Disk Jockey 2/DIO

User's Manual

MORROW DESIGNS

* *

Much care has been taken in preparing this manual. Morrow Designs believes that the information presented is accurate and reliable, but takes no responsibility, financial or otherwise, for any consequences arising out of this material. comments regarding this manual may be directed to the Documentation Department at Morrow Designs, 5221 Central Avenue, Richmond, California 94804.

* *

TABLE OF CONTENTS

L I S T O F F I G U R E S

L I S T O F T A B L E S

User's Manual

DISK JOCKEY 2/DIO tm

1. INTRODUCTION

The Thinker Toys DISK JOCKEY 2/D IO (DJ IO) board features five distinct subsections:

1. A floppy disk controller, capable of reading and writing data in either single density FM (Frequency Modulation code or double density MFM (Modified Frequency Modulation) code with write precompensation, which can be connected to any floppy disk drive plug compatible with the Shugart 800/850.

2. A baud rate selectable hardware UART (Universal Asynchronous Receiver/Transmitter) that allows communication with a terminal device at TTY 20ma current loop or RS-232 levels.

3. Automatic address generation upon reset or power-up which allows a "jump start" to the boot strap program in the EPROM contained on the board.

4. Bank select logic which allows the on-board memory 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.

5. Extended addressing capabilities, which allow on-board memory to be addressed anywhere within the 16 megabyte address space of the IEEE 696 S-100 specification.

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

INTRODUCTION

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

Most users will use the operating system sold with the DJ IO. Custom software interfaces do not need to deal directly with the hardware level registers. Instead, they can call standard disk and serial I/O subroutines contained in 2048 bytes of EPROM memory on the DJ board. This EPROM occupies a 2048 byte block of S-100 bus memory address space during booting. Both the EPROM and 1024 bytes of RAM are available by bank selecting. Most of the functions of the EPROM are duplicated in the memory address space as part of the operating system.

The Disk Jockey IO uses eight port addresses, which are switch selectable. When the boot on the Disk Jockey IO is used, the board will be bank selected at the beginning of the boot, and turned off before the end of the boot. A provision is made to turn off any conflicting memory, by using the PHANTOM* line. The DJ IO requires 3k of address space, starting at F000H, to FBFFH.

The actual addresses where the EPROM and RAM appear are controlled by another PROM, referred to as the address selection PROM. The PROM is supplied with standard addresses burned into it for the 2k EPROM and the 1k Ram. If the standard addresses would conflict with some other device on the system bus, a PROM burned with non-standard addresses can be substituted. If extended addressing is used, the 64K block where the on-board memory resides is switch selectable without changing the PROM.

2. DISK JOCKEY INSTALLATION GUIDE

This section of the manual is intended to provide all the information needed to install the DJ IO in your computer. This includes switch settings, memory and bootstrap configuration of your computer, and cable connections. Exactly what is required to use the DJ IO as the disk controller in your computer is described. Other uses, such as interfacing two disk controllers in the same computer, are described elsewhere.

In the back of this manual, there is a Reference Guide, which briefly describes all of the switches, and provides normal settings. These "normal" settings provide a base from which to work, but may require some changes for your computer. The basis for these changes is explained below.

2.1. PREPARING YOUR COMPUTER

To use the DJ IO, you must have an S-100 bus, a CPU capable of executing 8080 instructions and at least 24K of memory starting at 0. The Disk Jockey IO and its software are designed to handle bootstrapping, your 8" disk drives and your console. To properly set up your computer, conflicts between what the DJ will do and what the rest of your computer does must be settled. These conflicts fall into four categories: memory, bootstrap, IO port locations, and console IO.

2.1.1. MEMORY

If your computer has less than 60K of memory starting at the lowest possible location, then no conflict between the DJ IO and your computer exists, and you can skip this section. More likely, you are using the DJ IO to take advantage of all 64K of address space. Please read this section to remedy the conflict between your computer's upper 4K of memory and the EPROM and RAM on the DJ IO. If you wish to use extended addressing, please refer to the Hardware Features Guide later in this manual.

The Disk Jockey IO will usually not take up memory space inside your system. However, during a DJ IO bootstrap, a section of memory on the board will be turned on. This only occurs during the first part of bootstrap, while low memory is loaded with the routines that bring in the rest of CPM. The DJ IO memory is then turned off, allowing loading of high memory. The DJ IO memory is in the highest 4k of memory, between F000H and FBFFH. There can only be one section of memory located in the same space. The answer to this problem is to make use of the PHANTOM* signal or not use memory that overlaps this space.

The PHANTOM* signal is a command that is sent to all boards plugged into a S-100 bus. When it is on, memory boards that have been set to watch for this command turn off. On some memory boards, this can be done by setting a switch. On others, a wire must be soldered into place. How to use the PHANTOM* should be described in your memory's manual. If you have a 64K memory

board, it is alright to allow the PHANTOM* command to turn it all off and on.

