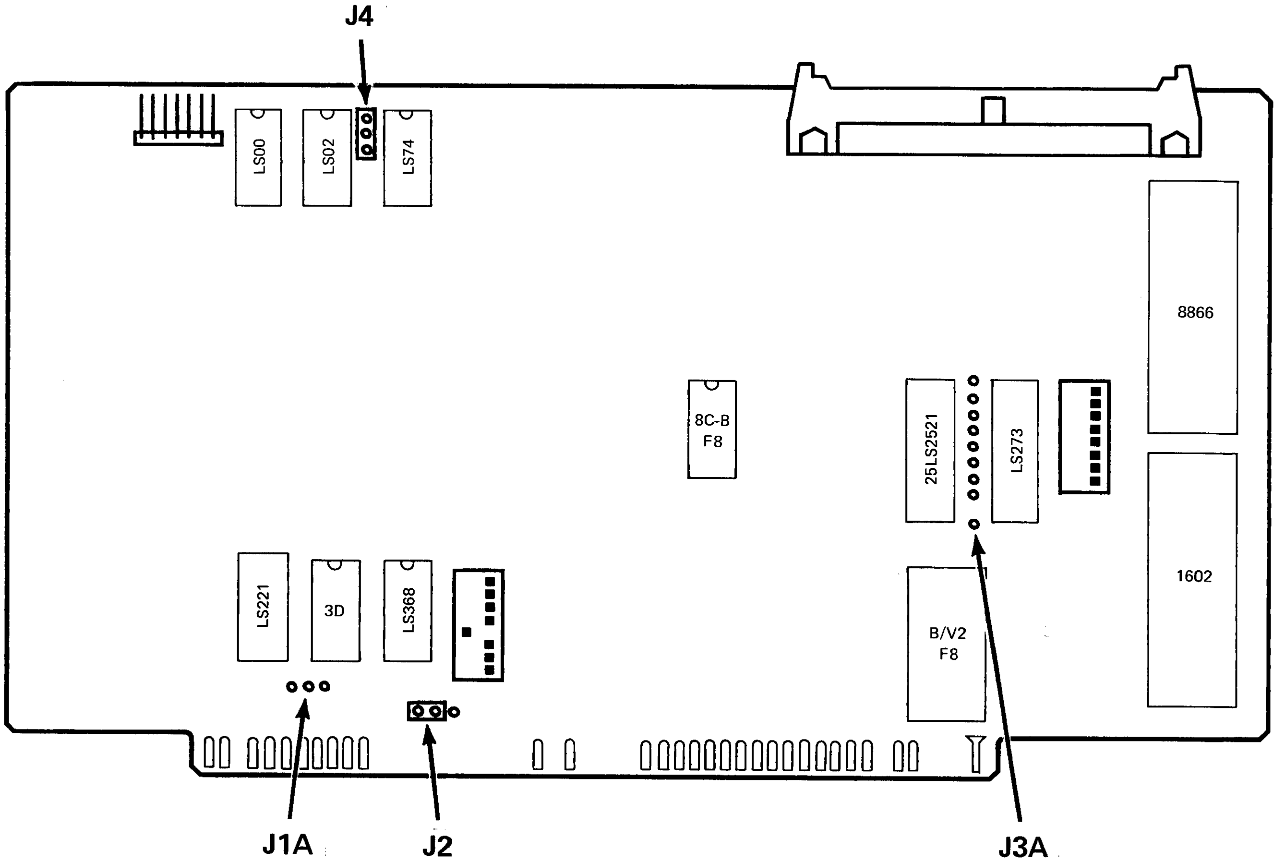


Figure 3-4: DJ2D/B Jumper Locations

## 3.6. FACTORY SET JUMPER SETTINGS

J4 - Jumper installed between A and B (refer to Figure 2-1). This allows the board to become active on power up or reset.

J2 - Jumper installed, allowing the board to generate S-100 PHANTOM signals which disable any memory board sharing the DJ2D/B's memory space.

J1A - No jumpers installed (refer to Section 5 on INTERRUPTS in OPERATION IN A NON-STANDARD ENVIRONMENT).

J3A - No jumpers installed (refer to Section 5 on BANK SELECT in OPERATION IN A NON-STANDARD ENVIRONMENT).

## 3.7. SEATING THE DJ2D/B BOARD

The DJ2D/B controller board plugs into an S-100 bus slot in a system with an 8080, 8085 or Z-80 (1.75MHz to 6MHz) CPU.

Place the bottom of the board (the side with the gold S-100 edge connectors) in the system's S-100 bus slot. You should be facing the component side of the board (the side containing the silk-screened legend).

NOTE: The S-100 bus is keyed so the board can be inserted with one orientation. Home-built systems, however, might not have card guides or a cage, and care must be taken to maintain proper orientation (Pin 1 is the leftmost finger of the card edge connector, as viewed from the component side of the board, edge connector fingers down.

Placing your thumbs on the top of the board, rock the board from side to side to seat the DJ2D/B securely in the S-100 slot.

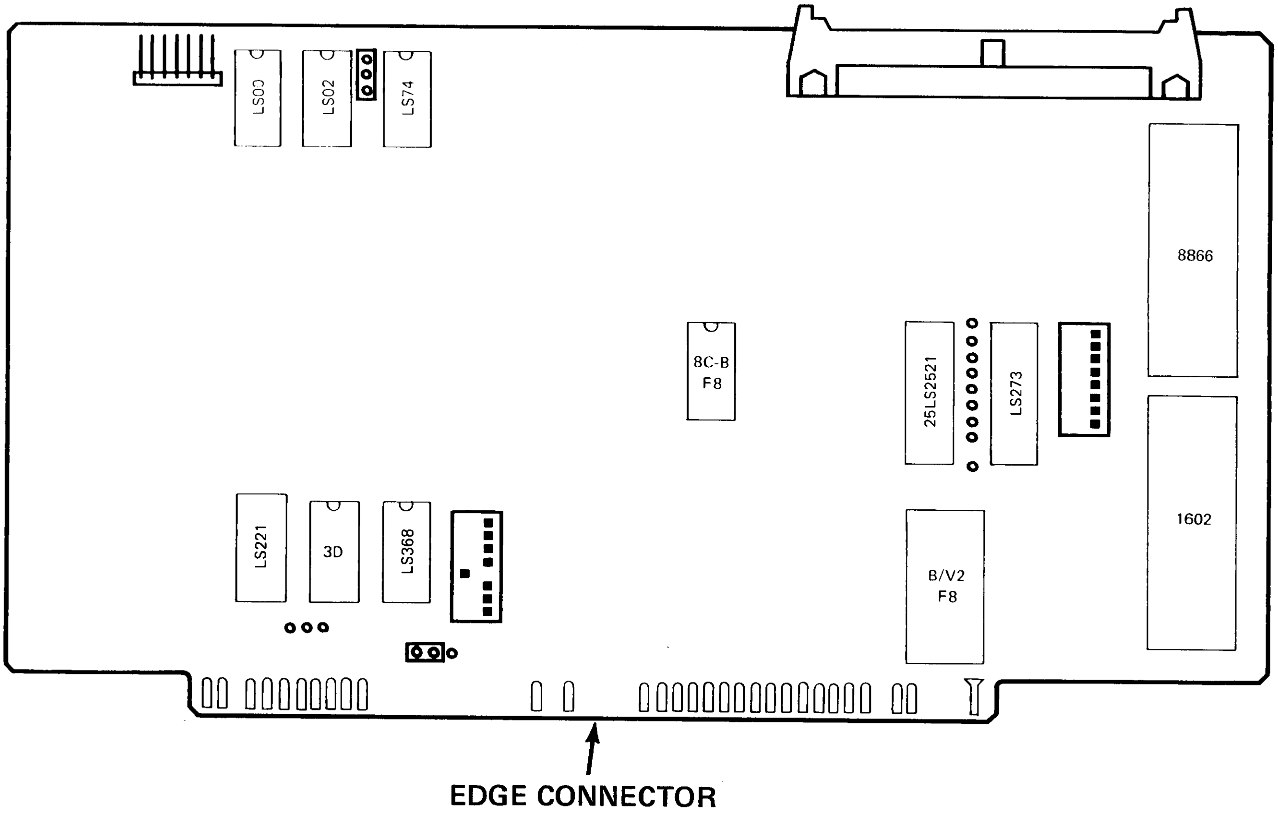


Figure 3-5: Seating the S-100 Board

## 3.8. I/O CONNECTORS

P1 and P2 (see Figure 3-6) are the two connectors provided on each DJ2D/B board. P1 is a 50 pin header for connection to a floppy disk drive; P2 is a 7 pin header used to connect a serial I/O device.

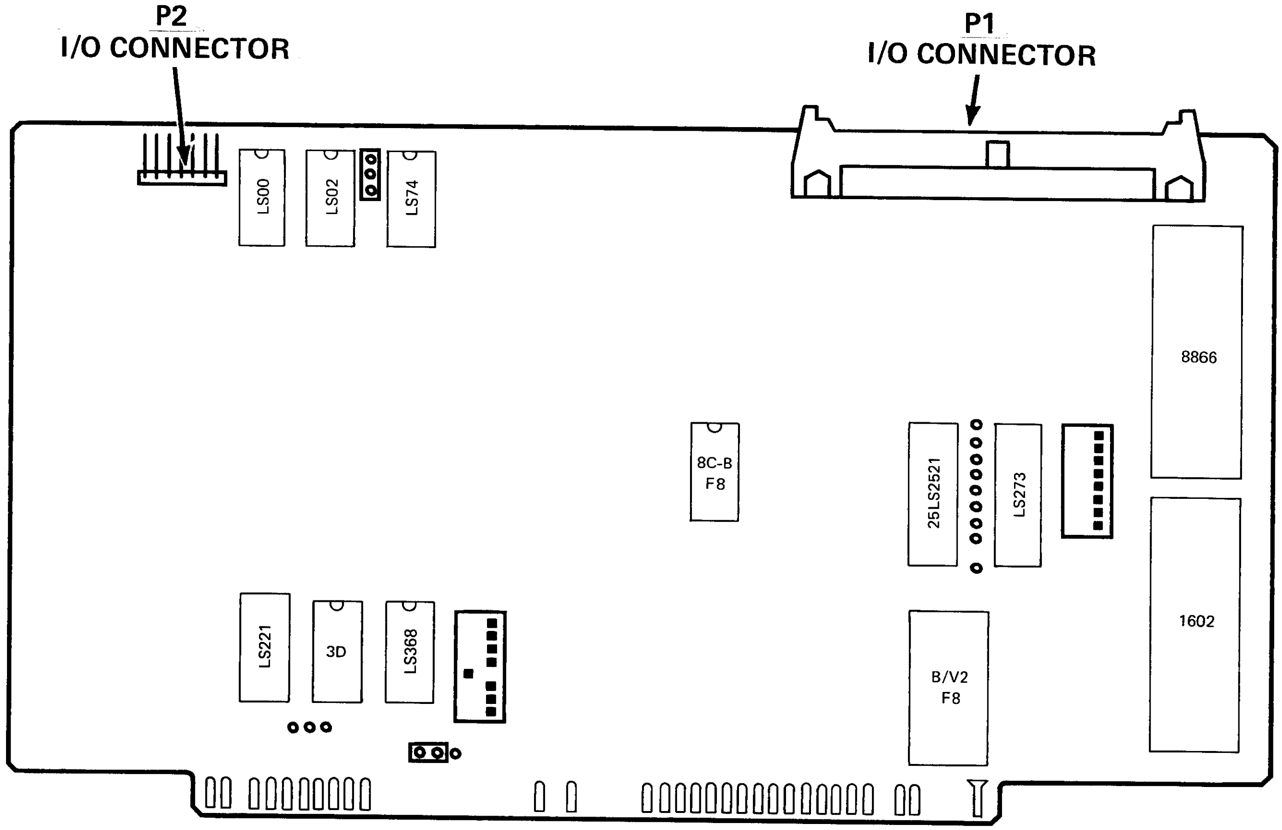


Figure 3-6: DJ2D/B I/O Connectors

## 3.9. CABLE CONNECTION

The 50 pin flat ribbon cable required for connection to the floppy disk drive may either be included with the purchase of your floppy disk drive, or ordered from Morrow Designs. This cable must not exceed 8 feet in length. If the DJ2D/B is purchased separately (not as part of a Morrow Designs system), a seven pin molex connector kit is provided for cabling the DJ2D/B and terminal device (cable must not exceed 25 feet at 9600 baud).

Customers that purchase a Morrow Designs Discus system receive the cables required for connecting the DJ2D/B to the I/O devices (a 50 pin flat ribbon cable and 7 pin molex connector kit). This section describes the cabling procedure for floppy disk drive connection (refer to Appendix C for assembling the molex connector for the serial port).

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 (see Figure 3-7). The cable must connect to each and every drive on the system. Thus, Pl Pin 50 on the DJ2D/B 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.

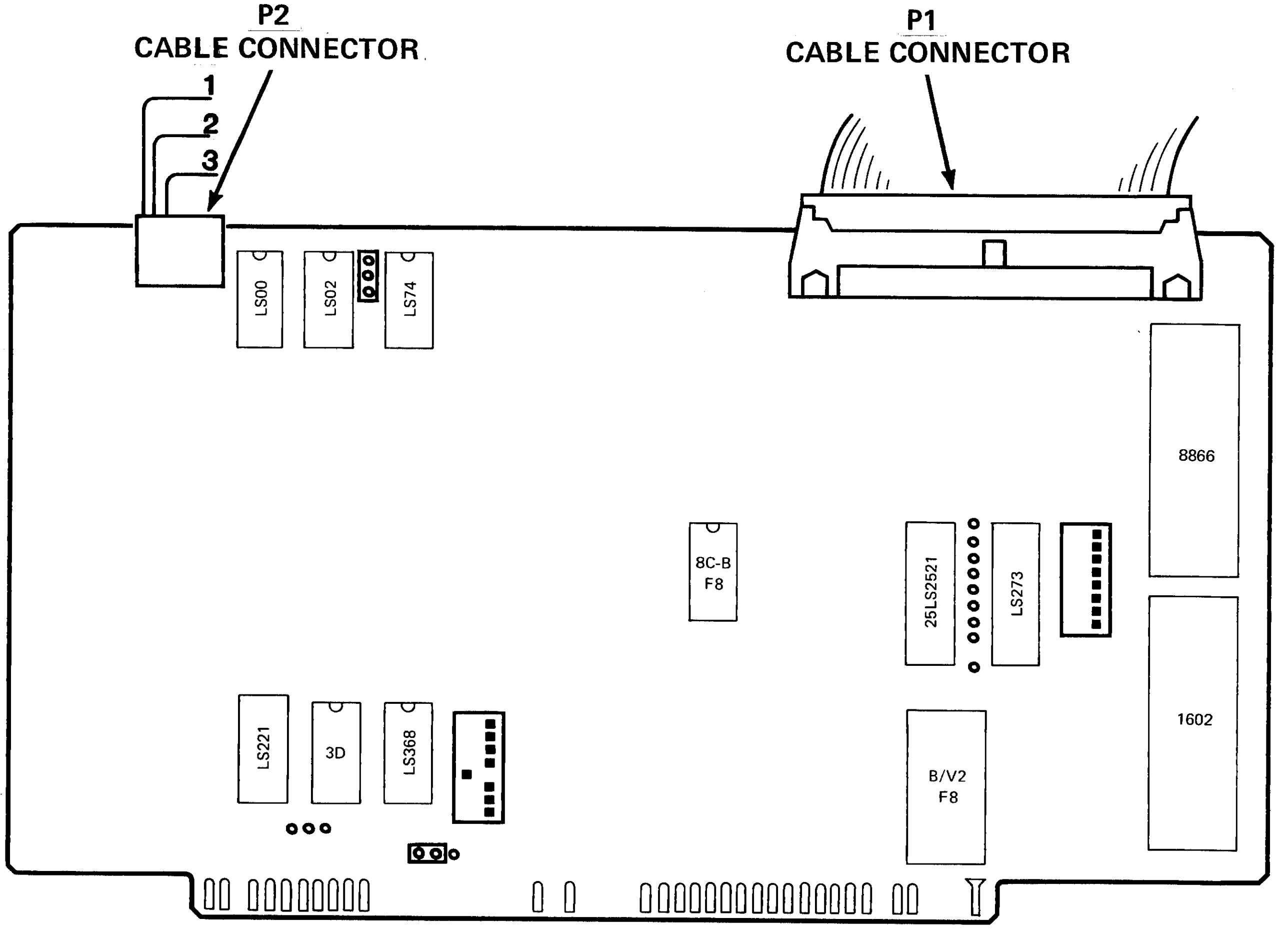


Figure 3-7: Cable Connections

Note: Incorrect disk cable installation will cause all drives in the system to become selected. This will also activate the drive write logic and destroy data on diskettes that are in these drives. Caution at this step will prevent this from occurring.

### 3.9.1. Single Drive Connections

Follow these instructions to install your DJ2D/B:

1. Turn off power to computer.

2. If not already done, install the DJ2D/B PC board into the motherboard. Press board into a rear slot GENTLY, but firmly.

Internal Connections:

1. Locate the 50-pin socket on the left side (rear view) of the DJ2D/B board labeled P1 (see Figure 3-8).

2. Connect a 50-pin cable to P1 on the DJ2D/B, matching pin-1 on the connector to pin-1 on the socket. Connection is correct if ribbon leads down and toward the back panel (see Figure 3-8).

External Connections:

3. Connect the end of this 50-pin cable to the 50-pin socket on the floppy disk drive rear panel.

### 3.9.2. Multiple Drive Connections

Multiple drives are connected to the DJ2D/B in a daisy chain fashion (see Figure 3-8). As illustrated, Drive D is located at one end of the cable and is the only terminated drive on the cable. The drive at the <u>END</u> of the cable is the only one which must be terminated (regardless of selection). 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 (see Appendix A for location of drive terminators).

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. Make sure the boards for your disk drives have been properly configured for multiple drive connection. Strapping a drive for termination and drive selection is documented in the OEM 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 must be filled for the system to function.

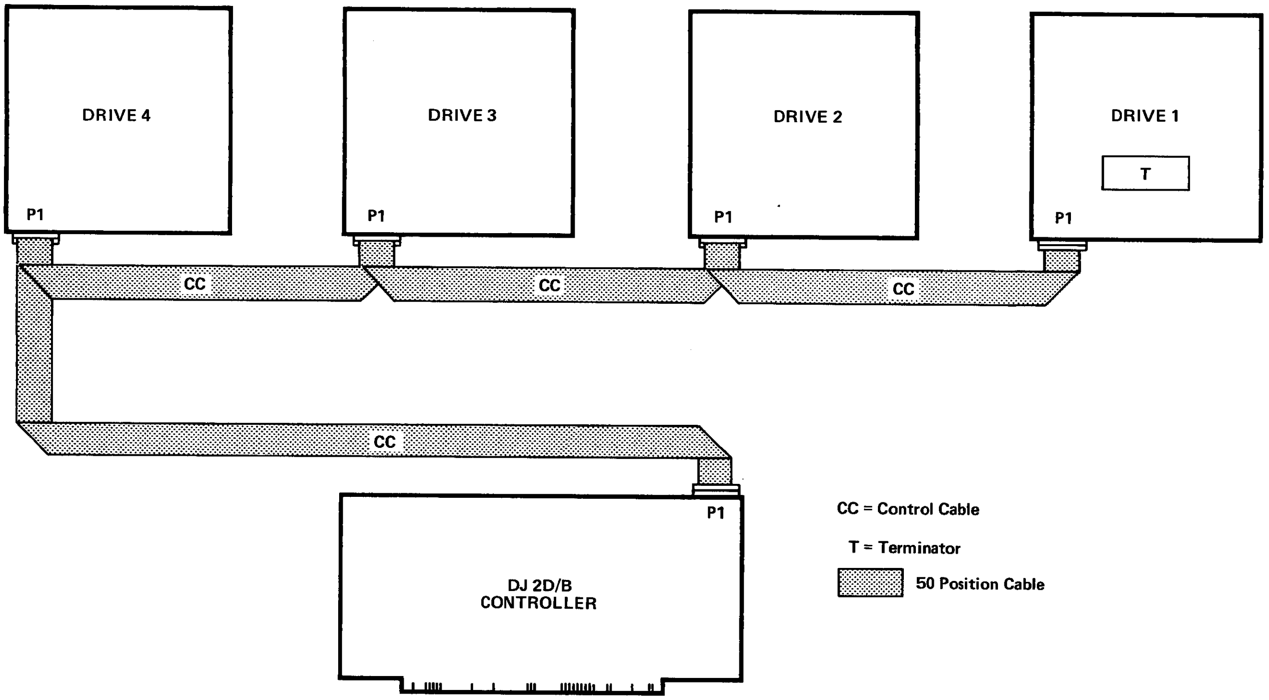


Figure 3-8: Drive Connection

## 3.10. INSTALLATION CHECKS

Once the DJ2D/B controller board is installed, perform the following check-out procedure:

1. Turn the system power on.

2. 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.

3. Check components for excessive heat or smoke on power up. If there are no signs of faulty operation, proceed to Section 4, SYSTEM START-UP.