If your memory does not use the PHANTOM* command, switch off the portion of the board that covers the last 4K. It may also be possible to use bank addressing of the conflicting memory board.

2.1.2. BOOTSTRAP

A bootstrap is a set of instructions that control what a computer does when it is turned on or reset. If anything happens after you turn on your computer without the DJ IO board, (besides the power light coming on), your computer already has a bootstrap. If your computer is not set up to bootstrap, you can skip this section, and use the normal switch settings.

If your system has a bootstrap, you have two choices: turn off the system's bootstrap and use the bootstrap on the DJ IO; or change the system's bootstrap so that it starts the DJ bootstrap. To do either of these, you will need to know how your system is creating its bootstrap.

The bootstrap on the DJ IO can be turned off by turning off the first switch on the Power on Jump Switch Bank at 9C. Then, set-up your system to begin execution at F000H. This will cause the Disk Jockey to load memory with the operating system on the disk. Ifyou turn off your system's bootstrap, the switch settings in the Reference Guide should work for you.

2.1.3. IO PORT LOCATIONS

The Disk Jockey IO uses eight IO ports. These are located between 70H and 77H. If your computer is not using any of these ports, you can skip this section.

The IO locations for the DJ IO were picked to minimize conflicts with other popular devices. Examples of devices that use port addresses are: multiple IO port boards or the bank select feature of memory boards. If there are other devices using the same IO ports, then the port address must be changed. Changing port addresses also involves changing the software that uses these ports. In the case of the DJ IO, this would involve changing the bootstrap EPROM and the CBIOS. It is strongly suggested that the other device using the same port be changed.

2.1.4. CONSOLE IO

The Disk Jockey software expects to use its own serial port. In other words, communicating with your computer will happen through the DJ IO board, and you must have your console connected to it. It is possible to use other serial ports for console IO, but this can't be done until you have a system capable of altering CPM. If you do not have a cable for the serial port on the DJ, please read the section on cable connections.

Most terminals support a wide variety of character transmission formats and transmission speeds. For best performance, the highest possible speed the terminal can use should be selected, and the baud rate switches on the DJ IO set to the same rate. These switches are on the BAUD rate switch bank, and a list of possible rates is part of the Reference Guide.

The character format is also variable, but the main criteria is that both the terminal and the DJ IO are set up the same way. The "Normal Use" settings suggested are 8 bit characters, with no parity bit and two stop bits. Other permissable settings are described in the Reference Guide in the back of this book.

2.2. PHYSICAL INSTALLATION

Once you have compared the switch settings on the DJ IO board with the suggested settings in the Reference Guide (last pages of this manual), and read the preceding section on preparing your computer, you are ready to plug in the board.

First, turn OFF the power to your computer. It is important to never plug in or remove any board from your computer with the power on. Then, remove the cover over the S-100 bus. The S-100 bus is series of 6 inch wide connectors, with a rack on either side of it that supports printed circuit boards. The connector is offset to the left, so that boards cannot be put in backwards.

Pick an empty an slot near the back of the S-100 bus. Unless your computer has a special bus, any slot will work. Then slide the DJ IO board, with the chips and connectors facing the front of the computer, into the slot. Press it in until it is firmly seated. The top of the board should be level with the other boards in the rack. Do not force the board. If it does not go in easily, check for obstructions along the S-100 bus. Or, sometimes the board will need to be flexed slightly by pressing gently in the middle near the bottom of the board. This is because the printed circuit board is somewhat flexible. Next, you should connect the cables.

2.2.1. CABLE CONNECTIONS

There are two cables that must be connected to use the DJ IO: the drive cable and the console cable. The drive cable is normally supplied with the drive. It is a ribbon cable about 3 inches wide. It is connected to the socket in the upper right of the board labeled P1. When it is connected, it should hang over the back (the side without parts) of the board. The other end should be connected to the drive cabinet as follows. The last connector (or only connector) should be plugged into Drive A. If there are other drives, they can be connected to the middle connectors.

Each connection to a drive should be oriented so that the edge of the cable that came from the left side of the DJ IO board is UP on the drive end. When the cables are incorrectly oriented, the

red LED on the front of the drive will come on and stay lit when the power is turned on. To correct this, unplug the connector and plug it back in the other side up at the drive end of the cable.

There are two types of console connections supported by the DJ IO board: RS232, the most common, and TTY, also called current loop. A short description of an RS232 cable follows.

Most terminals use RS232 cables for communication. Although there are usually many wires in these cables, only three are necessary: ground, input (receive) and output (transmit). To construct a cable, first solder three wires to three of the connector lugs in the package supplied with your DJ IO. Then slip the lugs into the plastic piece so that they snap into place in three slots together at the edge of the connector. There is a drawing of this connector attached to the board in the back of this manual.