   If you see or smell smoke, immediately turn system power OFF. Refer to the Warranty Return Procedures at the beginning of this manual for instructions on returning the board to Morrow Designs.

# 4. SYSTEM START-UP AND POWER DOWN

## 4.1. STANDARD ENVIRONMENT- OVERVIEW

In a standard environment, an assembled DJ2D/B is part of a Morrow Designs DISCUS 2 system. A copy of CP/M is included with this system and is tailored to the DJ2D/B's requirements. This version of CP/M expects that a serial RS-232 terminal is connected to the serial port (P2) on the DJ2D/B.

CP/M is supplied on a "write protected" diskette (the notch is OPEN). This disk MUST remain write protected--DO NOT COVER THE NOTCH ON THE DISKETTE. The system is designed to self load the CP/M operating system when the disk is placed in Drive A and a jump is made to F800H, through either a CPU jump or a DJ2D/B power on jump.

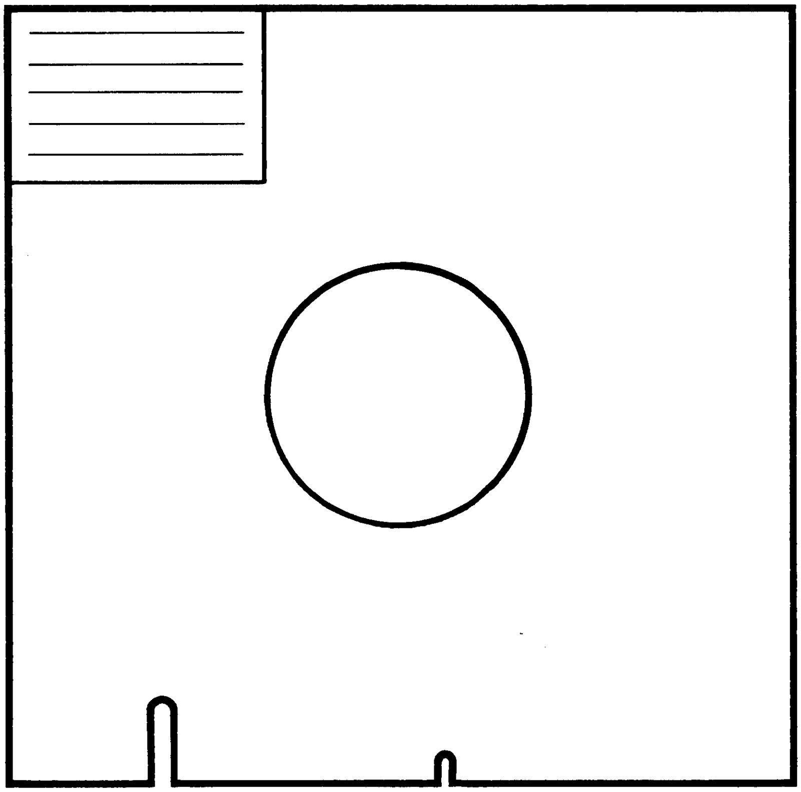


Figure 4-1: Write Protected Floppy Disk

## 4.2. BOOTSTRAP LOAD

If a diskette has NOT 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. The boot LED will turn on without flashing.

To execute a bootstrap load in this mode:

1. Insert a system diskette into Drive A. Do NOT lower the door latch yet.

2. 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).

3. Wait for the activity light at the front of the drive to flash on and off.

4. When it goes off, close the drive door by pushing down on the door latch until a click is heard. The system will boot the next time the drive activity light goes on.

5. Watch for a message describing your system configuration, followed by the CP/M prompt.

At this point we strongly recommend that you format some blank diskettes and make a backup copy of your system diskette. Instructions for formatting diskettes are given in the next section; see the CP/M manual for instructions on backing-up diskettes.

If you were not successful in booting up your system (no prompt appears), refer to the Troubleshooting Table in Appendix D.

## 4.3. DISKETTE INITIALIZATION AND FORMATTING

Diskettes, as purchased, must be initialized (formatted) before copying or writing data on them. Software included with the DJ2D/B contains a format program (FORMT#.COM) which allows the user to format diskettes in various formats. The 128 byte single density format option allows the user maximum compatability with other systems with minimum storage. It should be used only when diskettes are to be transferred to other non-Morrow Designs disk systems. When maximum storage capacity is desired or when diskettes are to be used in DJ2D/B systems only, the 1024 byte/sector format should be chosen.

NOTE: Due to the size of the Morrow Designs CP/M implementation, diskettes which are to have the operating system resident (able to be cold booted from) on them MUST be in 1024 byte/sector format. The 128 byte, single density disks are data diskettes only and cannot have the operating system resident. The Morrow Designs SYSGEN.COM program will put the operating system onto a 1024 byte formatted diskette but will report errors if the diskette is formatted for other than 1024 bytes/sector.

The CP/M operating systems reserves the first two tracks of the diskette for the operating system (track 0 and 1). In addition, track 0 sector 1 (the very first sector on the diskette), contains a CP/M cold boot loader. This is the program which, once loaded into memory by the DJ2D/B controller on cold boot, commands the controller to load in the CP/M operating system from the system tracks into memory and then enter the operating system. Track 2 is reserved for the CP/M directory and contains imformation about the files on the diskette (e.g. how big the file is and where it is stored on the diskette). It is like a road map of the diskette.

This program is menu driven, making it simple to use. To enter the program, type:

```
FORMT#
```

and press the RETURN key. The program will step the heads, formatting each track sequentially, and print an * for each track completed.

## 4.4. SYSTEM SHUT DOWN

Shut down of the system involves:

1. Removing all diskettes from the disk drives, and

2. Shutting off power to all system devices.

It is strongly recommended that you make a back-up copy of the disks previously in use, prior to shutting down the system.

# 5. OPERATION IN A NON-STANDARD ENVIRONMENT

## 5.1. OVERVIEW

This section offers a more detailed description of the options available with your DJ2D/B. It is directed toward those users desiring to configure their board for a non-Morrow Designs operating environment. As such, the information given here is more technically oriented than that in previous sections.

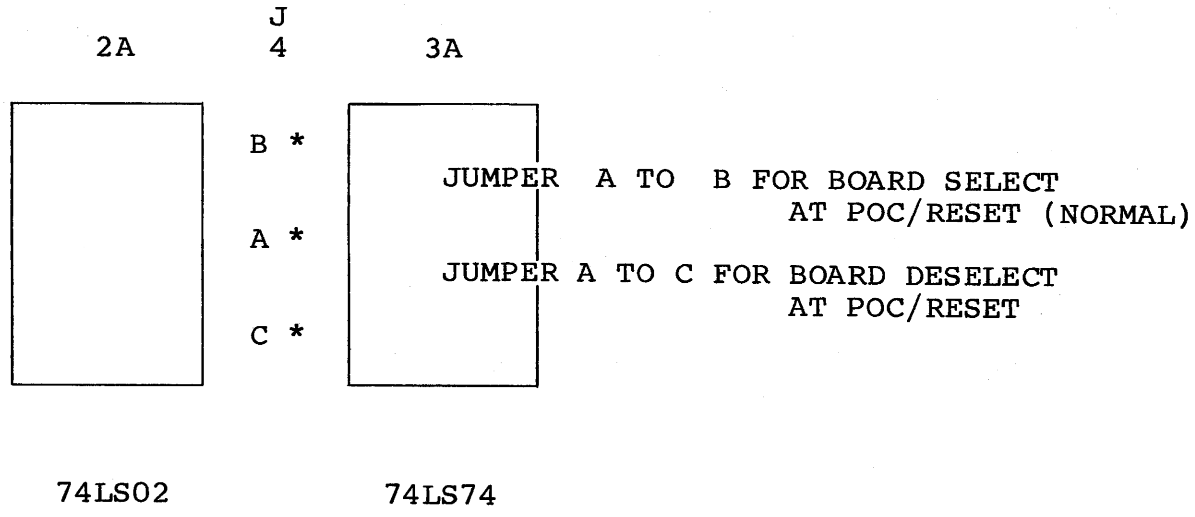
## 5.2. BANK SELECTION

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 41H. Port 41H 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 512K bytes of memory on the S-100 bus without addressing conflicts.

Since system software and user programs are continually requiring greater memory, memory mapped devices such as the Disk Jockey must exercise care in the use of S-100 bus memory space. To allow for the increased memory requirements, the Disk Jockey has implemented the Bank Select Port, Port 41H. This port allows the 2K of memory space used by the board to be assigned to any of eight banks within the bank 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, decide which bit within Port 41H 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. Most bank select schemes will require the DJ2D/B to be active on power-on. This decision is made by the installation of another jumper. The details of these two jumper options are presented in Figure 3-4 and 5-1.



JUMPER A TO B FOR BOARD SELECT
AT POC/RESET (NORMAL)

JUMPER A TO C FOR BOARD DESELECT
AT POC/RESET

**Figure 5-1: Bank Select Jumper Options**

Factory assembled boards are 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, remove the jumper and install it between A and C. NOTE: ONE of the two jumpers MUST 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.

**NOTE: BOTH jumpers should NEVER be installed simultaneously.**

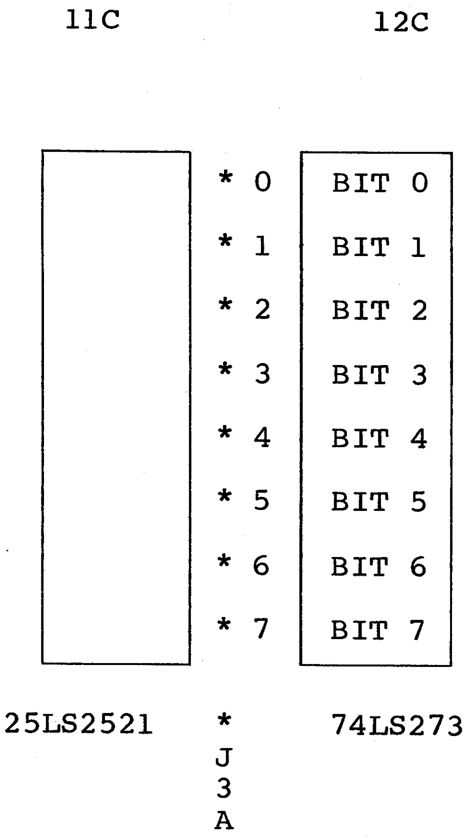


Figure 5-2: Bank Select Bits

The bank select scheme will provide for eight banks of memory each having 64K bytes. These banks are numbered 0 through 7 and correspond to the bit positions in Figure 4-6 at the the right. The pad just above J3A should be jumpered to one of the pads to the right. The bit number to the right of the pad will determine the memory bank in which the Disk Jockey will reside. Once this choice is made, the Disk Jockey will be enabled or disabled when the CPU executes an OUT 41H insruction. 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 41H 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 41H 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.

## 5.3. INTERRUPT LOGIC

Whenever the 1791 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.

The line will go low when an interrupt is generated.

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 is recognized by software vendors. It is also fair to say that interrupt driven operating systems are somewhat more complex. Their implementation requires a great deal more thought than implementation of non-interrupt driven operating systems. 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 area of the circuit board. The jumper should originate at the open pad, just to the left of J1A. Connect the jumper 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 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. Figure 5-3 below shows the jumper pad layout for installing interrupts on the DJ2D/B board.

```
             * 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

*   *   *   *   *   *   *   *   *
```

**Figure 5-3: Jumper Pad Layout**

5.4. POWER-ON JUMP LOGIC

The Disk Jockey controller has the ability to generate a power on jump address on the system S-100 bus when power is first applied or when a system reset is active. This address generating ability forces the CPU to jump to the DBOOT routine in the EPROM on the DJ2D/B board and causes the system to boot an operating system from a diskette into memory.

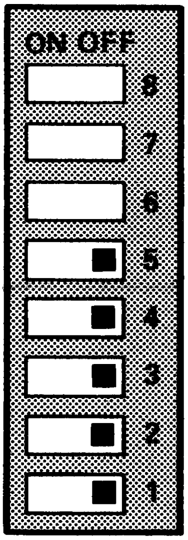
Six paddles on Switch 1 control the Power-On Jump logic of the controller. Paddle 8, at the top of the switch, enables or disables the Power-On Jump circuitry. The logic is enabled if Paddle 8 is in the "on" position; Power-On Jump logic is disabled if Paddle 8 is in the "off" position.

If the logic is DISABLED, the settings of Paddles 1 through 5 are not important. If the logic is ENABLED, settings of Paddles 1 through 5 inform the CPU of the DJ2D/B starting address within a 64K region of memory.

The controller uses 2K of address space which starts on a 2K boundary. It is necessary, then, to specify the 5 high order address bits to affect a branch to the controller. Paddles 1 through 5 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 DJ2D/B board located at F800H, paddles 1, 2, 3, 4 and 5 should be "OFF".



Enable/Disable POJ

Power-On Jump Switch Settings

(set for jump to F800H)

Figure 5-4: Power-On Jump Settings

Table 5-1 illustrates the factory settings for Switch 1 Power-On Jump logic. Appendix A details each of the available Switch 1 Power-On Jump paddle settings.

| Address | SW1-1 | SW1-2 | SW1-3 | SW1-4 | SW1-5 |
|---|---|---|---|---|---|
| E000 | off | off | off | on | on |
| E800 | off | off | off | on | off |
| F000 | off | off | off | off | on |
| F800 | off | off | off | off | off |

Table 5-1: Power-On Jump Table

## 5.5. PHANTOM ENABLE

The DJ2D/B 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 DJ2D/B 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. In most other systems, this switch should remain off.

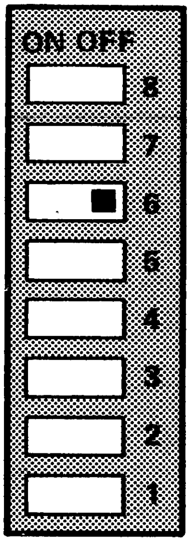


Figure 5-5: Phantom Enable

At no time should the DJ2D/B generate and respond to PHANTOM simultaneously. If this happens, the board will generate the PHANTOM signal, then disable itself.

The DJ2D/B can also generate a PHANTOM signal that allows the controller to coexist with 64K memory boards. Normally, this option is installed at the factory.

The memory board must support the PHANTOM disable signal. If it does not, it must be addressed so it will not conflict with the DJ2D/B's memory space (locations F800-FFFFH). Generally, the DJ2D/B will not allow you to boot the system if these addresses do conflict.

## 5.6. 2, 4 and 6 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 6 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 instruction fetches from this part. The DJ2D/B 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. In systems like the Morrow Designs Decision 1, this switch should be in the "off" position. The Morrow Designs MPZ80 CPU inserts one wait state on each instruction fetch automatically, and therefore requires no wait states be generated by the DJ2D/B. When operating with other CPU boards which cannot generate wait states and are operating at speeds above 2 MHz, this switch should be "on".

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.

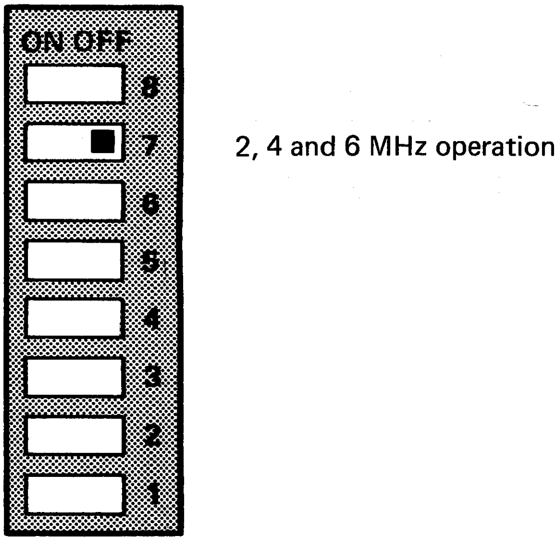


Figure 5-6: Bus Speed Setting

## 5.7. SERIAL I/O SWITCH SETTINGS

### 5.7.1. Baud Rate Selection

Paddles 1 to 4 of Switch 2 control the baud rate (number of characters per second transmitted or received by the serial port) for the serial port (1602 UART). Sixteen separate baud rates, ranging from 50 to 19,200, are available. The following table lists the most common switch settings for baud rate selection.

| SW2-1 | SW2-2 | SW2-3 | SW2-4 | BAUD RATE |
|---|---|---|---|---|
| on | on | off | on | 110 |
| on | off | off | off | 1200 |
| off | off | off | on | 9600 |
| off | off | off | off | 19200 |

Table 5-2: Baud Rate Switch Settings

NOTE: The baud rate set on this switch must match the baud rate of the terminal that will be used in system operation. Most CRT terminals will function at 9600 baud, and all boards are shipped from the factory at this baud rate.

### 5.7.2. 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 DJ2D/B. Normally, this switch is set to the "off" positions as most terminals support 8 bit data words.

NOTE: Morrow Designs software strips the eighth data bit (ignores it) if 8 bit support is required (this is not required for normal operation).

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

Table 5-3: Word Length Selection

### 5.7.3. Stop Bit Count

SW2-6 controls the number of stop bits (either one or two), sent from the UART after each data word. The "off" position sets the device to two stop bits; the "on" position sets the device to one stop bit.

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. Normally, this switch is "off".

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

Table 5-4: Stop Bit Count Selection

### 5.7.4. Parity

Normally, Paddle 8 of Switch 2 is in the "off" position, and the UART will not generate any parity (check) bits at the end of the serial data word. However, if the paddle is in the "on" position, refer to the table below for the proper parity setting via Paddle 7.

NOTE: Morrow Designs software/firmware does not check for parity errors during transmission; this switch will be "off", unless the user supplies his/her own serial port drivers.

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

Table 5-5: Parity Switch Setting

## FAST REFERENCE GUIDE FOR DJ2D/B DIP SWITCHES

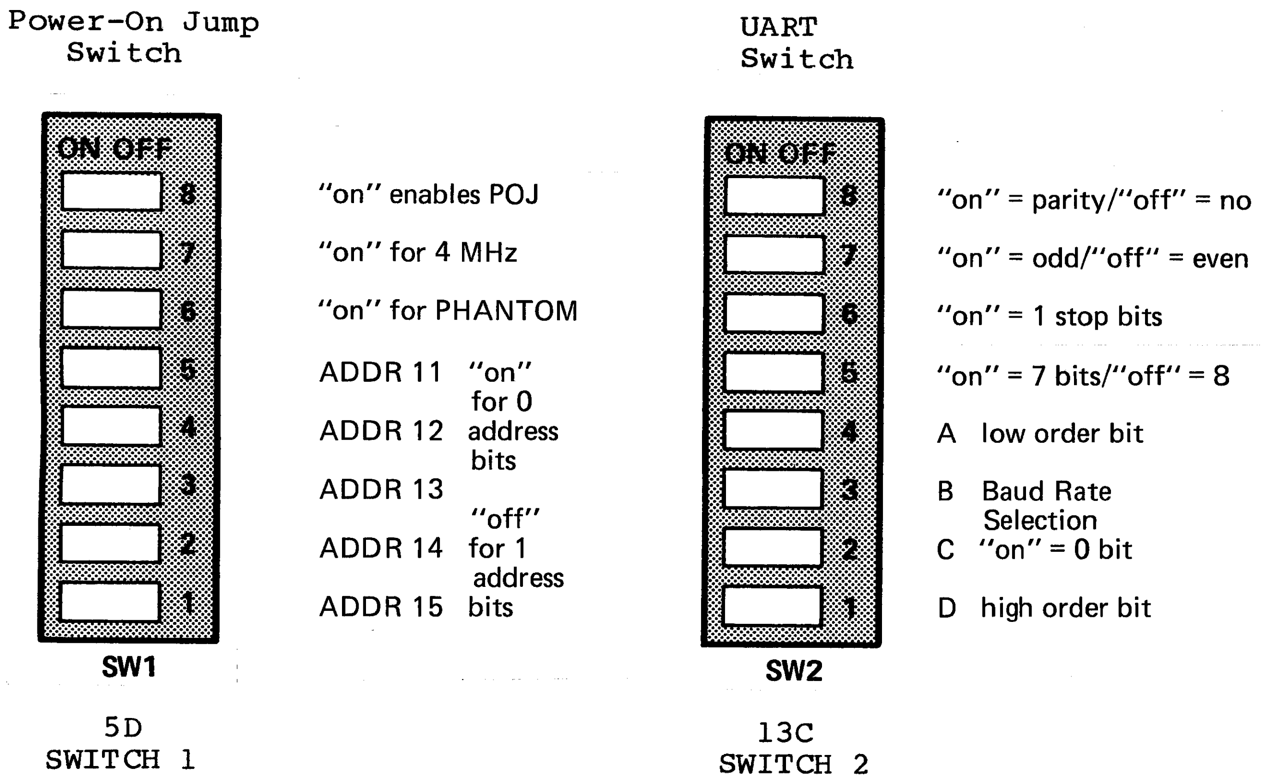


Figure 5-7: Quick Reference for DJ2D/B DIP Switches

## 5.8. DYNAMIC MEMORY CONSTRAINTS

The DJ2D/B uses the S-100 signal pRDY to synchronize the CPU to the disk drive. During data transfers to and from the disk, the pRDY line may be held low for as long as 2 milliseconds. Dynamic memory boards which perform refresh using the Z80 refresh capabilities or S-100 control or status signals, will not be refreshed for up to 2 millisecond intervals; therefore, these boards must have provisions for internal refresh.

# 6. HARDWARE REGISTERS

## 6.1. DATA ORGANIZATION

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. Figure 6-1 illustrates the concept of data organization.

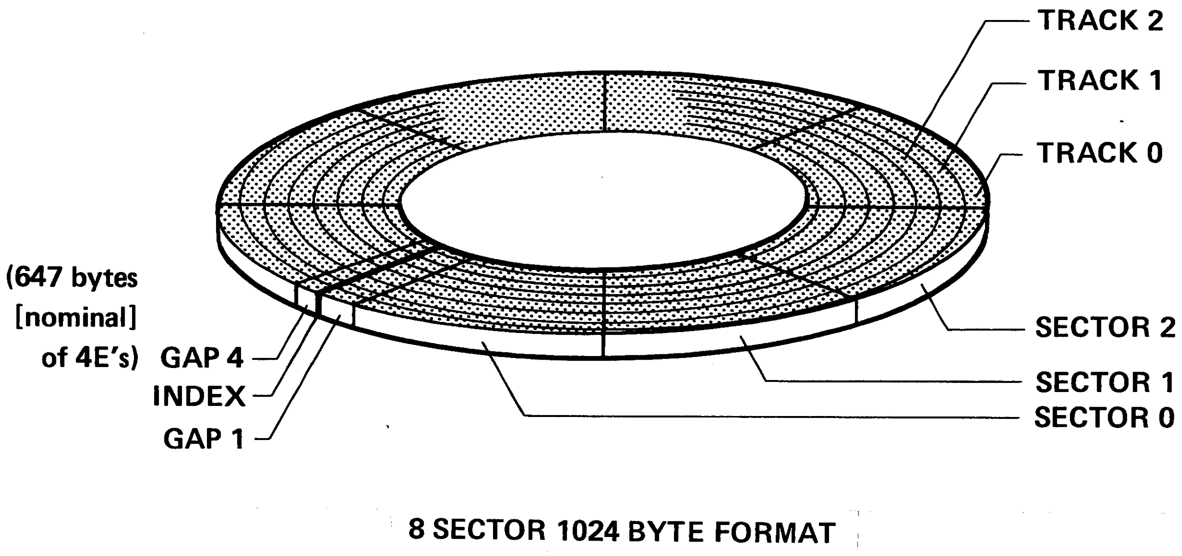

TRACK 2
TRACK 1
TRACK 0
(647 bytes [nominal] of 4E's) GAP 4
INDEX
GAP 1
SECTOR 2
SECTOR 1
SECTOR 0
8 SECTOR 1024 BYTE FORMAT

Figure 6-1: Floppy Disk Data Organization

Information on the diskette is organized into 77 concentric cylinders called tracks. The drive read/write head(s) 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(s) one track toward the center of the diskette. A step out command moves the head one track away from the center of the diskette. The numbering of the tracks is arranged so that track zero is the farthest from the center of the diskette. The DJ2D/B uses the Western Digital 1791 controller. This controller senses the current track number over which the read/write head is located and calculates the number of "Step in" or "Step out" commands required to move the head to a new track.

## 6.2. SECTOR FORMATS

Once the read/write head has been moved to the desired track, rotation of the diskette moves a circle of magnetic material beneath the head. Within this circle of material, data is recorded in distinct regions called sectors. A sector is the smallest amount of information that can be separately read from or written to the diskette. Presently, IBM supports three different sector formats. Table 6-1 details the relationship between the size of a sector and the number of sectors per single track.

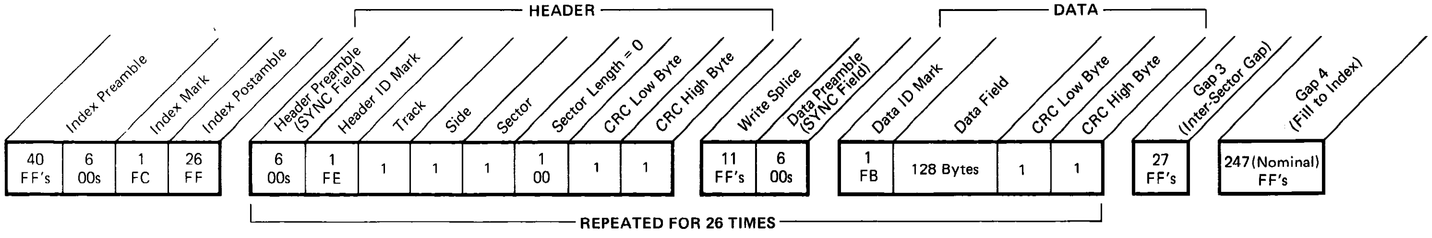
NOTE: Morrow software supports only 128 byte format in the single density mode.

| | bytes of data per sector | sectors per track |
|---|---|---|
| SINGLE DENSITY | 128 | 26 |
| DOUBLE DENSITY | 256 | 26 |
| | 512 | 15 |
| | 1024 | 8 |

Table 6-1: Sector Formats

Within a sector, there is a header field and a data field. The header field preceeds the data field, and uniquely defines that field by:

- track number
- side (for double sided drives)
- sector number
- sector length



| Index Preamble | Index Mark | Index Postamble | Header Preamble (SYNC Field) | Header ID Mark | Track | Side | Sector | Sector Length = 0 | CRC Low Byte | CRC High Byte | Write Splice | Data Preamble (SYNC Field) | Data ID Mark | Data Field | CRC Low Byte | CRC High Byte | Gap 3 (Inter-Sector Gap) | Gap 4 (Fill to Index) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40<br>FF's | 6<br>00s | 1<br>FC | 26<br>FF | 6<br>00s | 1<br>FE | 1 | 1 | 1 | 1<br>00 | 1 | 1 | 11<br>FF's | 6<br>00s | 1<br>FB | 128 Bytes | 1 | 1 | 27<br>FF's | 247(Nominal)<br>FF's |

Figure 6-2: 128 Bytes/Sector Format

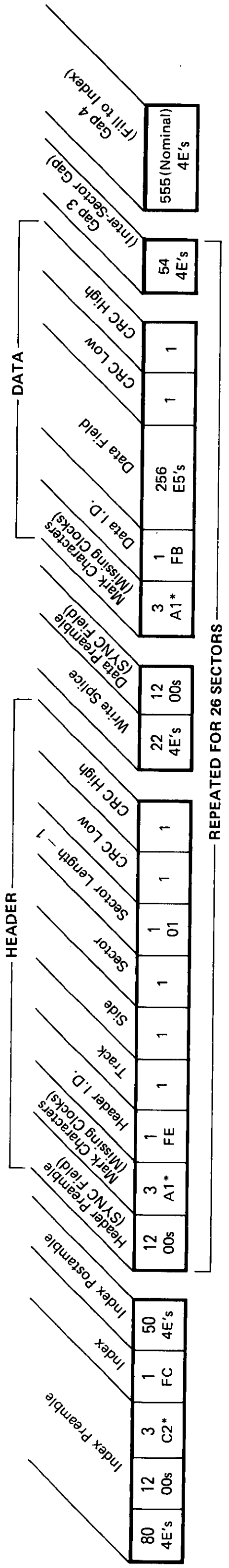


Figure 6-3: 256 Bytes/Sector Format

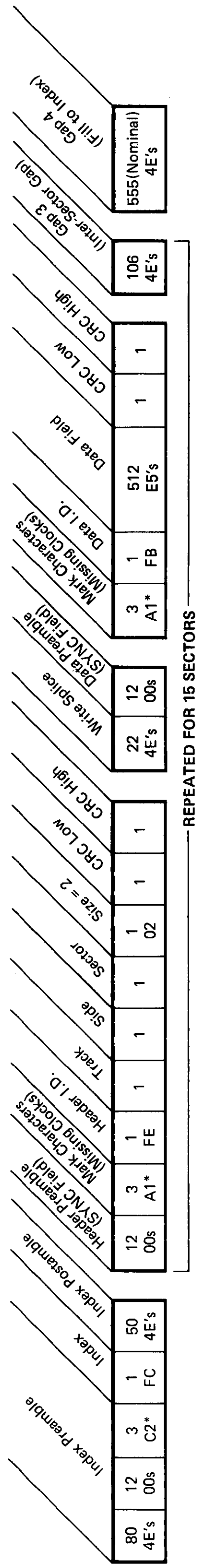


Figure 6-4: 512 Bytes/Sector Format

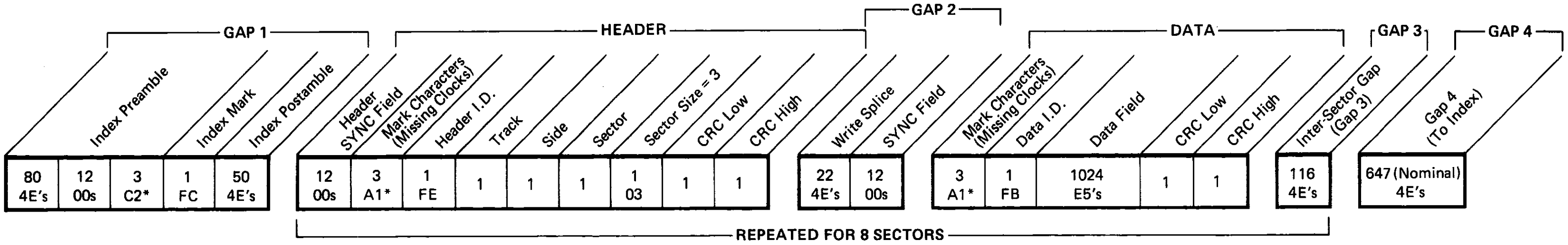


Figure 6-5: 1024 Bytes/Sector Format

## 6.3. I/O REGISTERS

To exercise a greater level of control over the floppy diskette or serial interface, the user may refer directly to the DJ2D/B device registers from your 8080 or Z80 program. There are 14 eight bit registers; each associated with one of eight memory addresses from FBF8-FBFF on the S-100 bus. Table 6-2 below presents the registers located on the controller and UART.

| DEVICE | I/O REGISTERS | | |
|---|---|---|---|
| | READ Only | WRITE Only | READ/WRITE |
| 1791 Controller | Status | Command | Track<br>Sector<br>Data |
| 1602 UART | Data Input<br>Status | Data Output | -- |
| DJ2D/B Firmware | DJ Status Register | Drive Control Register<br>Function | -- |
| **Total # on DJ2D/B | 5 | 6 | 3 |

Table 6-2: I/O Registers

Note: When the addressed register is READ only or WRITE only, a different register is selected during a read operation and a write operation.

The 1791 controller has a negative logic data bus. For this reason the internal bidirectional data bus of the DJ2D/B 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.

Refer to the 1791 controller and 1602 UART data sheets for a detailed description of register functions.

The DJ2D/B Memory Map, Table 6-3, presents an overview of the hardware registers and their location. Each register is discussed separately in the sections following.

| HEX ADDRESS | FUNCTION | |
|---|---|---|
| F800-FBF7 | ROM FIRMWARE | |
| | I/O REGISTERS | |
| | WHEN READ | WHEN WRITTEN |
| FBF8 | UART INVERTED DATA INPUT | UART INVERTED DATA OUTPUT |
| FBF9 | UART INVERTED STATUS | DISK JOCKEY FUNCTION |
| FBFA | DISK JOCKEY STATUS | DRIVE CONTROL REGISTER |
| FBFB | NOT USED | NOT USED |
| FBFC | 1791 CONTROLLER STATUS | 1791 CONTROLLER COMMAND |
| FBFD | 1791 TRACK REGISTER | |
| FBFE | 1791 SECTOR REGISTER | |
| FBFF | 1791 DATA REGISTER | |
| FC00-FFFF | RAM | |

Table 6-3: DJ2D/B Memory Map

## 6.4. READ ONLY REGISTERS

### 6.4.1. Register 0

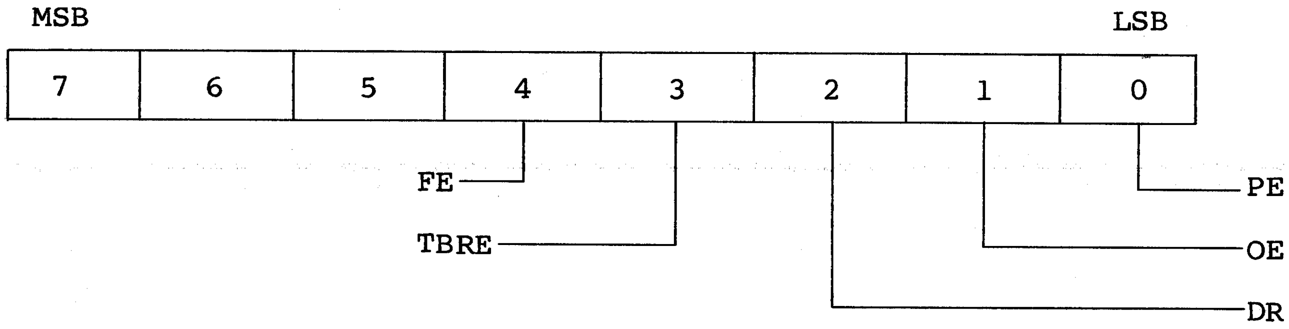
**Register 0 - The inverted UART data output register**
**Location FBF8 standard Disk Jockey**

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.

### 6.4.2. Register 1

**Register 1 - The inverted UART status register**
**Location FBF9 standard Disk Jockey**

Only the low order five bits of this register have any significance. The meaning of these bits is presented in Figure 6-6. Refer to the 1602 data sheet for a more detailed discussion of these bits. The signals are listed, using their positive logic mnemonics with the understanding that the actual signals read will be the negation of these mnemonics.



FE = Framing Error
TBRE = Transmitter Buffer Register Empty
DR = Data Ready
OE = Overrun Error
PE = Parity Error

**Figure 6-6: Inverted UART Status Bits**

### 6.4.3. Register 2

**Register 2** - Disk Jockey status register
Location FBFA standard Disk Jockey

This register contains bits that identify the current status of the DJ2D/B and the currently selected drive. Only the six low order bits have any significance in this register. Figure 6-7 presents the meanings of these bits. For a detailed specification of these signals, see the documentation that accompanies the floppy disk drive. If no drive is currently selected or if the head is not loaded, these bits are all high.

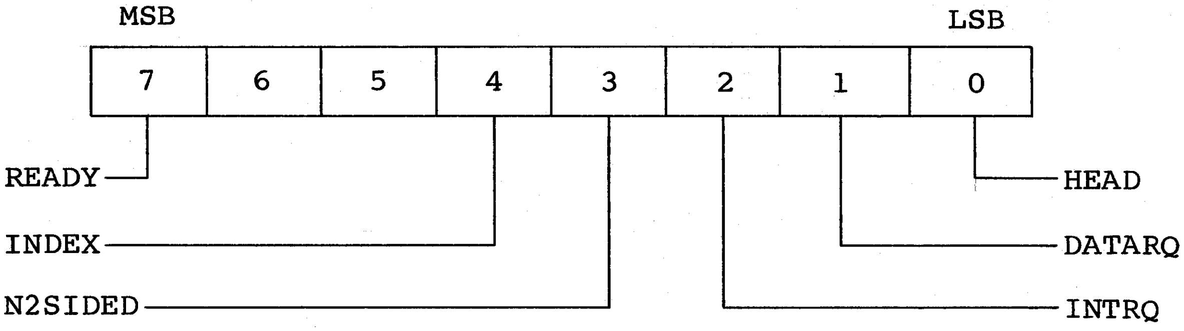


**Figure 6-7: Disk Jockey Status Register**

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 low once per diskette revolution.

N2SIDED- This line is a 0 when a double sided drive is connected to the controller AND there is a double sided diskette 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 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.

INTRQ - The 1791 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 controller.

### 6.4.4. Register 3

Register 3 - Not currently used
Location FBFB standard Disk Jockey

### 6.4.5. Register 4

Register 4 - 1791 controller status register
Location FBFC standard Disk Jockey

This is the status register of the 1791 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 data document for a detailed discussion of this register.

## 6.5. WRITE ONLY REGISTERS

### 6.5.1. Register 0

Register 0 - The inverted UART data input register
Location FBF8 standard Disk Jockey

Inverted data is stored is 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.

### 6.5.2. Register 1

Register 1 - Disk Jockey drive control register
Location FBF9 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's data register, and
- set or clear the master reset pin of the 1791 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 in Figure 6-8 below:

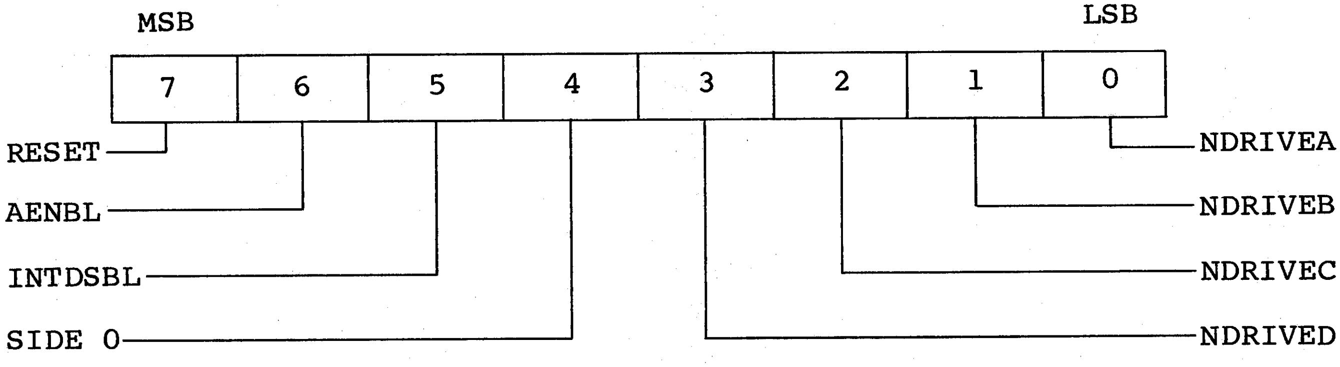


Figure 6-8: Drive Control Register

RESET - When a one is stored in this bit, the master reset pin of the 1791 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. This bit is used to reinitialize the 1791 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. The 1791 will execute a home command and place itself in a state to accept commands.

AENBL - When the CPU references the 1791 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 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 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. This enables the CPU to synchronize its data transfers to the 1791 controller.

INTDSBL - When this bit is a zero, the Interrupt Request line of the 1791 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. Consult the 1791 data sheet for a thorough understanding of the chip's Interrupt Request line.

The DJ2D/B controller can control up to 4 eight inch drives. These drives "share" the drive signal lines going to the controller. Only one disk drive is allowed to drive the signal lines back to the controller but all drives "listen" to the control signal lines coming from the controller. This is accomplished by use of the drive select lines. When multiple drives are used, they must be jumpered to determine which drive they will respond to (drive 1- drive 4). When a command is sent to a drive, only the drive that matches the select lines from the controller will respond. All other drives will ignore the command. The boot drive (CP/M 'A' drive) with all Morrow softare is drive 1.

The only directly programmable control over the disk drive from software is the drive select register (at memory address FBF9h). The low 5 bits of this register select a drive and, on double sided drives, a head on the selected drive. The bits are mapped as follows:

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Drive selected |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | Drive 1 ('A') |
| 1 | 1 | 0 | 1 | Drive 2 ('B') |
| 1 | 0 | 1 | 1 | Drive 3 ('C') |
| 0 | 1 | 1 | 1 | Drive 4 ('D') |

NOTE: Only one of the lower four bits can be low at one time or more than one drive will be selected causing data errors when reading.

In addition to the bits above this port also controls which side of a double sided drive is selected. Bit 4 of this register when high selects side 0 and when low selects side 1 drives .

Control of all the other drive signal lines, except head load (see 6.5.3, Register 2), is handled by the 1791 Floppy Disk Controller chip and are not directly accessible to the programmer through registers.

### 6.5.3. Register 2

Register 2 - The Disk Jockey function register
Location FBFA 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 at which the 1791 controller operates, 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 as if ones had been written in all four bits. The specific function of the various bits in this register is detailed in Figure 6-9.

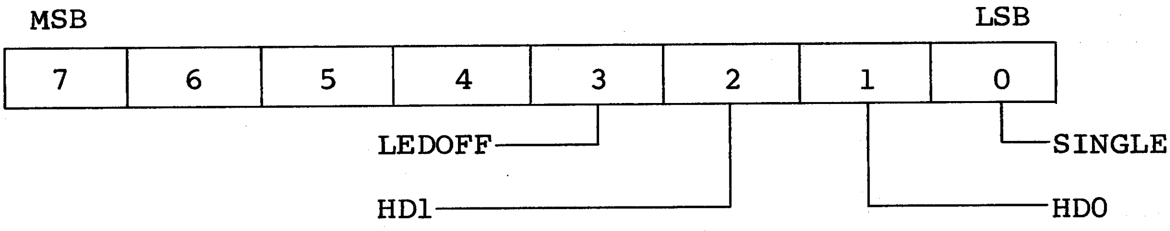


Figure 6-9: Disk Jockey Function Register

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 DJ2D/B board will read and write data to and from the disk in single density. When this bit is a zero, read and write operations 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.

| HD1 | HD0 | Read/Write Head Function |
|---|---|---|
| 0 | 0 | Head is loaded |
| 0 | 1 | Not allowed |
| 1 | 0 | 1791 may unload head |
| 1 | 1 | Head is unloaded |

**Figure 6-10: Read/Write Head Functions**

### 6.5.4. Register 3

**Register 3** - Not currently used
Location FBFB standard Disk Jockey

## 6.6. 1791 CONTROLLER REGISTERS

The DJ2D/B disk controller makes use of the Western Digital 1791 LSI disk controller chip (optionally the Fujitsu 8866 or 8876). This reduces the parts and software overhead dramatically from first generation controllers. This controller chip will read and write single and double density diskettes.

The 1791 chip has 4 registers which reside in the DJ2D/B memory map. The 1791 has inverted data lines which is why the controller inverts the data bus internally from the S-100 bus. This means all 1791 functions go through a double inversion or are the same in the 1791 registers as the CPU registers. This was done to ease programming requirements for software for the DJ2D/B board. The registers are as follows:

| Memory Location | Function |
|---|---|
| FBFAh | 1791 Status Register (read) |
| FBFAh | 1791 Command register (write) |
| FBFDh | Track register (read/write) |
| FBFEh | Sector Register (read/write) |
| FBFFh | Data register (read/write) |

Table 6-4: 1791 Controller Registers

### 6.6.1. Register 4 (WRITE)

**Register 4 (WRITE) - 1791 controller command register**
Location E3FC standard Disk Jockey

This is the command register of the 1791 controller. There are four different classes of commands. Within each class, there is a number of separate commands the controller can execute. See the tables on the following page and the 1791 data document for a detailed discussion of this register and its use.

| | | BITS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| TYPE | COMMAND | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I | Restore | 0 | 0 | 0 | 0 | h | V | r1 | r0 |
| I | Seek | 0 | 0 | 0 | 1 | h | V | r1 | r0 |
| I | Step | 0 | 0 | 1 | u | h | V | r1 | r0 |
| I | Step In | 0 | 1 | 0 | u | h | V | r1 | r0 |
| I | Step Out | 0 | 1 | 1 | u | h | V | r1 | r0 |
| II | Read Command | 1 | 0 | 0 | m | X | E | 0 | 0 |
| II | Write Command | 1 | 0 | 1 | m | X | E | X | a0 |
| II | Read Address | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| III | Read Track | 1 | 1 | 1 | 0 | 0 | 1 | 0 | X |
| III | Write Track | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| IV | Force Interrupt | 1 | 1 | 0 | 1 | I3 | I2 | I1 | I0 |

X = Don't care

Note: Bits shown in TRUE form.

Table 6-5: Command Summary

TYPE I

h = Head Load Flag (Bit 3)

h = 1, Load head at beginning

h = 0, Unload head at beginning

V = Verify flag (Bit 2)

V = 1, Verify on last track

V = 0, No verify

r1r0 = Stepping motor rate (Bits 1-0)

| r1 | r0 |  |
|---|---|---|
| 0 | 0 | 3 ms |
| 0 | 1 | 6 ms |
| 1 | 0 | 10 ms |
| 1 | 1 | 15 ms |

u = Update flag (Bit 4)

u = 1, Update Track register

u = 0, No update

Table 6-6: Flag Summary

TYPE II

m = Multiple Record flag (Bit 4)

m = 0, Single Record<br>
m = 1, Multiple Records

a0 = Data Address Mark (Bit 0)

a0 = 0, FB (Data Mark)<br>
a0 = 1, F8 (Deleted Data Mark)

E = 15 ms Delay

E = 1, 15 ms delay<br>
E = 0, no 15 ms delay

Table 6-7: Flag Summary

TYPE IV

Ii = Interrupt Condition flags (Bits 3-0)

I0 = 1, Not-Ready to Ready Transition

I1 = 1, Ready to Not-Ready Transition

I2 = 1, Index Pulse

I3 = 1, Immediate Interrupt

Table 6-8: Flag Summary

### 6.6.2. Register 4 (READ)

Register 4 (READ) - 1791 Controller status register
Location FBFC standard Disk Jockey

NOTE: The discussion that follows refers to the state of bits as their actual state, as read from the 1791 controller. Bit status is as the bits appear to the host CPU.

The status register contains information regarding the completion of the last command issued to the controller chip. This information change is dependent upon the command executed. Upon receipt of a command, the busy status bit (data bit 0) is set high, and all other status bits are cleared to 0.

Data bit 1 of this register is the Index signal from the selected disk drive, whenever a seek command is issued. Whenever a command which requires reading or writing to the diskette, this status bit becomes the DRQ (DATA REQUEST) line. When this bit is high during a read operation, it means the Data register has a byte of data to be transferred to the CPU. This data must be read by the CPU before the next byte comes from the diskette into the data register. When this bit is high during a write operation, the next character destined for the diskette must be written into the data register. This status bit is also an actual signal output of the 1791 controller. The DJ2D/B uses this output rather than sampling the status register (more on this later).

Data bit 2 when high during a seek operation is the Track 0 signal from the selected disk drive. When high, this indicates the disk drive heads have reached track 0. During a read or write operation this bit becomes Lost Data. When high this indicates that the CPU did not respond correctly to the DRQ line. This means that the data register overflowed during a read operation.

Data bit 3 is the CRC (Cyclic Redundancy Check) error line which, when high, indicates the last command was not sucessful because the CRC pattern at the end of a header or data field did not match what was written onto the diskette.

Data bit 4 when high, indicates a seek error occurred during a disk drive head seek operation (heads are not on the correct track on completion of the command). When high during a disk read/write operation indicates that the track and sector number which was to be read or written could not be matched by the controller chip.

Data bit 5 when high, indicates the heads are loaded subsequent to a seek type operation (settled, on track and in contact with the media). During a write operation this bit indicates a write fault when high (drive became not ready during a write operation).

Data bit 6 when high, indicates the diskette in the currently selected drive has been write protected (write protect notch on diskette is not covered).

Data bit 7, when high, indicates the currently selected drive is not ready (diskette is not inserted, power has not been applied or the drive is not jumpered/connected correctly).

### 6.6.3. Register 5

**Register 5** - 1791 track register
Location FBFD standard Disk Jockey

The 1791 controller uses this register as a reference to determine the position of the disk drive's read/write head. Exercise extreme care when writing in this register. If care is not exercised, seek errors may likely occur. See the 1791 data document for a more detailed discussion.

### 6.6.4. Register 6

**Register 6** - 1791 sector register
Location FBFE standard Disk Jockey

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

### 6.6.5. Register 7

**Register 7** - 1791 data register
Location FBFF standard Disk Jockey

This is the data register of the 1791 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. Refer to the 1791 data document for a more complete discussion.

## FINAL NOTE

The Disk Jockey firmware contains numerous examples illustrating the use of the hardware registers listed in this sections. 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.

# 7. SOFTWARE FUNCTIONS

## 7.1. ROM JUMP TABLE

There are 17 standard I/O subroutines supplied in PROM on the DJ2D/B. To use these subroutines, the first few words of the system ROM must branch to the appropriate address in the jump table. 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 the 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 =F800. Refer to Table 7-1 for the address(es) to call for the utility routines.

| ADDRESS | STANDARD VALUE | SYMBOLIC VALUE | FUNCTION |
|---|---|---|---|
| A | F800 | DBOOT | DOS bootstrap routine |
| A+3 | F803 | TERMIN | Serial input |
| A+6 | F806 | TRMOUT | Serial output |
| A+9 | F809 | TKZERO | Recalibrate (seek to TRK0) |
| A+12 | F80C | TRKSET | Seek |
| A+15 | F80F | SETSEC | Select sector |
| A+18 | F812 | SETDMA | Set DMA address |
| A+21 | F815 | DREAD | Read a sector of disk data |
| A+24 | E018 | DWRITE | Write a sector of disk data |
| A+27 | F81B | SELDRV | Select a disk drive |
| A+30 | F81E | TPANIC | Test for panic character |
| A+33 | F821 | TSTAT | Serial status input |
| A+36 | F824 | DMAST | Read current DMA address |
| A+39 | F827 | STATUS | Disk status input |
| A+42 | F82A | DSKERR | Loop to strobe error LED |
| A+45 | F82D | SETDEN | Set density |
| A+48 | F830 | SETSID | Set side for 2-headed drives |

Table 7-1: ROM Jump Table

Each 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.

## 7.2. SERIAL PORT (P2)

The DJ2D/B contains a hadware Universal Asynchronous Receiver Transmitter (UART) chip. This chip replaces several MSI and SSI circuits in one versatile 40 pin package. This UART (location 14D) provides an RS-232 communications interface for easy connection to a standard video display terminal.

The serial port can communicate with devices at speeds from 50 Baud to 19.2 Kbaud (selectable by switches at board location 13C). In addition to baud rate selection, the user can select the number of data bits, stop bits and parity via these switches. The vast majority of users will connect their terminal to this port for the CP/M console. Most popular terminals can be quickly interfaced to the DJ2D/B by setting the switches as follows:

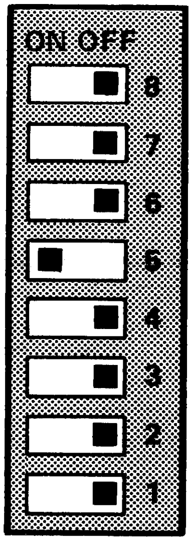


Figure 7-1: Serial Port (P2) Switch Setting

Software Functions

There is one read/write register and one read only register associated with the serial port. All data is transferred to and from the UART via the register at memory location FBF8h. Data which is to be sent to the terminal should be complemented (due to the inverted data bus on the DJ2D/B) and then written to memory location FBF8h. Likewise, data to be received from the serial port is obtained by reading memory location FBF8h and complementing the data obtained.

The other register associated with the UART is the Status register, located at memory address FBF9h. This register must be examined in order to tell if the UART has just received a character or is ready to accept a character for transmission. There are really only two bits of this register which are applicable for the majority of applications: TBRE (bit 3) and DR (bit 2).

The TRBE bit must be examined every time a character is to be sent to the serial port. This assures that the last character sent has finished before attempting to send the next charcater. Upon reading location FBF9h, the user will expect this bit (bit 3) to be low if the UART is ready to send a character, and high if the UART is busy sending out the last character (see the software example below).

```
          org  100h

status    equ  0FBF9h         ;DJ Uart status register
data      equ  0FBF8h         ;DJ Uart data register

;  call routine with character to print in register c, register a
;  is not preserved upon return

conout:   lda  status         ;read the uart status register
          ani  08             ;check for TBRE high
          jnz  conout         ;loop until TBRE goes low
          mov  a,c            ;get the character to send
          cma                 ;complement for DJ inverted bus
          sta  data           ;write the uart data register
          ret
```

Software Functions

The DR bit is examined each time the software is expecting a character to be received from the serial terminal. Upon reading memory location FBF8h, the user will expect this bit (bit 2) to be low if the UART has recieved a character from the terminal , and high if no character has been received (see the software example below).

```
          org  100h

status    equ  0FBF9h          ;DJ Uart status register
data      equ  0FBF8h          ;DJ Uart data register

;  register a is not preserved and upon return contains character
;  received from terminal

conin:    lda  status          ;read the uart status register
          ani  04              ;check for DR high
          jnz  conin           ;loop until DR goes low
          lda  data            ;get the character
          cma                  ;complement for DJ inverted bus
          ret
```

The serial port connection to the DJ2D/B is via 7 pin Molex type connector at the upper left corner of the board. The connector for the DJ2D/B board is included with the board. Most terminals will require a DB-25P (ITT Canon) type connector at the terminal end. Cable lengths should be kept as short as possible to insure data integrity. The standard serial connection would be as illustrated in Figure 3-7.

In the receiver section there is a shift register which assembles a parallel data word from the input serial stream from a CRT terminal after start and stop bits have been removed. When a complete data word has been assembled in this register, it is loaded into the UART data register that is accessible from the system bus.

## 7.3. ROM SUBROUTINES (Serial I/O)

### 7.3.1. TERMIN

This subroutine is used to collect input characters from the terminal connected to the DJ2D/B serial port. 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 is received from the terminal.

### 7.3.2. TRMOUT

This subroutine is used to transmit characters to the terminal that is connected to the DJ2D/B serial port (P2). 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 data register. This automatically resets the TBRE bit.

### 7.3.3. 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 collects 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.

### 7.3.4. 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.

## 7.4. EPROM SUBROUTINES OVERVIEW (Disk Functions)

Data transfers to and from the disk must be preceeded by calls to certain Disk Jockey routines. Therefore, transferring a sector of disk data between memory and the disk involves the steps listed below; each step corresponds to a subroutine call to the DJ2D/B firmware (with the exception of error checking):

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

2. Check for error conditions.

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

4. Specify the starting memory address of the block of data to be transferred to or from the disk.

5. Check for error conditions.

6. Perform the read or write operation.

7. Check for error conditions.

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 two-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. Initialize the drive.

   If the drive has not been accessed before, the read/write head of the drive is in an unknown position. To initialize the drive, call the subroutine "TKZERO"; this will 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 is necessary.

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,do not call "TRKSET".

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 omitted.

6. Read or write the desired sector.

   Command the controller to read from or write to the disk by calling "DREAD" or "DWRITE" respectively.

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

EXAMPLE:

Suppose Sectors 5, 6, 7 and 8 of Track 12, Drive 1 are to be read to or from memory starting at location 700H. The following programs illustrate this procedure:

Example of disk read

```
read Page 1 Mon Dec 27 14:49:17 1982

F800 =          fdorig  equ     0f800h          ;Controller address
F800 =          dboot   equ     fdorig+0        ;Initialization + bootstrap
F803 =          termin  equ     fdorig+3        ;Serial input
F806 =          trmout  equ     fdorig+6        ;Serial output
F809 =          tkzero  equ     fdorig+9        ;Recalibrate (seek to TRK0)
F80C =          trkset  equ     fdorig+12       ;Seek
F80F =          setsec  equ     fdorig+15       ;Select sector
F812 =          setdma  equ     fdorig+18       ;Set DMA address
F815 =          dread   equ     fdorig+21       ;Read a sector of disk data
F818 =          dwrite  equ     fdorig+24       ;Write a sector of disk data
F81B =          seldrv  equ     fdorig+27       ;Select a disk drive
F81E =          tpanic  equ     fdorig+30       ;Test for panic character
F821 =          tstat   equ     fdorig+33       ;Serial status input
F824 =          dmast   equ     fdorig+36       ;Read current DMA address
F827 =          status  equ     fdorig+39       ;Disk status input
F82A =          dskerr  equ     fdorig+42       ;Loop to strobe error LED
F82D =          setden  equ     fdorig+45       ;Set density
F830 =          setsid  equ     fdorig+48       ;Set side for 2-headed drives

0100                    org     0100h

0100 310002     read:   lxi     sp,0200h        ;Set up the stack
0103 0E00               mvi     c,0             ;Select drive A
0105 CD1BF8             call    seldrv
0108 CD09F8             call    tkzero          ;Recalibrate the drive
010B 0E0C               mvi     c,12            ;Seek to track 12
010D CD0CF8             call    trkset
0110 010504             lxi     b,4*100h+ 5     ;4 sectors starting at sector 5
0113 C5                 push    b
0114 010007             lxi     b,700h          ;Set up the DMA address

0117 CD12F8     loop:   call    setdma
011A C1                 pop     b               ;Get sector to read in C
011B C5                 push    b
011C CD0FF8             call    setsec
011F CD15F8             call    dread           ;Read a sector
0122 DA3801             jc      error           ;Skip on error
0125 C1                 pop     b               ;Restore sector count
0126 05                 dcr     b               ;Bump counter
0127 CA3B01             jz      done
012A 0C                 inr     c               ;Bump the sector number
012B C5                 push    b
012C CD24F8             call    dmast           ;Load the DMA address into BC
```

```
read Page 2 Mon Dec 27 14:49:17 1982

 012F 210001                lxi     h,256             ;Assume 256 bytes/sector
 0132 09                    dad     b                 ;Go to the next sectors address
 0133 44                    mov     b,h               ;DMA address in BC for SETDMA
 0134 4D                    mov     c,l
 0135 C31701                jmp     loop

 0138 C33801      error:    jmp     $                 ;Error stop

 013B C33B01      done:     jmp     $                 ;Operation complete
```

```
write Page 1 Mon Dec 27 14:51:53 1982


F800 =          fdorig  equ     0f800h          ;Controller address
F800 =          dboot   equ     fdorig+0        ;Initialization + bootstrap
F803 =          termin  equ     fdorig+3        ;Serial input
F806 =          trmout  equ     fdorig+6        ;Serial output
F809 =          tkzero  equ     fdorig+9        ;Recalibrate (seek to TRK0)
F80C =          trkset  equ     fdorig+12       ;Seek
F80F =          setsec  equ     fdorig+15       ;Select sector
F812 =          setdma  equ     fdorig+18       ;Set DMA address
F815 =          dread   equ     fdorig+21       ;Read a sector of disk data
F818 =          dwrite  equ     fdorig+24       ;Write a sector of disk data
F81B =          seldrv  equ     fdorig+27       ;Select a disk drive
F81E =          tpanic  equ     fdorig+30       ;Test for panic character
F821 =          tstat   equ     fdorig+33       ;Serial status input
F824 =          dmast   equ     fdorig+36       ;Read current DMA address
F827 =          status  equ     fdorig+39       ;Disk status input
F82A =          dskerr  equ     fdorig+42       ;Loop to strobe error LED
F82D =          setden  equ     fdorig+45       ;Set density
F830 =          setsid  equ     fdorig+48       ;Set side for 2-headed drives

0100                    org     0100h

0100 310002     write:  lxi     sp,0200h        ;Set up the stack
0103 0E00               mvi     c,0             ;Select drive A
0105 CD1BF8             call    seldrv
0108 CD09F8             call    tkzero          ;Recalibrate the drive
010B 01007F             lxi     b,8000h-256     ;Set initial DMA address
010E CD12F8             call    setdma
0111 3E04               mvi     a,4             ;Initial track number

0113 324A01     tloop:  sta     temp            ;Save the track
0116 4F                 mov     c,a             ;Set the track
0117 CD0CF8             call    trkset
011A 01011A             lxi     b,26*100h + 1   ;26 sectors/track, sector 1

011D C5         sloop:  push    b               ;Save counts
011E CD24F8             call    dmast           ;Get DMA address
0121 210001             lxi     h,256           ;Assume 256 bytes/sector
0124 09                 dad     b               ;Make new DMA address
0125 44                 mov     b,h             ;DMA address in BC for SETDMA
0126 4D                 mov     c,l
0127 CD12F8             call    setdma          ;Set the DMA address
012A C1                 pop     b               ;Get sector to read in C
012B C5                 push    b
```

```
write Page 2 Mon Dec 27 14:51:53 1982


 012C CD0FF8             call    setsec
 012F CD18F8             call    dwrite                ;Write a sector
 0132 DA4701             jc      error                 ;Skip on error
 0135 C1                 pop     b                     ;Restore sector count
 0136 0C                 inr     c                     ;Bump the sector number
 0137 05                 dcr     b                     ;Bump the sector counter
 0138 C21D01             jnz     sloop

 013B 3A4A01             lda     temp                  ;Get the current track
 013E 3C                 inr     a                     ;Bump to the next track
 013F FE07               cpi     7                     ;Test for the last track
 0141 C21301             jnz     tloop

 0144 C34401     done:   jmp     $                     ;Operation complete

 0147 C34701     error:  jmp     $                     ;Error stop

 014A 00         temp:   db      0                     ;Current track
```

## 7.5. EPROM SUBROUTINES (Disk Controller)

Disk Controller EPROM subroutines are supplied with the DJ2D/B to ease system integration. These routines are written with CP/M requirements in mind.

### 7.5.1. TRKSET

The value in the C register of the CPU specifies the track over which the read/write head will be positioned 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 content of the C register is written into the RAM location TRACK. If the C register value is not within these bounds, no action is taken, the carry flag is set, and the subroutine returns to the calling program.

### 7.5.2. SECTOR

The value in the C register of the CPU specifies the sector that will be involved in the next Disk Read or Disk 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, the DJ2D/B firmware compares the value in SECTOR and the maximum number of sectors on the track to which the transfer is to occur. If the value in SECTOR exceeds the maximum number of sectors, the transfer operation is aborted and error information is reported.

### 7.5.3. 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. When the SETDMA subroutine is called, the starting address of a data block to be involved in the next Disk Transfer operation is specified by the B-C register pair.

Since the disk registers are memory mapped, the firmware is designed to try to protect the disk registers 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 DJ2D/B 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 DJ2D/B ROM during a subsequent disk operation, the content of the B-C pair is written into the memory location of the DJ2D/B RAM specified by the label DMAADR. The carry flag is cleared and the routine ends.

### 7.5.4. SELDRV

The value of the C register determines which of the possible four 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.

### 7.5.5. 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 is 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 determine which head will be selected for a transfer operation. A zero in Bit 0 will select the bottom head and a one 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.

### 7.5.6. SETDEN

The 1791 Floppy Disk Controller operates in two modes:

1) Single density FM (Frequency Modulation) mode or
2) Double density MFM (Modified Frequency Modulation) mode

Bit 0 of the C register determines the density in which the 1791 will operate when the next Disk Transfer operation is initiated (0=single,1=double).

Exercise care when using this routine. Under certain conditions, if the density is changed in between disk transfers occurring on the same track, the micro-program executed by the 1791 controller may fall into an error loop from which it could not recover. In such a case, the system MUST be RESET before further disk operations are performed.

The density mode of the 1791 can safely be changed when a subsequent Disk Transfer operation will occur on a track different from the last.

NOTE: The DJ2D/B firmware has the ability to automatically set the density mode of the 1791/8866. Whenever a new drive is to be selected, or the head is not loaded, the DJ2D/B 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 set the density correctly through use of SETDEN, the process of reading the sector header is slightly faster (by approximately one and one-half diskette revolutions) than it would be if the initial assumption concerning the density was wrong.

### 7.5.7. 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. No "Read Header" verify operation is performed. There are two side effects of positioning the head at Track Zero:

1. A flag is set in the DJ RAM, forcing a "Read Header" Density/Position Verify operation prior to the next disk transfer operation
2. The 1791/8866 controller will be forced to a single-density mode as long as Disk Transfer operations occur on Track Zero.

Track Zero of all IBM-compatible diskettes is formatted in single density. 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 DJ2D/B firmware will automatically switch back to double density when the head is moved away from Track Zero, without the intervention of external software.

### 7.5.8. READ

This subroutine transfers information from the diskette to memory. The procedure is outlined below:

1. Select the proper disk drive.
2. 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.
3. The "head loaded" flag is tested.

4. 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.

5. 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).

6. If the head load time-out bit is set, a 40 milli-second delay occurs to allow for the head to settle after loading.

7. The ready line from the drive is tested, insuring the drive has been selected properly and a diskette has been inserted.

   a. 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.

   b. 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.

8. The correct density of the track is determined, and the density mode is changed if necessary.

9. If the 1791 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.

10. When the Disk Jockey firmware positions the head to a new track, it reads a header both to determine the proper density and to determine the length and number of the sectors on the new track.

11. The DJ RAM location SECLEN is updated during read header operations and contains encoded data that determines both the number and the size of sectors on the current track.

12. 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.

    a. If the desired sector is too large, the carry flag is set and the routine returns with a 10H in the A register.

    b. If the value is acceptable, the data from this sector is 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 chip processes these bytes and then updates its status register.

    c. 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.

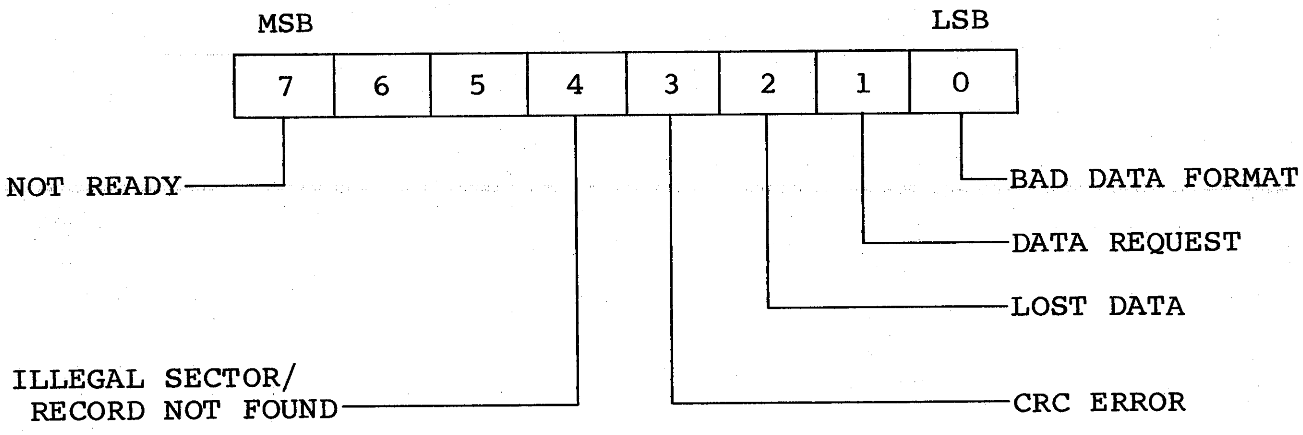


Figure 7-2: "DREAD" Register A Error Bits

7.5.9. DWRITE

The flow of logic for this routine is exactly the same as described above in the Read Data operation, up to the point of actual information transfer. If all the conditions for a data transfer as described above are satisfied, a Write Sector command is issued to the 1791 controller. Information is then transfered 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 are illustrated below.

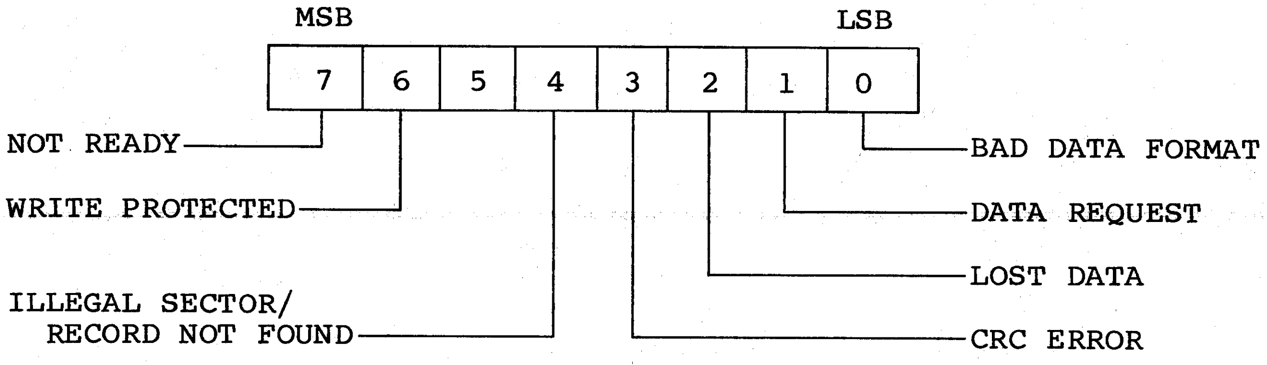


Figure 7-3: "DWRITE" Register A Error Bits

7.5.10. 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 fourth page of the DJ RAM (FF00H). The Bootstrap routine terminates with a branch to the first location of this block.

Typically, Sector 1 of Track 0 will contain another bootstrap program whose job it is to load a Disk Operating System (DOS) such as CP/M. If the Bootstrap Read is not successful, control is passed to the DSKERR utility which is described below.

Before Sector 1 is read into memory, various memory locations of the DJ RAM are initialized. Also, DBOOT performs 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 designed so that DBOOT could be started before a diskette was inserted in the drive.

When a diskette is inserted, the door should be closed just AFTER the activity light has been pulsed.

### 7.5.11. DMAST

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

### 7.5.12. STATUS

This subroutine performs the following functions:

1. Loads the B register with the sector number involved in the last disk transfer operation.
2. Loads the C register with the number of the track over which the head is currently positioned.
3. Finally, it loads the A register with a bit pattern indicating:
    a. the drive involved in the last disk transfer operation,

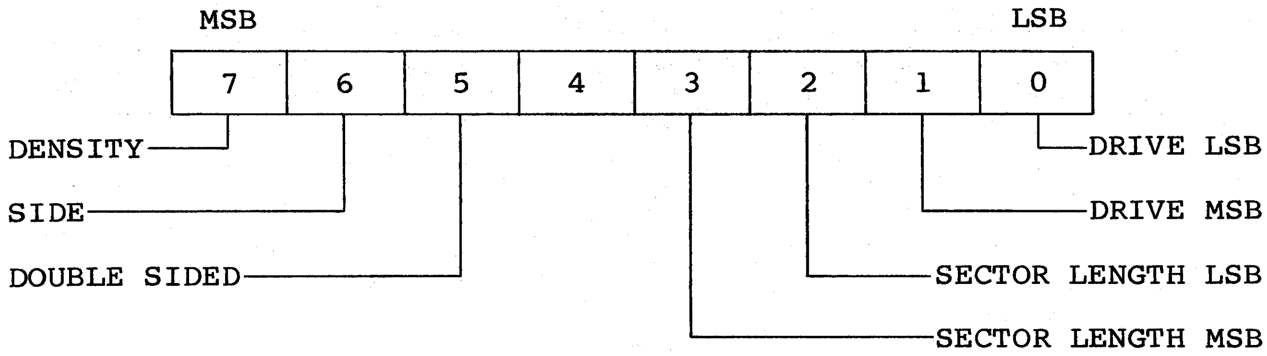
    b. the length of the sectors on the current track,

    c. the side specified by the last SETSID call,

    d. the density of the data during the most recent disk transfer operation,

    e. whether the drive selected during the most recent disk operation was double sided WITH double sided media in place.

Figure 7-4 details how this information is encoded in the A register.



| 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

Figure 7-4: A Register Bit Pattern

### 7.5.13. 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 (approximately) one-second intervals. This routine takes no parameters and will not return. Its primary function is to indicate a hard error occurrence during the Bootstrap Load operation.

MSB LSB

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

MSB LSB

| "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 | |

MSB LSB

| "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 | |

Figure 7-5: Recap of Register A Error Bits

# 8. SOFTWARE AND HARDWARE NOTES

## 8.1. OVERVIEW

The DJ2D/B controller has some unique hardware onboard to handle data transfers to and from the diskettes. Commands such as seeking and reading status, or issuing a command are asynchronous commands. This means that they may be executed at any time without regard to the data on the diskette. Commands which read or write data mean the CPU must be synchronized to the disk drive and the 1791 controller. Remember that on a read operation, the 1791 can hold only one byte from the disk, and this byte must be read by the CPU or an error will result. Likewise on a write operation, the CPU must always write the data to the 1791 controller, byte by byte, before the data is sent to the diskette. The DJ2D/B uses wait states to accomplish this synchronization.

## 8.2. WAIT STATES

A wait state means that the host CPU's ready line is asserted (pin 72 on the S-100 bus is brought low). This ready line is controlled by the 1791 DRQ (Data Request) bit, forcing the CPU to become synchronous with the 1791. Data is typically written and read from the disk at the rate of one byte every 16 usec. (MFM double density). This means that the CPU must read or write this data every 16 usec.

## 8.3. TYPICAL READ OPERATION

On a read operation, the CPU is entered into a tight instruction loop where it reads a byte of data from the controller and stores it until the INTRQ line on the 1791 goes high (indicating the command was completed). A typical read routine is provided on the following page.

```
               org 100

data    equ  0FBFFh                    ;1791 data register
status equ   0FBFAh                    ;1791 status register

          lxi  d,data                  ;pointer to the data register
          lxi  h,200h                  ;destination of read data
          mvi  c,ff                    ;byte count for 1024 bytes
rdloop:   ldax d                       ;get the data from 1791
          mov m,a                      ;store it
          inx  h                       ;bump pointer
          ldax d
          mov  m,a
          inx  h
          ldax d
          mov  m,a
          inx  h
          dcr  c                       ;bump count
          ldax d
          mov  m,a
          inx  h
          jnz  rdloop
          jmp 00

          end
```

It is important that the loop be tight to ensure that the instruction executes before the next byte from the disk to the 1791 arrives. The stall logic on the DJ2D/B hangs after each byte is read from the data register until the next DRQ from the 1791 arrives. The CPU then has 16 usec. to get the byte and store it; hence the tight loop requirements.

## 8.4. TYPICAL WRITE OPERATION

The write operation is the same procedure in reverse. The following routine is a typical write data to disk routine:

```
data   equu  0FBFFh                 ;1791 data register
status equ   0FBFAh                 ;1791 status register

          lxi  d,data               ;pointer to the data register
          lxi  h,200h               ;destination of read data
          mvi  c,ff                 ;byte count for 1024 bytes

wrloop:   mov  a,m                  ;get write data from ram
          inx  h
          stax d                    ;write it to disk
          mov  a,m
          inx  h
          stax d
          mov  a,m
          inx  h
          stax d
          dcr  c                    ;bump count
          mov  a,m
          inx  h
          stax d
          jnz  wrloop
          jmp  00

          end
```

Appendix A: Power-On Jump and Drive Control Summary

## A. POWER-ON JUMP AND DRIVE CONTROL SUMMARY

### POWER-ON JUMP TABLE

| JUMP ADDRESS | SWITCH SETTING | | | | |
|---|---|---|---|---|---|
| Hex | SW1-1 (A15) | SW1-2 (A14) | SW1-3 (A13) | SW1-4 (A12) | SW1-5 (A11) |
| 0000 | on | on | on | on | on |
| 0800 | on | on | on | on | off |
| 1000 | on | on | on | off | on |
| 1800 | on | on | on | off | off |
| 2000 | on | on | off | on | on |
| 2800 | on | on | off | on | off |
| 3000 | on | on | off | off | on |
| 3800 | on | on | off | off | off |
| 4000 | on | off | on | on | on |
| 4800 | on | off | on | on | off |
| 5000 | on | off | on | off | on |
| 5800 | on | off | on | off | off |
| 6000 | on | off | off | on | on |
| 6800 | on | off | off | on | off |
| 7000 | on | off | off | off | on |
| 7800 | on | off | off | off | off |
| 8000 | off | on | on | on | on |
| 8800 | off | on | on | on | off |
| 9000 | off | on | on | off | on |
| 9800 | off | on | on | off | off |
| A000 | off | on | off | on | on |
| A800 | off | on | off | on | off |
| B000 | off | on | off | off | on |
| B800 | off | on | off | off | off |
| C000 | off | off | on | on | on |
| C800 | off | off | on | on | off |
| D000 | off | off | on | off | on |
| D800 | off | off | on | off | off |
| E000 | off | off | off | on | on |
| E800 | off | off | off | on | off |
| F000 | off | off | off | off | on |
| F800 | off | off | off | off | off |

## DRIVE CONTROL

The DJ2D/B has been designed to interface with industry standard 8 inch floppy drive via a 50 pin flat cable. This cable plugs into the DJ2D/B at the top of the board (connector P2).

All signal lines coming from the disk drives are terminated (pulled up to +5 volts through a 180 resistor). All lines which go to the disk drives are driven with TTL type drivers capable of sinking 24 ma. All odd numbered lines on the 50 pin flat cable are grounded on the controller to reduce cable crosstalk.

### DJ2D/B Drive Interface Signal Summary

| Pin | Direction | Description |
|---|---|---|
| 46 | from drive | Read Data - Composite clock and data from the read heads of the drive. |
| 44 | from drive | Write Protect - low when the diskette currently in drive is write protected. |
| 42 | from drive | Track 0 - low when the heads of the drive have reached track 0 |
| 40 | to drive | Write Gate - When low indicates the controller is writing data to the drive on the write data line. |
| 38 | to drive | Write Data - contains the data which will be recorded on the diskette when write gate is active. |
| 36 | to drive | Step - low going pulse commands the heads of the drive to move to the next track. |
| 34 | to drive | Direction - when high the heads will move to track zero when a step pulse is issued. When low, the heads will move away from track 0 when a step pulse is issued. |
| 32 | to drive | Drive Select 4 - When low enables drive 4 to be accessed. |
| 30 | to drive | Drive Select 3 - When low enables drive 3 to be accessed. |
| 28 | to drive | Drive Select 2 - When low enables drive 2 to be accessed. |

| | | |
|---|---|---|
| 26 | to drive | Drive Select 1 - When low enables drive 1 to be accessed. |
| 22 | from drive | Ready - low when the selected drive is ready to accept a command (diskette is inserted correctly and is spinning at the correct speed). |
| 20 | from drive | Index - indicates to the controller that the diskette in the selected drive has completed one revolution. |
| 18 | to drive | Head Load - when low will press the heads against the media in the selected drive. |
| 16* | to drive | In use - when low will cause the activity light on the front door to light (used on single sided drives only). |
| 14* | to drive | Side Select - used on double sided drives to select which side will be accessed. When high, side 0 will be selected; when low, side 1 will select side 1. |
| 10 | from drive | Double Sided - when low indicates the selected drive is double sided and has a double sided diskette installed. |
| 2 | to drive | Low current - when low reduces the write current applied to the head during a write operation. Normally this line is brought low when writing on tracks beyond 43 due to the increase in bit density on the innermost tracks. |

* These lines are connected together on the DJ PC board.

## B. P1 AND P2 PIN CONNECTIONS

This appendix lists the the pin connections of P1 and P2. Note that the top of the circuit board is assumed to be to the right of the table. 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.

| | | P2 |
|---|---|---|
| RS232 GROUND | * | 1 |
| RS232 INPUT | * | 2 |
| RS232 OUTPUT | * | 3 |
| TTY+ INPUT | * | 4 |
| TTY- INPUT | * | 5 |
| TTY+ OUTPUT | * | 6 |
| TTY- OUTPUT | * | 7 |

| | | P1 | | |
|---|---|---|---|---|
| | 50 | * * | 49 | GND |
| | 48 | * * | 47 | GND |
| -DISK DATA | 46 | * * | 45 | GND |
| -WRITE PROTECT | 44 | * * | 43 | GND |
| -TRACK ZERO | 42 | * * | 41 | GND |
| -WRITE GATE | 40 | * * | 39 | GND |
| -WRITE DATA | 38 | * * | 37 | GND |
| -STEP | 36 | * * | 35 | GND |
| -DIRECTION | 34 | * * | 33 | GND |
| -DRIVE SELECT 4 | 32 | * * | 31 | GND |
| -DRIVE SELECT 3 | 30 | * * | 29 | GND |
| -DRIVE SELECT 2 | 28 | * * | 27 | GND |
| -DRIVE SELECT 1 | 26 | * * | 25 | GND |
| -SECTOR | 24 | * * | 23 | GND |
| -READY | 22 | * * | 21 | GND |
| -INDEX | 20 | * * | 19 | GND |
| -LOAD HEAD | 18 | * * | 17 | GND |
| -IN USE | 16 | * * | 15 | GND |
| | 14 | * * | 13 | GND |
| | 12 | * * | 11 | GND |
| -TWO SIDED | 10 | * * | 9 | GND |
| | 8 | * * | 7 | GND |
| | 6 | * * | 5 | GND |
| | 4 | * * | 3 | GND |
| | 2 | * * | 1 | GND |

## C. MOLEX PIN CONNECTOR ASSEMBLY INSTRUCTIONS

A Molex pin connector kit is included with each DJ2D/B board purchased from Morrow Designs. This connector facilitates use of P2 as an RS-232 serial port. Instructions for assembling this connector follow.

Tools needed:
- Wire clipper
- Wire stripper
- Needlenose pliers
- RS-232 25-pin ("D" style) male connector
- 3 ft. of RS-232 cable

Each kit includes a 7 pin connector and 7 single barb spring pins. The spring pins are clamped onto the ends of the wires extending from the RS-232 connector. These wires are then inserted into the corresponding slots of the Molex connector.

Only three wires from the RS-232 connector are used to connect a terminal to the DJ2D/B. The following tables detail the pinouts of the RS-232 connector and the corresponding connections to the Molex connector:

Top

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Left | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| | | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 |

Table C-1: RS-232 Male Connector Pinout

| Molex pin connector | Corresponds to | RS-232 pin connector and signal | |
|---|---|---|---|
| 1 | | 1 | Ground |
| 2 | | 2 | RS-232 INPUT |
| 3 | | 3 | RS-232 OUTPUT |
| 4 | | 4 | *(TTY +IN |
| 5 | | 5 | TTY -IN |
| 6 | | 6 | --- |
| 7 | | 7 | Ground) |

Table C-2: Corresponding Molex Connections

* Pins 4, 5 and 7 are used only if connecting the DJ2D/B to a teletype device; pin 6 is normally not used.

# Appendix C: Molex Pin Connector Assembly Instructions

1. Begin by making sure pins 1,2 and 3 are connected correctly on the 25 pin RS-232 connector. It may be necessary to undo the casing on the RS-232 connector.

2. Make note of the color of the wires connected to the pins on the RS-232 connector. Beginning with the wire leading from pin 1, strip off 1/8 inch of insulation from the end of the wire.

3. Place the wire end in a spring pin. Notice the two clamps. Position the insulation part of the wire at the first clamp (at the end of the spring pin). Use needlenose pliers to crimp the wire securely in place. Next crimp the bare wire securely under the second clamp, making sure there are no stray wires.

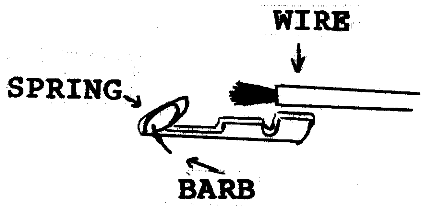


Figure C-1: Spring Pin Connections

4. Follow the above instructions for pins 2 and 3.

5. Insert the wires with the spring pins attached into the corresponding slots on the back side of the Molex connector. Looking from the rear, pin 7 is at the left side of the Molex connector, pin 1 is on the right.

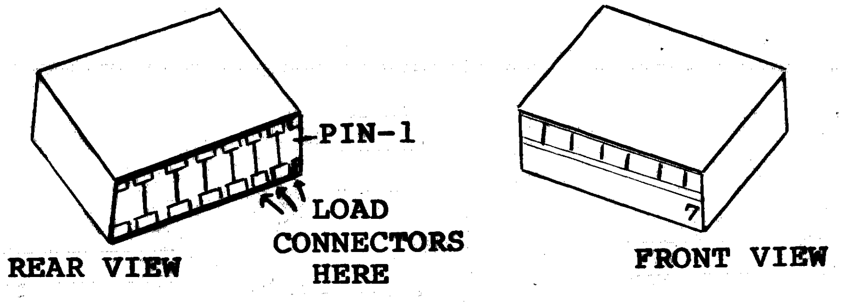


Figure C-2: Molex Connector Orientation

6. Because of the barb, there is only one way these pins may be inserted. Push the spring pin through the slot until it locks in place. Test the connection by pulling back on the wire. If it is not secure, it may be necessary to pull it out, widen the gap between the barb and the pin, then re-insert it.

7. Double check that the wires are in the correct slots, then place the Molex connector on P2, making sure that the pins correspond to the correct slots on the connector. Facing the plain side of the Molex connector (side without the part number), pin 7 is at the right, pin 1 on the left. Refer to the component layout for the pin orientation of P2.

## D. TROUBLESHOOTING TABLE

| PROBLEM | PROBABLE CAUSE | REMEDY/REFER TO |
|---|---|---|
| DRIVE INDICATOR LIGHT ALWAYS ON (DOES NOT BLINK): | 50-pin cable connected to Pl incorrectly. | Sections 3.8 - 3.10 |
| | 50-pin cable connected to drive incorrectly. | Sections 3.8 - 3.10 |
| | Incorrect jumper settings on drive pc board. | Disk drive OEM manual. |
| | Incorrect jumper settings on DJ2D/B. | Sections 3.5-3.6 |
| | CPU power on jump address set incorrectly. | Sections 3.4, 5.4 |
| | System bus speed faster than 6 MHz. | Section 5.6 |
| | Memory conflict with DJ2D/B EPROM and system memory. | Using front panel or monitor, examine locations of EPROM jump table (F800h). The locations from F800-F80Eh should contain the following hex bytes:<br>C3 69 F8 C3 E9 F8<br>C3 DA F8 C3 5A F9<br>C3 8B F9 |
| INDICATOR LIGHT BLINKS, BUT HEADS DO NOT LOAD: | Disk drive not jumpered correctly. | Disk drive OEM manual |
| | Faulty cable. | Try a new cable. |
| | Diskette not inserted. | Section 4 |
| DRIVE INDICATOR LIGHT NOT ON: | No AC power to drives. | Check AC power cord connection. |

Appendix D: Troubleshooting Table

| PROBLEM | PROBABLE CAUSE | REMEDY/REFER TO |
|---|---|---|
| MORE THAN ONE DRIVE INDICATOR LIGHT BLINKING IN A MULTI-DRIVE SYSTEM: | Drive select jumpers on drive pc board set incorrectly. | Disk drive OEM manual; Sect. 3.9.2. |
| | Multiple drives terminated as drive A: | Disk drive OEM manual; Sect. 3.9.2. |
| DRIVE LOADS, SEEKS, BUT SCREEN IS BLANK: | Diskette inserted incorrectly. | Section 4 |
| | Incorrect diskette; use CP/M system diskette. | Section 4 |
| | Terminal is off. | Turn it on. |
| | Wires 2 and 3 on Molex connector switched. | Appendix C |
| GARBLED CHARACTERS ON TERMINAL: | No ground wire, or wrong wire grounded on Molex connector. | Appendix C |
| | Serial I/O settings incorrect. | Sections 3.4, 5.7 |
| ERROR LIGHT ON DJ2D/B FLASHES DURING BOOT SEQUENCE: | Diskette inserted incorrectly. | Section 4 |
| | Corrupted system tracks on boot diskette. | Make/use another diskette. |
| | Faulty 50-pin cable. | Try a new cable. |
| | Drives not terminated correctly. | Section 3.9 and disk drive manual. |
| | Disk drives dirty or out of alignment. | Contact Customer Service. |

# SOFTWARE LISTINGS

```
                ***********************************************************
                *                                                         *
                * Firmware for Disk Jockey Model B.                       *
                *                                                         *
                ***********************************************************

 F800 =         ORIGIN  EQU     0F800H

 F800                   ORG     ORIGIN

 FC00 =         RAM     EQU     ORIGIN+400h
 FBF8 =         IO      EQU     ORIGIN+3f8h
 FBF8 =         UDATA   EQU     IO
 FBF9 =         DREG    EQU     IO+1
 FBF9 =         USTAT   EQU     IO+1
 FBFA =         DCMD    EQU     IO+2
 FBFA =         DSTAT   EQU     IO+2
 FBFB =         CSTALL  EQU     IO+3
 FBFC =         CMDREG  EQU     IO+4
 FBFC =         CSTAT   EQU     IO+4
 FBFD =         TRKREG  EQU     IO+5
 FBFE =         SECREG  EQU     IO+6
 FBFF =         DATREG  EQU     IO+7

 0001 =         LIGHT   EQU     1
 0001 =         HEAD    EQU     1
 0001 =         DENSITY EQU     1
 0004 =         INTRQ   EQU     4
 0004 =         ISTAT   EQU     4
 0004 =         TZERO   EQU     4
 0004 =         LOAD    EQU     4
 0006 =         ULOAD   EQU     6
 0008 =         OSTAT   EQU     10Q
 0008 =         DSIDE   EQU     10Q
 0009 =         NOLITE  EQU     11Q
 0009 =         DCRINT  EQU     11Q
 0009 =         HCMD    EQU     11Q
 0010 =         INDEX   EQU     20Q
 0012 =         WINDXD  EQU     22Q
 0018 =         SKCMD   EQU     30Q
 001A =         RINDXD  EQU     32Q
 001D =         SVCMD   EQU     35Q
 0040 =         WPROT   EQU     100Q
 0040 =         ACCESS  EQU     100Q
 0080 =         RSTBIT  EQU     200Q
 0080 =         READY   EQU     200Q
 0088 =         RDCMD   EQU     210Q
 00A8 =         WRCMD   EQU     250Q
 00C0 =         STBITS  EQU     300Q
 00C4 =         RACMD   EQU     304Q
 00D0 =         CLRCMD  EQU     320Q

 F800 C369F8    DBOOT   JMP     BOOT
 F803 C3E9F8    TERMIN  JMP     CIN
 F806 C3DAF8    TRMOUT  JMP     COUT
```

```
F809 C35AF9     TKZERO  JMP     HOME
F80C C38BF9     TRKSET  JMP     SEEK
F80F C381F9     SETSEC  JMP     SECSET
F812 C343F9     SETDMA  JMP     DMA
F815 C3DDF9     DREAD   JMP     READ
F818 C3BCF9     DWRITE  JMP     WRITE
F81B C33CF9     SELDRV  JMP     DRIVE
F81E C3F8F8     TPANIC  JMP     CPAN
F821 C303F9     TSTAT   JMP     TMSTAT
F824 C334F9     DMAST   JMP     DMSTAT
F827 C309F9     STATUS  JMP     DISKST
F82A C3C5F8     DSKERR  JMP     LERROR
F82D C3B3FB     SETDEN  JMP     DENFIX
F830 C3E5FB     SETSID  JMP     SIDEFX


F833                    DS      66Q

F869 31FAFF     BOOT    LXI     SP,TRACK+1      ;initialize SP
F86C CDD2FB             CALL    TIMOUT          ;poc/reset timeout
F86F 210100             LXI     H,1
F872 E5                 PUSH    H               ;track 0, sector 1
F873 2E09               MVI     L,DCRINT        ;set up the
F875 E5                 PUSH    H               ;    side select
F876 26FF               MVI     H,377Q          ;    and initial
F878 E5                 PUSH    H               ;    drive
F879 E5                 PUSH    H               ;    parameters
F87A E5                 PUSH    H
F87B E5                 PUSH    H
F87C 210800             LXI     H,10Q           ;initialize
F87F E5                 PUSH    H               ;    tzflag & cdisk
F880 2E7E               MVI     L,176Q          ;initialize
F882 E5                 PUSH    H               ;    disk & drvsel
F883 2E08               MVI     L,10Q           ;initialize
F885 E5                 PUSH    H               ;    hdflag & dsflag
F886 2618               MVI     H,30Q           ;initialize
F888 E5                 PUSH    H               ;    timer constant
F889 3E7F               MVI     A,177Q          ;start 1791
F88B 32F9FB             STA     DREG
F88E 3ED0               MVI     A,CLRCMD        ;1791 reset
F890 32FCFB             STA     CMDREG
                LDHEAD
F893 AF                 XRA     A               ;load the head
F894 CD1BFB             CALL    HDCHK           ;    and test for
F897 D2A5F8             JNC     DOOROK          ;    drive ready
F89A 3E01               MVI     A,LIGHT         ;turn on the
F89C 32F6FF             STA     DCREG           ;    error LED
F89F CDD2FB             CALL    TIMOUT          ;timeout to
F8A2 C393F8             JMP     LDHEAD          ;    close drive door
                DOOROK
F8A5 3E09               MVI     A,NOLITE        ;turn off the
F8A7 32F6FF             STA     DCREG           ;    error LED
F8AA CD96FB             CALL    MEASUR          ;head load time
F8AD C1                 POP     B               ;adjust the stack
F8AE 0100FF             LXI     B,RAM+300H      ;DMA addr
F8B1 C5                 PUSH    B               ;initialize
F8B2 D5                 PUSH    D               ;    dmaadr & timer
```

```
 F8B3 210000            LXI     H,0             ;initialize
 F8B6 E5                PUSH    H               ;    intrfg & ramins
 F8B7 00                NOP                     ;debug instruction
 F8B8 C5                PUSH    B               ;boot address
 F8B9 060C              MVI     B,12            ;number of retrys
                LDLOOP
 F8BB C5                PUSH    B               ;save the retry no.
 F8BC CDDDF9            CALL    READ            ;read boot sector
 F8BF C1                POP     B               ;restor retry no.
 F8C0 D0                RNC                     ;successful read?
 F8C1 05                DCR     B               ;no - count down
 F8C2 C2BBF8            JNZ     LDLOOP          ;    and try again
                LERROR
 F8C5 0E09              MVI     C,11Q
 F8C7 11C3A2            LXI     D,0a2c3h
                LELOOP
 F8CA 1B                DCX     D
 F8CB 7A                MOV     A,D
 F8CC B3                ORA     E
 F8CD C2CAF8            JNZ     LELOOP
 F8D0 3E08              MVI     A,10Q           ;blink
 F8D2 A9                XRA     C               ;    the LED at
 F8D3 4F                MOV     C,A             ;    top of the
 F8D4 32FAFB            STA     DCMD            ;    circuit board
 F8D7 C3C7F8            JMP     LERROR+2

                COUT
 F8DA 3AF9FB            LDA     USTAT           ;get UART status
 F8DD E608              ANI     OSTAT           ;output ready mask
 F8DF C2DAF8            JNZ     COUT            ;test buffer empty
 F8E2 79                MOV     A,C             ;character data
 F8E3 2F                CMA                     ;negative logic bus
 F8E4 32F8FB            STA     UDATA           ;send data to UART
 F8E7 2F                CMA                     ;make positive
 F8E8 C9                RET

                CIN
 F8E9 3AF9FB            LDA     USTAT           ;get UART status
 F8EC E604              ANI     ISTAT           ;input ready mask
 F8EE C2E9F8            JNZ     CIN             ;wait for input
 F8F1 3AF8FB            LDA     UDATA           ;get the character
 F8F4 2F                CMA                     ;adjust for negative bus
 F8F5 E67F              ANI     177Q            ;trim to 7 bits
 F8F7 C9                RET

                CPAN
 F8F8 3AF9FB            LDA     USTAT           ;get UART status
 F8FB E604              ANI     ISTAT           ;input ready mask
 F8FD C0                RNZ                     ;test for data
 F8FE CDE9F8            CALL    CIN             ;get character
 F901 B9                CMP     C               ;test for panic chtr
 F902 C9                RET
                TMSTAT
 F903 3AF9FB            LDA     USTAT           ;get UART status
 F906 E604              ANI     ISTAT           ;input ready mask
 F908 C9                RET
```

```
                DISKST
 F909 21FDFB            LXI     H,TRKREG        ;most recent
 F90C 4E                MOV     C,M             ;    track to C
 F90D 23                INX     H               ;most recent
 F90E 46                MOV     B,M             ;    sector to B
 F90F 3AF6FF            LDA     DCREG           ;get current
 F912 2F                CMA                     ;    density in
 F913 E601              ANI     1               ;    the msb
 F915 0F                RRC                     ;    position
 F916 57                MOV     D,A             ;save in D
 F917 3AF7FF            LDA     SIDE            ;put the
 F91A 07                RLC                     ;    most recent
 F91B 07                RLC                     ;    side select
 F91C 07                RLC                     ;    in bit positin
 F91D B2                ORA     D               ;    6 and merge
 F91E 57                MOV     D,A             ;save in D
 F91F 3AE8FF            LDA     DSFLAG          ;get the
 F922 EE08              XRI     DSIDE           ;    most recent
 F924 17                RAL                     ;    double sided
 F925 17                RAL                     ;    status and place
 F926 82                ADD     D               ;    in bit position
 F927 57                MOV     D,A             ;    5 and merge
 F928 3AFDFF            LDA     SECLEN          ;get the
 F92B 17                RAL                     ;    sector length
 F92C 17                RAL                     ;    code bits in
 F92D B2                ORA     D               ;    positions 2 & 3
 F92E 57                MOV     D,A             ;    and merge
 F92F 3AECFF            LDA     CDISK           ;get the current
 F932 82                ADD     D               ;    disk no. in bit
 F933 C9                RET                     ;    positions 0 & 1

                DMSTAT
 F934 E5                PUSH    H               ;save the HL pair
 F935 2AE6FF            LHLD    DMAADR          ;move the
 F938 44                MOV     B,H             ;    DMA address to
 F939 4D                MOV     C,L             ;    the BC pair
 F93A E1                POP     H               ;recover HL
 F93B C9                RET

                DRIVE
 F93C 79                MOV     A,C             ;drive select
 F93D E603              ANI     3               ;    values must be
 F93F 32EBFF            STA     DISK            ;    between zero
 F942 C9                RET                     ;    and three

                DMA
 F943 210004            LXI     H,-RAM          ;test the
 F946 09                DAD     B               ;    DMA address
 F947 DA54F9            JC      DMASET          ;    for conflict
 F94A 210808            LXI     H,8-ORIGIN
 F94D 09                DAD     B               ;    with the I/O
 F94E D254F9            JNC     DMASET          ;    on the DJ/2D
 F951 3E10              MVI     A,20Q           ;    controller
 F953 C9                RET
                DMASET
 F954 60                MOV     H,B             ;store the
 F955 69                MOV     L,C             ;    BC pair
 F956 22E6FF            SHLD    DMAADR
 F959 C9                RET

               HOME
 F95A CDE3FA            CALL    HDLOAD          ;load the head
 F95D D8                RC                      ;not ready error
 F95E CD70F9            CALL    HENTRY          ;move the head
 F961 F5                PUSH    PSW             ;save status
 F962 9F                SBB     A               ;update the
 F963 32F9FF            STA     TRACK           ;    track
 F966 32FDFB            STA     TRKREG          ;    registers
 F969 AF                XRA     A               ;set the not
 F96A 32EDFF            STA     TZFLAG          ;    verified flag
 F96D C323FA            JMP     LEAVE+1         ;unload the head
               HENTRY
 F970 AF                XRA     A               ;set the force
 F971 32E9FF            STA     HDFLAG          ;    verify flag
 F974 210000            LXI     H,0             ;timeout constant
 F977 3E09              MVI     A,HCMD          ;move the head
 F979 CD62FB            CALL    CENTRY          ;to track 0
 F97C E604              ANI     TZERO           ;track zero bit
 F97E C0                RNZ
 F97F 37                STC                     ;error flag
 F980 C9                RET

               SECSET
 F981 AF                XRA     A               ;test for
 F982 B1                ORA     C               ;    zero value
 F983 37                STC                     ;error flag
 F984 C8                RZ                      ;error return
 F985 E61F              ANI     37Q             ;trim & clear cry
 F987 32F8FF            STA     SECTOR
 F98A C9                RET

               SEEK
 F98B 79                MOV     A,C             ;test for
 F98C FE4D              CPI     77              ;    track
 F98E 3F                CMC                     ;    too large
 F98F D8                RC
 F990 32F9FF            STA     TRACK
 F993 C9                RET

 F994 32E3FF   ISSUE    STA     ECOUNT+1        ;update count
 F997 CD96FB            CALL    MEASUR          ;find index
 F99A 0E01              MVI     C,1             ;Start with sector 1
               ISLOOP
 F99C 79                MOV     A,C             ;Initilize the
 F99D 32FEFB            STA     SECREG          ;    sector register
 F9A0 3AF8FF            LDA     SECTOR          ;Test for
 F9A3 B9                CMP     C               ;    target sector
 F9A4 C8                RZ
 F9A5 3E88              MVI     A,RDCMD         ;do a fake
 F9A7 CD5DFB            CALL    COMAND          ;    read command
 F9AA DA20FA            JC      PLEAVE          ;Abort on error
 F9AD 0C                INR     C               ;Increment sector number
 F9AE C39CF9            JMP     ISLOOP
```

```
F9B1 32FCFB    COMNDP  STA     CMDREG          ;Start operation
F9B4 48                MOV     C,B             ;Initilize block count
F9B5 11FFFB            LXI     D,DATREG        ;Data register
F9B8 2AE6FF            LHLD    DMAADR          ;transfer address
F9BB C9                RET

               WRITE
F9BC CD33FA            CALL    PREP            ;prepare for write
F9BF DA22FA            JC      LEAVE           ;abort operation
               WRENTRY
F9C2 3EA8              MVI     A,WRCMD         ;write sector cmd
F9C4 CDB1F9            CALL    COMNDP
               WRLOOP
F9C7 7E                MOV     A,M             ;load 1st byte of data
F9C8 23                INX     H
F9C9 12                STAX    D               ;write 1st byte of data
F9CA 7E                MOV     A,M             ;load 2nd byte of data
F9CB 23                INX     H
F9CC 12                STAX    D               ;write 2nd byte of data
F9CD 7E                MOV     A,M             ;load 3rd byte of data
F9CE 23                INX     H
F9CF 12                STAX    D               ;write 3rd byte of data
F9D0 0D                DCR     c               ;reduce block count
F9D1 7E                MOV     A,M
F9D2 23                INX     H
F9D3 12                STAX    D
F9D4 C2C7F9            JNZ     WRLOOP          ;write next 4 bytes
F9D7 21C2F9            LXI     H,WRENTRY
F9DA C3FBF9            JMP     CBUSY

               READ
F9DD CD33FA            CALL    PREP            ;prepare for read
F9E0 DA22FA            JC      LEAVE           ;abort operation
               RDENTRY
F9E3 3E88              MVI     a,RDCMD
F9E5 CDB1F9            CALL    COMNDP
               RDLOOP
F9E8 1A                LDAX    D               ;read 1st byte
F9E9 77                MOV     M,A             ;store 1st byte
F9EA 23                INX     H
F9EB 1A                LDAX    D               ;read 2nd byte
F9EC 77                MOV     M,A             ;store 2nd byte
F9ED 23                INX     H
F9EE 1A                LDAX    D               ;read 3rd byte
F9EF 77                MOV     M,A             ;store 3rd byte
F9F0 23                INX     H
F9F1 0D                DCR     c               ;reduce block count
F9F2 1A                LDAX    D               ;read 4th byte
F9F3 77                MOV     M,A             ;store 4th byte
F9F4 23                INX     H
F9F5 C2E8F9            JNZ     RDLOOP          ;read next 4 bytes
F9F8 21E3F9            LXI     H,RDENTRY

               CBUSY
F9FB E5                PUSH    H               ;Save return
F9FC 21FCFB            LXI     H,CSTAT         ;Wait for 1791
F9FF CD6CFB            CALL    BUSY            ;     to finish command
```

```
 FA02 E65F                ANI      137Q            ;Error mask
 FA04 CA21FA              JZ       LEAVE-1         ;No error
 FA07 FE10                CPI      20Q             ;Premature interupt
 FA09 C220FA              JNZ      PLEAVE          ;Other type of error
 FA0C 3AE2FF              LDA      ECOUNT          ;decrement error count
 FA0F 3D                  DCR      A               ;     by one
 FA10 FA17FA              JM       STEST           ;Hard interupt error
 FA13 32E2FF              STA      ECOUNT          ;Update count
 FA16 C9                  RET
                STEST
 FA17 3AE3FF              LDA      ECOUNT+1        ;Decrement error
 FA1A 3D                  DCR      A
 FA1B F294F9              JP       ISSUE
 FA1E 3E10                MVI      A,20Q
                PLEAVE
 FA20 37                  STC                      ;error flag
 FA21 E1                  POP      H

                LEAVE
 FA22 F5                  PUSH     PSW             ;save the status
 FA23 3AF6FF              LDA      DCREG           ;control bits
 FA26 EE04                XRI      LOAD            ;toggle the
 FA28 32FAFB              STA      DCMD            ;    head load bit
 FA2B 3AEAFF              LDA      DRVSEL          ;enable access to
 FA2E 32F9FB              STA      DREG            ;    the data register
 FA31 F1                  POP      PSW             ;recover the status
 FA32 C9                  RET

                PREP
 FA33 CDE3FA              CALL     HDLOAD          ;load the head
 FA36 D8                  RC                       ;test for drive ready
 FA37 3AFDFB              LDA      TRKREG          ;get old track
 FA3A 3C                  INR      A               ;test for head
 FA3B CC70F9              CZ       HENTRY          ;    not calibrated
 FA3E D8                  RC                       ;seek error?
 FA3F 21FDFB              LXI      H,TRKREG        ;old track
 FA42 3AF9FF              LDA      TRACK           ;new track
 FA45 BE                  CMP      M               ;test for head motion
 FA46 23                  INX      H               ;advance to the
 FA47 23                  INX      H               ;    data register
 FA48 77                  MOV      M,A             ;save new track
 FA49 79                  MOV      A,C             ;turn off data reg
 FA4A 32F9FB              STA      DREG            ;    access control bit
 FA4D CA6AFA              JZ       TVERFY          ;test for seek
 FA50 AF                  XRA      A               ;force a read
 FA51 32E9FF              STA      HDFLAG          ;    header operation
 FA54 3AFAFB              LDA      DSTAT           ;get the double
 FA57 E608                ANI      DSIDE           ;    sided flag
 FA59 32E8FF              STA      DSFLAG          ;save for status
 FA5C 1F                  RAR                      ;shift for
 FA5D 1F                  RAR                      ;    3/6 ms step
 FA5E 1F                  RAR                      ;    rate constant
 FA5F C618                ADI      SKCMD           ;do a
 FA61 210000              LXI      H,0             ;    seek
 FA64 CD62FB              CALL     CENTRY          ;    operation
 FA67 DA8EFA              JC       SERROR          ;seek error?
                TVERFY
```

```
FA6A 3AE9FF                 LDA     HDFLAG          ;get the force
FA6D B7                     ORA     A               ;    verify hdr flag
FA6E C2B9FA                 JNZ     CHKSEC          ;no seek & head OK
FA71 0602                   MVI     B,2             ;verify retry count
                    SLOOP
FA73 3E1D                   MVI     A,SVCMD         ;do a verify
FA75 CD5DFB                 CALL    COMAND          ;    command
FA78 E699                   ANI     231Q            ;error bit mask
FA7A 57                     MOV     D,A             ;save
FA7B CA95FA                 JZ      RDHDR           ;no error
FA7E 3AF6FF                 LDA     DCREG           ;denisty control
FA81 EE01                   XRI     DENSITY         ;flip the density
FA83 32F6FF                 STA     DCREG           ;update and
FA86 32FAFB                 STA     DCMD            ;    change density
FA89 05                     DCR     B               ;decrement retry
FA8A C273FA                 JNZ     SLOOP           ;    count & test
FA8D 7A                     MOV     A,D
                    SERROR
FA8E 37                     STC
FA8F F5                     PUSH    PSW
FA90 CD70F9                 CALL    HENTRY
FA93 F1                     POP     PSW
FA94 C9                     RET
                    RDHDR
FA95 060A                   MVI     B,12Q           ;number of retrys
                    RHLOOP
FA97 11FFFB                 LXI     D,DATREG        ;Data register
FA9A 21FAFF                 LXI     H,TRACK+1       ;Data pointer
FA9D 3EC4                   MVI     A,RACMD         ;Start read header command
FA9F 32FCFB                 STA     CMDREG
                    RHL1
FAA2 1A                     LDAX    D               ;get disk data 0
FAA3 77                     MOV     M,A             ;store in mem
FAA4 2C                     INR     L               ;advance pointer
FAA5 C2A2FA                 JNZ     RHL1            ;test end of page
FAA8 21FCFB                 LXI     H,CSTAT         ;wait for 1791
FAAB CD6CFB                 CALL    BUSY            ;    to finish cmd
FAAE B7                     ORA     A               ;test for errors
FAAF CAB9FA                 JZ      CHKSEC          ;transfer OK?
FAB2 05                     DCR     B               ;no - test for
FAB3 C297FA                 JNZ     RHLOOP          ;    hard error
FAB6 C38EFA                 JMP     SERROR          ;recalibrate
                    CHKSEC
FAB9 3AFDFF                 LDA     SECLEN          ;get the sector
FABC 4F                     MOV     C,A             ;    size and setup
FABD 0600                   MVI     B,0             ;    the table offset
FABF 21DFFA                 LXI     H,STABLE        ;sector table
FAC2 09                     DAD     B               ;sector size pntr
FAC3 3AF8FF                 LDA     SECTOR          ;get the sector
FAC6 47                     MOV     B,A             ;    and save in B
FAC7 86                     ADD     M               ;compare w/table
FAC8 3E10                   MVI     A,20Q           ;error flag
FACA D8                     RC                      ;error return
FACB 78                     MOV     A,B             ;initialize 1791
FACC 32FEFB                 STA     SECREG          ;    sector register
FACF 3E20                   MVI     A,40Q           ;128 byte sector
FAD1 210505                 LXi     h,505h
```

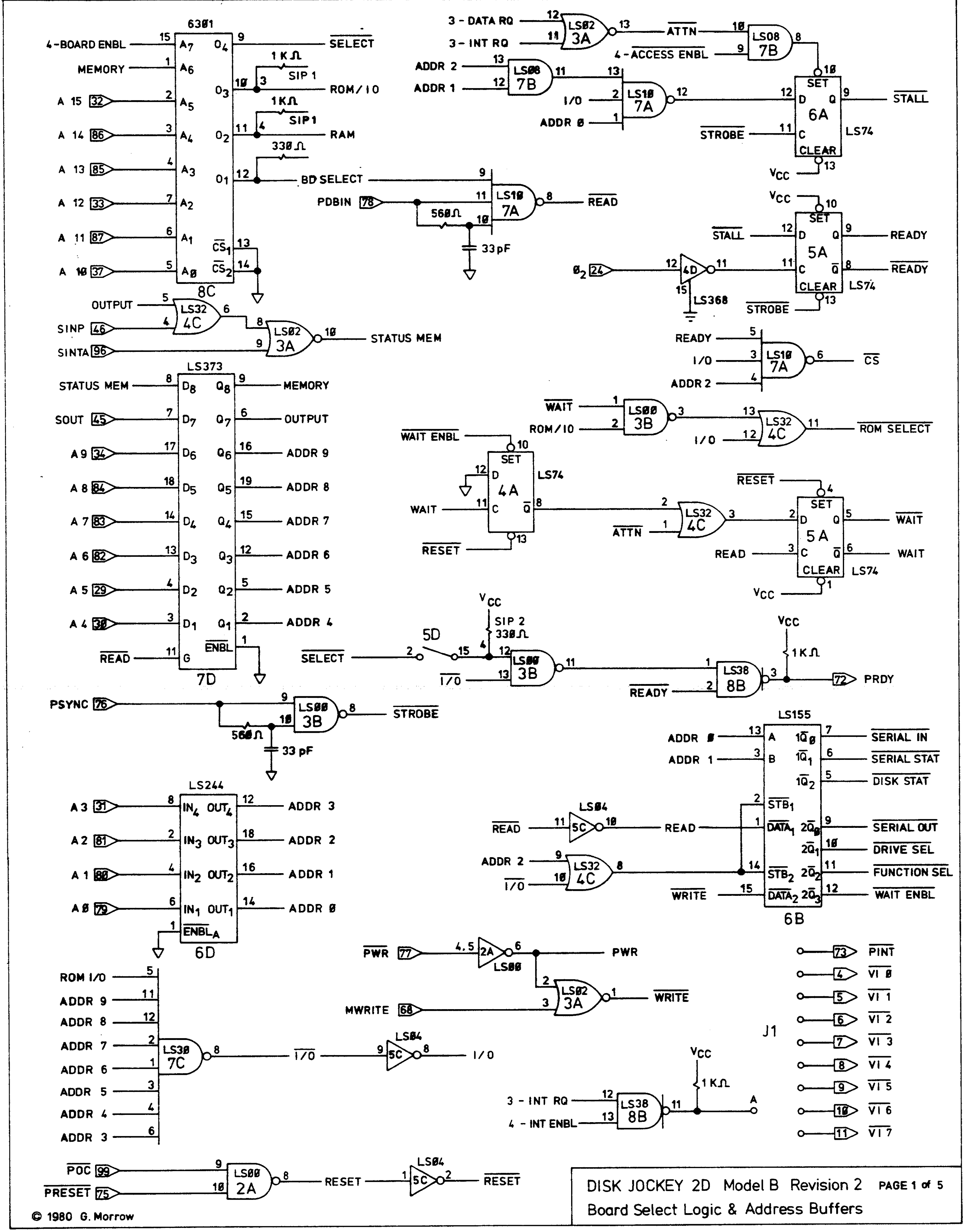
```
 FAD4 22E2FF              SHLD    ECOUNT
                  SZLOOP
 FAD7 0D                  DCR     C                 ;reduce size count
 FAD8 47                  MOV     B,A               ;sector size to B
 FAD9 F8                  RM                        ;return on minus
 FADA 17                  RAL                       ;double the count
 FADB B7                  ORA     A                 ;clear the carry
 FADC C3D7FA              JMP     SZLOOP

                  STABLE
 FADF E5                  DB      345Q              ;26 sector diskettes
 FAE0 E5                  DB      345Q              ;26 sector diskettes
 FAE1 F0                  DB      360Q              ;15 sector diskettes
 FAE2 F7                  DB      367Q              ;8 sector diskettes
                  HDLOAD
 FAE3 21EBFF              LXI     H,DISK            ;new drv ptr
 FAE6 4E                  MOV     C,M               ;save new drv in C
 FAE7 23                  INX     H                 ;current drv ptr
 FAE8 5E                  MOV     E,M               ;save old drv in E
 FAE9 71                  MOV     M,C               ;update current drv
 FAEA 23                  INX     H                 ;home cmd flag
 FAEB 7B                  MOV     A,E               ;test for
 FAEC B9                  CMP     C                 ;    drive change
 FAED 7E                  MOV     A,M               ;head load mask
 FAEE 3601                MVI     M,HEAD            ;update the mask
 FAF0 CA1BFB              JZ      HDCHK             ;no drive change?
 FAF3 23                  INX     H                 ;addr of drive table
 FAF4 E5                  PUSH    H                 ;save table addr
 FAF5 1600                MVI     D,0               ;set up the
 FAF7 42                  MOV     B,D               ;    offset address
 FAF8 19                  DAD     D                 ;calculate the
 FAF9 19                  DAD     D                 ;    parameter addr
 FAFA 3AF6FF              LDA     DCREG             ;save the
 FAFD 77                  MOV     M,A               ;density status
 FAFE 23                  INX     H                 ;track pointer
 FAFF 11FDFB              LXI     D,TRKREG          ;1791 trk reg
 FB02 1A                  LDAX    D                 ;get current track
 FB03 77                  MOV     M,A               ;save in the table
 FB04 E1                  POP     H                 ;beginning of table
 FB05 09                  DAD     B                 ;new drive
 FB06 09                  DAD     B                 ;    table pointer
 FB07 7E                  MOV     A,M               ;get density status
 FB08 32F6FF              STA     DCREG             ;update DCREG
 FB0B 23                  INX     H                 ;get the old
 FB0C 7E                  MOV     A,M               ;    track number
 FB0D 12                  STAX    D                 ;    and update 1791
 FB0E 3E7F                MVI     A,177Q            ;drive select bits
                  DSROT
 FB10 07                  RLC                       ;rotate to
 FB11 0D                  DCR     C                 ;    select the
 FB12 F210FB              JP      DSROT             ;    proper drive
 FB15 E67F                ANI     177Q              ;set the run bit
 FB17 32EAFF              STA     DRVSEL            ;save in drv reg
 FB1A AF                  XRA     A                 ;force a head load
                  HDCHK
 FB1B 21FAFB              LXI     H,DSTAT           ;test for
 FB1E A6                  ANA     M                 ;    head loaded
 FB1F 32E9FF             STA     HDFLAG          ;save the head
 FB22 F5                 PUSH    PSW             ;    loaded status
 FB23 3AEAFF             LDA     DRVSEL          ;get current drive
 FB26 4F                 MOV     C,A             ;save
 FB27 3AF7FF             LDA     SIDE            ;get current side
 FB2A 2F                 CMA                     ;    and merge
 FB2B A1                 ANA     C               ;    with drive select
 FB2C 32F9FB             STA     DREG            ;select drive & side
 FB2F EE40               XRI     ACCESS          ;toggle access bit
 FB31 4F                 MOV     C,A             ;save for PREP routine
 FB32 3AF6FF             LDA     DCREG           ;den & head cntl bits
 FB35 47                 MOV     B,A             ;save
 FB36 3AF9FF             LDA     TRACK           ;get the new track
 FB39 D601               SUI     1               ;force single
 FB3B 9F                 SBB     A               ;    density
 FB3C 3D                 DCR     A               ;    if track = 0
 FB3D 2F                 CMA                     ;compliment
 FB3E B0                 ORA     B               ;merge w/control bits
 FB3F 77                 MOV     M,A             ;load head & set density
 FB40 F1                 POP     PSW             ;head load status
 FB41 C24FFB             JNZ     RDYCHK          ;conditionally
 FB44 E5                 PUSH    H               ;    wait for head
 FB45 2AE4FF             LHLD    TIMER           ;    load time out
                 TLOOP
 FB48 2B                 DCX     H               ;count down
 FB49 7C                 MOV     A,H             ;    40 ms for
 FB4A B5                 ORA     L               ;    head load
 FB4B C248FB             JNZ     TLOOP           ;    time out
 FB4E E1                 POP     H
                 RDYCHK
 FB4F 7E                 MOV     A,M             ;test for
 FB50 E680               ANI     READY           ;    drive ready
 FB52 C0                 RNZ
                 UNLOAD
 FB53 3AF6FF             LDA     DCREG           ;force a
 FB56 F606               ORI     ULOAD           ;    head
 FB58 77                 MOV     M,A             ;    unload
 FB59 3E80               MVI     A,READY         ;set drive
 FB5B 37                 STC                     ;    not ready
 FB5C C9                 RET                     ;    error flag

                 COMAND
 FB5D 2AE4FF             LHLD    TIMER           ;get index count
 FB60 29                 DAD     H               ;    and multiply
 FB61 29                 DAD     H               ;    by four
                 CENTRY
 FB62 EB                 XCHG                    ;save in D-E pair
 FB63 21FCFB             LXI     H,CSTAT         ;issue command
 FB66 77                 MOV     M,A             ;    to the 1791
                 NBUSY
 FB67 7E                 MOV     A,M             ;wait
 FB68 1F                 RAR                     ;    for the
 FB69 D267FB             JNC     NBUSY           ;    busy flag
                 BUSY
 FB6C 7E                 MOV     A,M             ;test for
 FB6D 1F                 RAR                     ;    device busy
 FB6E 7E                 MOV     A,M             ;restore status
```

```
FB6F D0                RNC                     ;return if not busy
FB70 C376FB            JMP     PATCH+3         ;jump around patch
               PATCH
FB73 C3E3FA            JMP     HDLOAD          ;patch for old ATE

FB76 1B                DCX     D               ;test for
FB77 7A                MOV     A,D             ;    two disk
FB78 B3                ORA     E               ;    revolutions
FB79 C26CFB            JNZ     BUSY            ;47 machine cycles
FB7C 5E                MOV     E,M             ;get error code
FB7D E5                PUSH    H               ;save cmd address
FB7E 23                INX     H               ;track register
FB7F 56                MOV     D,M             ;save present track
FB80 3AEAFF            LDA     DRVSEL          ;control bits
FB83 EE80              XRI     RSTBIT          ;reset the 1791
FB85 32F9FB            STA     DREG            ;    controller to
FB88 EEC0              XRI     STBITS          ;    clear the
FB8A E3                XTHL                    ;    command busy
FB8B 32F9FB            STA     DREG            ;    fault condition
FB8E 36D0              MVI     M,CLRCMD        ;force interrupt
FB90 E3                XTHL                    ;restore the
FB91 72                MOV     M,D             ;    the track reg
FB92 E1                POP     H               ;restore the stack
FB93 7B                MOV     A,E             ;error code to A
FB94 37                STC                     ;    error flag
FB95 C9                RET

               MEASUR
FB96 110000            LXI     D,0             ;initialize count
FB99 21FAFB            LXI     H,DSTAT         ;status port
FB9C 0E10              MVI     C,INDEX         ;index bit flag
               INDXLO
FB9E 7E                MOV     A,M             ;wait for
FB9F A1                ANA     C               ;    index
FBA0 CA9EFB            JZ      INDXLO          ;    pulse high
               INDXHI
FBA3 7E                MOV     A,M             ;wait for
FBA4 A1                ANA     C               ;    index
FBA5 C2A3FB            JNZ     INDXHI          ;    pulse low
               INDXCT
FBA8 13                INX     D               ;advance count
FBA9 E3                XTHL                    ;four dummy
FBAA E3                XTHL                    ;    instructions
FBAB E3                XTHL                    ;    to lengthen
FBAC E3                XTHL                    ;    the delay
FBAD 7E                MOV     A,M             ;wait for
FBAE A1                ANA     C               ;    the index
FBAF CAA8FB            JZ      INDXCT          ;    to go high
FBB2 C9                RET                     ;98 machine cycles
               DENFIX
FBB3 79                MOV     A,C             ;trim the
FBB4 E601              ANI     1               ;    excess bits
FBB6 2F                CMA                     ;compliment and
FBB7 47                MOV     B,A             ;    save in B
FBB8 21EBFF            LXI     H,DISK          ;new disk ptr
FBBB 5E                MOV     E,M             ;get disk no.
FBBC 1600              MVI     D,0             ;offset addr
```

```
FBBE 23                INX     H               ;current disk ptr
FBBF 7E                MOV     A,M             ;move to ACC
FBC0 AB                XRA     E               ;cmpr old w/new
FBC1 F5                PUSH    PSW             ;save status
FBC2 23                INX     H               ;disk table
FBC3 23                INX     H               ;    address
FBC4 19                DAD     D               ;add the
FBC5 19                DAD     D               ;    offset
FBC6 7E                MOV     A,M             ;get parameters
FBC7 F601              ORI     1               ;mask off density
FBC9 A0                ANA     B               ;set new density
FBCA 77                MOV     M,A             ;update parameters
FBCB F1                POP     PSW             ;test new=old?
FBCC C0                RNZ
FBCD 7E                MOV     A,M             ;updata CDISK
FBCE 32F6FF            STA     DCREG           ;    also
FBD1 C9                RET

               TIMOUT
FBD2 210000            LXI     H,0             ;time-out delay
               TILOOP
FBD5 2B                DCX     H               ;decrement count
FBD6 7C                MOV     A,H             ;test for delay
FBD7 B5                ORA     L               ;    count equal zero
FBD8 E3                XTHL                    ;long NOP
FBD9 E3                XTHL                    ;    instruction
FBDA C2D5FB            JNZ     TILOOP
FBDD C9                RET

FBDE E5        SBEGIN  PUSH    H
FBDF 21E2FB            LXI     H,DSTALL
FBE2 E9        DSTALL  PCHL
FBE3 E1                POP     H
FBE4 C9                RET

               SIDEFX
FBE5 79                MOV     A,C             ;get the side bit
FBE6 E601              ANI     1               ;trim the excess
FBE8 17                RAL                     ;move the bit
FBE9 17                RAL                     ;    to the side
FBEA 17                RAL                     ;    select bit
FBEB 17                RAL                     ;    position
FBEC 32F7FF            STA     SIDE            ;save side bit
FBEF C9                RET

               PWRJMP
FBF0 00                NOP                     ;power-on
FBF1 00                NOP                     ;    jump
FBF2 00                NOP                     ;    sequence
FBF3 00                NOP                     ;    with NOP
FBF4 00                NOP                     ;    padding
FBF5 C300F8            JMP     DBOOT

FBF8                   DS      10Q             ;I/O locations

FFC9                   org     ram+3c9h
```

```
FFC9           STACK   DS      31Q

FFE2 0000      ECOUNT  DW      0
FFE4 0018      TIMER   DW      1800h          ;head load time out
FFE6 00FF      DMAADR  DW      RAM+300H       ;dma address
FFE8 08        DSFLAG  DB      10Q
FFE9 00        HDFLAG  DB      0              ;read header flag
FFEA 7E        DRVSEL  DB      176Q           ;drive select constant
FFEB 00        DISK    DB      0              ;new drive
FFEC 08        CDISK   DB      10Q            ;current disk
FFED 00        TZFLAG  DB      0              ;home cmd indicator
FFEE 09        D0PRAM  DB      11Q            ;drive 0 parameters
FFEF FF        D0TRK   DB      377Q           ;drive 0 track no
FFF0 09        D1PRAM  DB      11Q            ;drive 1 parameters
FFF1 FF        D1TRK   DB      377Q           ;drive 1 track no
FFF2 09        D2PRAM  DB      11Q            ;drive 2 parameters
FFF3 FF        D2TRK   DB      377Q           ;drive 2 track no
FFF4 09        D3PRAM  DB      11Q            ;drive 3 parameters
FFF5 FF        D3TRK   DB      377Q           ;drive 3 track no
FFF6 09        DCREG   DB      11Q            ;current parameters
FFF7 00        SIDE    DB      0              ;new side
FFF8 01        SECTOR  DB      1              ;new sector
FFF9 00        TRACK   DB      0              ;new track
FFFA 00        TRKNO   DB      0              ;disk
FFFB 00        SIDENO  DB      0              ;    sector
FFFC 00        SECTNO  DB      0              ;    header
FFFD 00        SECLEN  DB      0              ;    data
FFFE 00        CRCLO   DB      0              ;    buffer
FFFF 00        CRCHI   DB      0
```

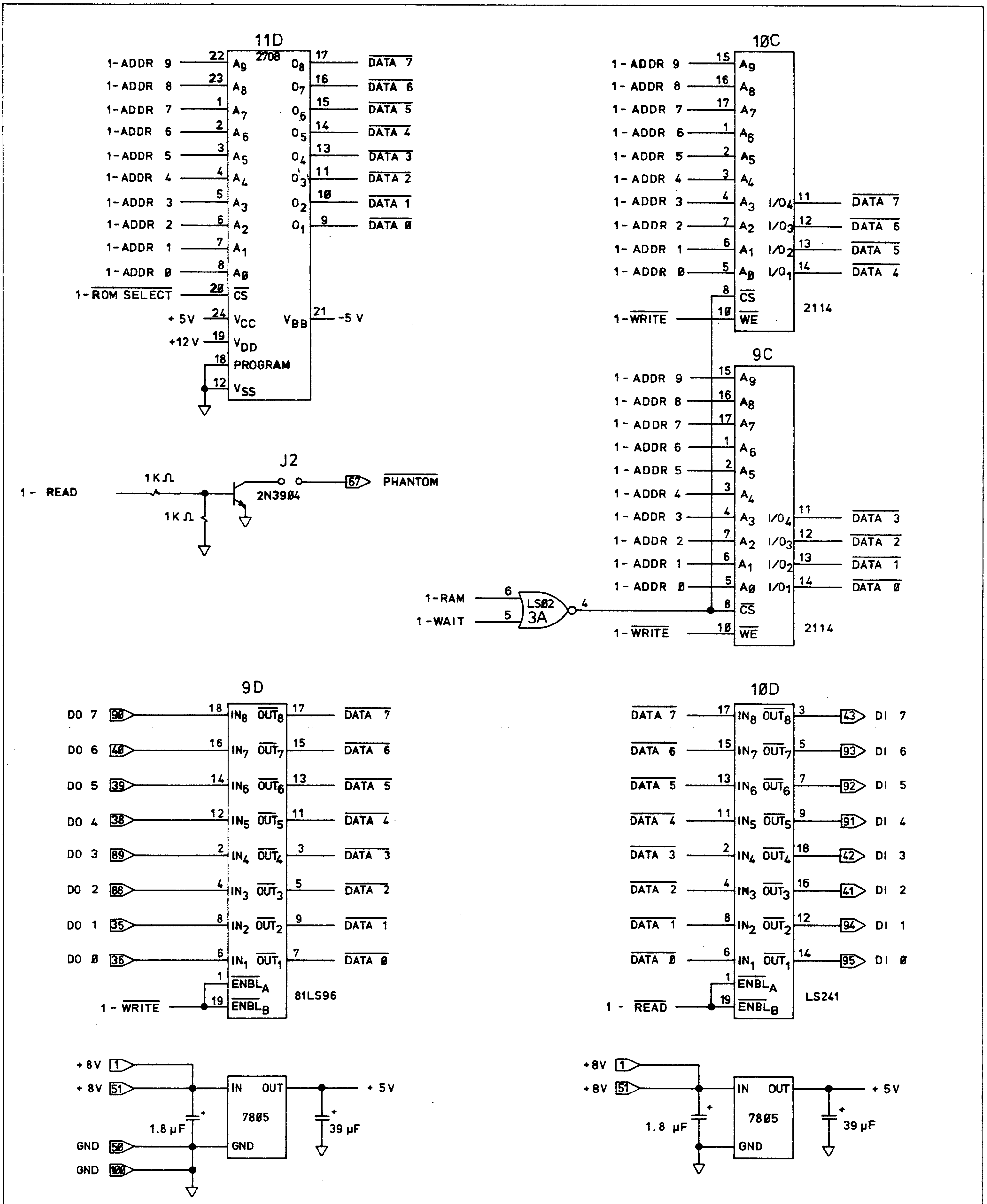
# SCHEMATICS



DISK JOCKEY 2D Model B Revision 2 PAGE 1 of 5

Board Select Logic & Address Buffers

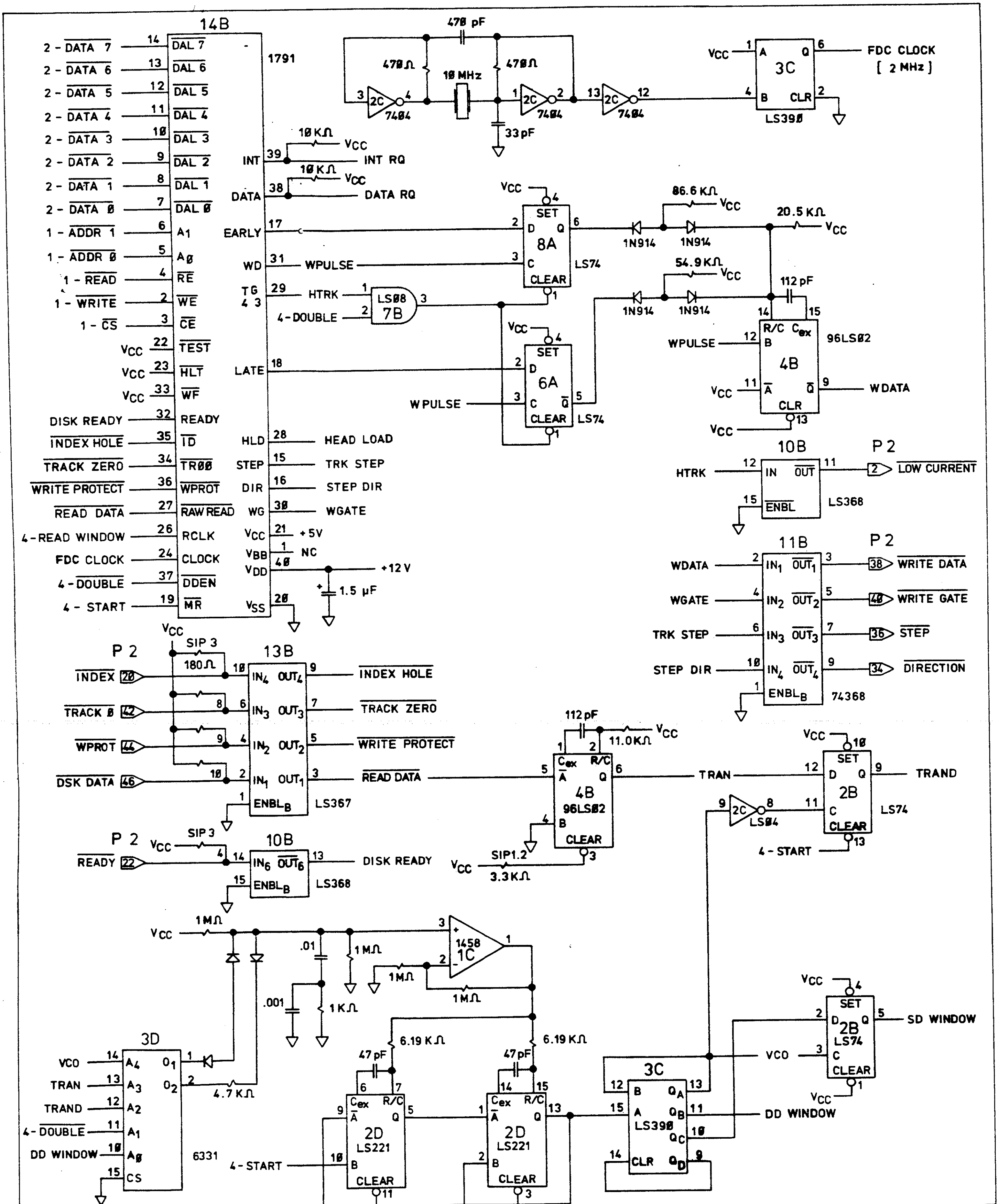
© 1980 G. Morrow



DISK JOCKEY 2D Model B Revision 2 PAGE 2 of 5

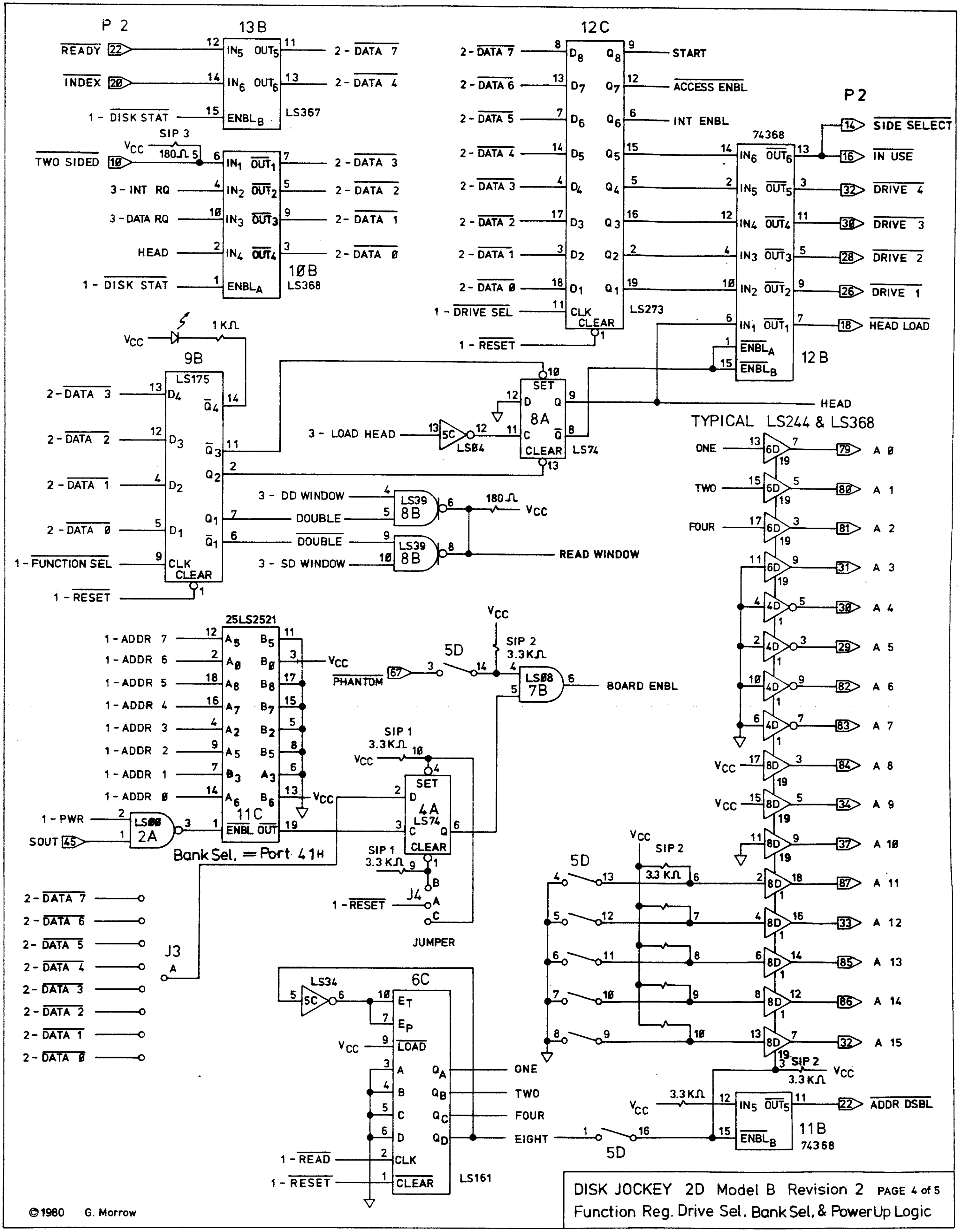
RAM, ROM, & Data Buffers

©1980 G. Morrow



DISK JOCKEY 2D Model B Revision 2 PAGE 3 of 5

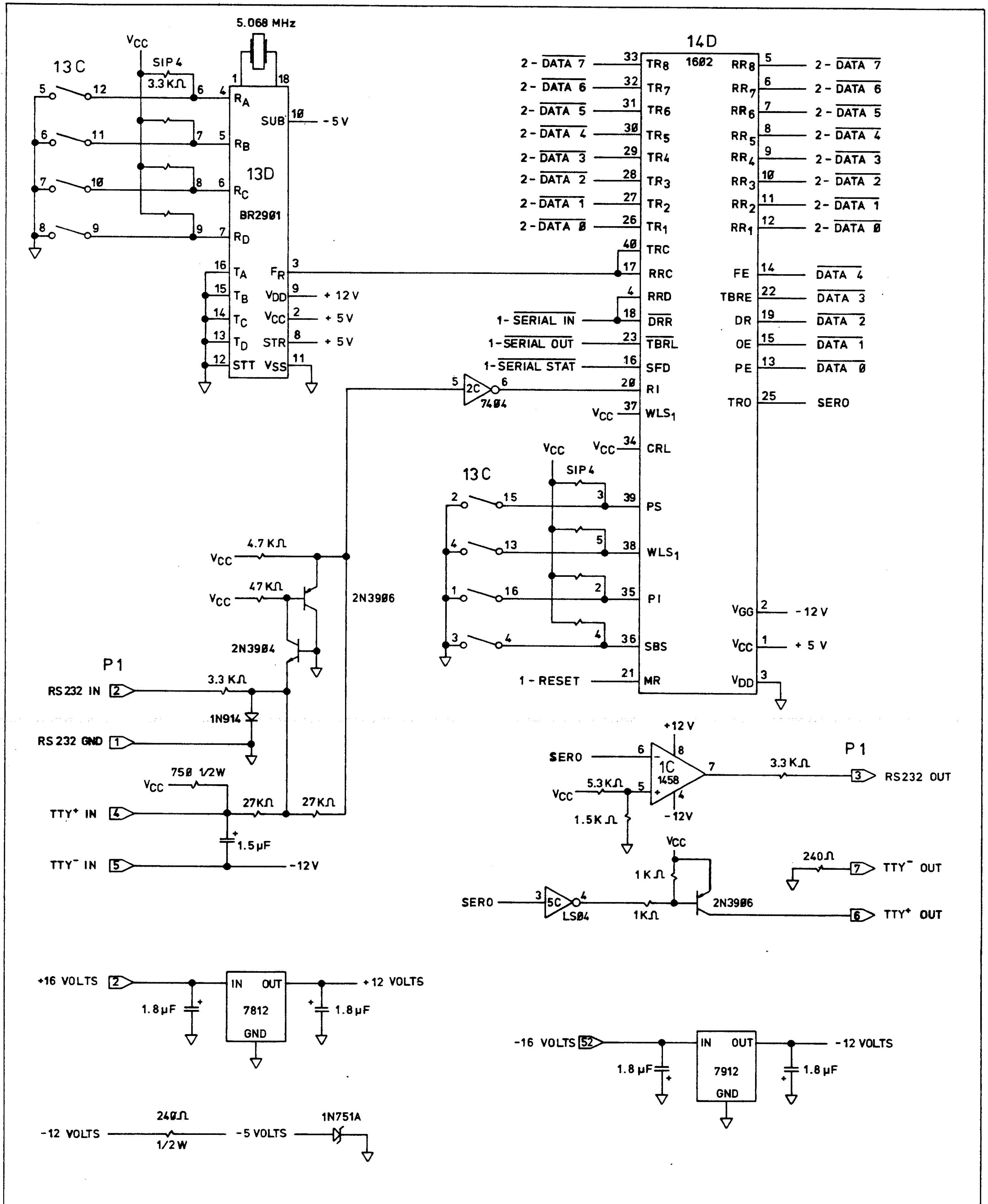
Write Data & Separation Logic, FDC Controller



DISK JOCKEY 2D Model B Revision 2 PAGE 4 of 5

Function Reg. Drive Sel. Bank Sel. & PowerUp Logic

©1980 G. Morrow



©1980 G. Morrow

DISK JOCKEY 2D Model B Revision 2 PAGE 5 of 5
Serial Interface Logic

## PARTS LIST

| | | | |
|---|---|---|---|
| [ ] | 1 | 5" x 10" printed circuit board w/solder mask & legend | |
| [ ] | 1 | 3.3 Ohm 1/4 watt 5% resistor | orange-orange-gold |
| [ ] | 1 | 180 Ohm 1/4 watt 5% resistor | brown-grey-brown |
| [ ] | 1 | 240 Ohm 1/4 watt 5% resistor | red-yellow-brown |
| [ ] | 2 | 470 Ohm 1/4 watt 5% resistors | yellow-purple-brown |
| [ ] | 1 | 750 Ohm 1/2 watt 5% resistor | purple-green-brown |
| [ ] | 9 | 1k Ohm 1/4 watt 5% resistors<br>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. | brown-black-red |
| [ ] | 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 | 10k Ohm 1/4 watt 5% resistors | brown-black-orange |
| [ ] | 1 | 14.0k Ohm 1/4 watt 1% resistor | brown-yellow-black-red |
| [ ] | 1 | 16.2k Ohm 1/4 watt 1% resistor | brown-blue-red-red |
| [ ] | 2 | 27k Ohm 1/4 watt 5% resistors | red-purple-orange |
| [ ] | 1 | 47k Ohm 1/4 watt 5% resistor | yellow-purple-orange |
| [ ] | 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 capacitors | |
| [ ] | 1 | 47 picofarad 2% silver mica capacitor | |
| [ ] | 1 | 112 picofarad 2% silver mica capacitor | |

## PARTS LIST

[ ] 1 470 picofarad 5% silver mica capacitor

[ ] 1 .001 microfarad ceramic disk capacitor

[ ] 1 .01 microfarad mylar capacitor

[ ] 4 1.0 - 2.0 microfarad dipped tantalum capacitor

[ ] 6 1.0 - 4.7 microfarad axial lead tantalum capacitors

[ ] 2 39 microfarad axial lead tantalum capacitors

[ ]16 ceramic disk capacitors - 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 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

[ ] 4 1N914/4820-0201 signal diodes

> NOTE: The silk screened legend on the circuit board shows a group of four diodes just above the 1791 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

[ ] 1 2N3904 transistor

[ ] 2 2N3906 transistor

[ ] 1 8 pin low-profile socket

[ ]16 14 pin low-profile sockets

[ ]13 16 pin low-profile sockets

# PARTS LIST

| | | | |
|---|---|---|---|
| [ ] | 3 | 18 pin low-profile sockets | |
| [ ] | 7 | 20 pin low-profile sockets | |
| [ ] | 1 | 24 pin low-profile socket | |
| [ ] | 2 | 40 pin low-profile sockets | |
| [ ] | 3 | 74LS00 quad 2-input NAND gate | 2A,3B,5B |
| [ ] | 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-inpur 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 | 74LS123 dual monostable | 2D |
| [ ] | 1 | 74LS155 dual 1 of 4 decoder | 6B |
| [ ] | 1 | 74160/LS160/74161/LS161 4 bit counter | 6C |
| [ ] | 1 | 74165/74LS165 8 bit load shift register | 4B |
| [ ] | 1 | 74175/LS175 4 bit dual rail register | 9B |
| [ ] | 1 | 74LS240 octal tri-state inverting buffer | 10D |
| [ ] | 2 | 74LS244 octal tri-state buffer | 6D,8D |
| [ ] | 1 | 74273/LS273 octal latch | 12C |
| [ ] | 1 | 74365/LS365/74367/LS367 hex tri-state buffer | 4D |
| [ ] | 1 | 74367/LS367 hex tri-state buffer | 13B |
| [ ] | 2 | 74368 hex tri-state inverting buffer | 11B,12B |
| [ ] | 1 | 74368/LS368 tri-state inverting buffer | 10B |
| [ ] | 1 | 74LS373 octal tri-state buffer/latch | 7D |

# PARTS LIST

| | | | |
|---|---|---|---|
| [ ] | 1 | 74390/LS390 dual decade counter | 3C |
| [ ] | 1 | 81LS96/LS98 octal tri-state inverting buffer | 9D |
| [ ] | 1 | 25LS2521 octal comparator | 11C |
| [ ] | 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 UART | 14D |
| [ ] | 1 | FD1791 dual density floppy disk controller | 14B |
| [ ] | 1 | 1448/4558 dual operational amplifier | 1C |
| [ ] | 2 | 7805 monolithic 5 volt 1 amp regulators | |
| [ ] | 1 | 7812/78M12 monolithic 12 volt .5 amp regulator | |
| [ ] | 1 | 79L05 monolithic -5 volt 100 ma regulator | |
| [ ] | 1 | 7912/79M12 monolithic -12 volt regulator | |

# SUBJECT INDEX

## SUBJECT INDEX

# SUBJECT INDEX

## SUBJECT INDEX