The wire coming out of the plug at the edge is the ground wire, and should be connected to pin 7 of an RS232 connector or cable. If you are building a cable to directly connect to your terminal, use a male DB 25. If your are installing the connector in the back of your computer, use a female DB 25 connector. The input wire gets connected to pin 2 of the R232. The output wire is connected to pin 3 of the RS232. For a diagram of pins on the DJ IO board, see the section on IO connectors. The plastic plug should be connected to socket P2 on the DJ IO board, with the ground wire connected to the leftmost pin on the board.

2.2.2. BOOTING UP THE FIRST TIME

Now that you have set the switches, plugged in the board and connected the cables, you are ready to power up. Before putting the system diskette in the drive, turn on the power to the console, the drive(s) and the computer. If everything is okay, the LED on the front of drive A should begin blinking. Hold the diskette so that the label is up and to the right. Insert the system diskette into drive A and close the door when the LED blinks off. The LED will turn on, drive A will make noises for about 4 seconds, and a message and a prompt (A>) will appear on your console. If this didn't happen, read the diagnostic guide.

Once you have successfully booted your Disk Jockey IO, you will first want to make a backup copy of the system diskette. Please read the instructions in the beginning of the Programming Guide. The system diskette provided must remain WRITE PROTECTED, that is, never cover the slot in the edge of the diskette. Once you have made a copy of the system diskette, put it away in a safe place, where it will not be folded, crimped, heated or dampened. Use the copy you have made, and follow the instructions for using CPM.

2.3. DIAGNOSTIC GUIDE

This guide is written from the standpoint that the hardware is okay, but the setup is wrong. It will help you configure a new system, but cannot diagnose for Read/Write Errors.

When the computer is first turned on or reset, and there is no diskette in the drive, the LED on drive A should begin flashing. If the light stays on continuously, then the connector to the drive should be reversed. If it never came on at all, check the LED on top of the DJ IO board. If it comes on several seconds after a power up or reset, but the drive LED is not on, check the power to the drive and the drive cable.

If the drive LED and the LED on the DJ IO never come on, there could be several problems. First, there could be a faulty boot. The bootstrap will fail if there are more than one devices trying to bootstrap at the same time or the bootstrap is set to begin at the wrong location in memory. Please check the Bootstrap section above and the settings on the Power on Jump Switch Bank. Remember, if you remove a board from the S-100 bus, turn off the power first!

Secondly, there could be a memory conflict. If the bootstrap on the DJ IO is being used and there is memory located in the same place, the information from the bootstrap will be mixed with the information in memory. Read over the section on Memory.

Now the drive light is flashing. You should put in a diskette, holding the label up and in your hand, until it clicks into place. Close the door when the drive LED turns off. The drive should make noises for a period of not more than 5 seconds. If it starts, but stops right away, check to see if your system has memory at 0. If it continues making noises for longer than 5 seconds, it is unable to read the diskette. Make sure your are using the DJ IO system diskette, and that it is oriented correctly. Correct orientation is: the label is up and to your right, and the long narrow slot, with shiny brown disk material visible in it, is inserted first. The door is closed by sliding a plastic piece into place that covers the end of the diskette, and prevents its removal.

If drive A stopped after 5 seconds, but no prompt appeared, there could be a problem in the connection and set up of your console. Make sure your console is on (a cursor should be visible), it is on-line (as opposed to local), and it is connected to the serial port connector on the DJ IO board. After you have checked these, check out the terminal configuration settings. These should agree with the settings of the BAUD rate switch bank on the DJ IO. If a jumble of letters appears on the console, the settings are still incorrect.

3. PROGRAMMING GUIDE

For most users, the operating system purchased with the DJIO board, such as CP/M, will be all that is needed to use the Disk Jockey IO. The Disk Jockey IO CP/M is tailored to the I/O of the Disk Jockey IO controller. It expects that a serial TTY/RS-232 terminal is connected to P2 (serial port) of the Disk Jockey. CP/M is supplied on a write protected diskette (notch open) which should be kept that way. DO NOT COVER THE NOTCH ON THE DISKETTE. Finally, the system disk is designed to self load when the disk is placed in drive A and a branch is made to F000H. Once the Disk Jockey IO has been bootstrapped, it is normally banked switched out of memory.

3.1. BACKING THE SYSTEM DISKETTE

To make a copy of your system diskette, you will first need to format a diskette. Use the FMTIO# program provided on your system diskette by typing:

A>FMTIO# (carriage return)

The format program will ask which drive the blank diskette is in, and what size sectors to format in. The DJ IO software allows formatting in 1024 byte sectors which provide the most storage per diskette and is the suggested size.

The next step is to run the SYSGEN program. SYSGEN will ask which drive to read the system from. Specify drive A, which should still have the system diskette in it. SYSGEN next asks for the destination diskette. Type B for drive B, which should still have the formatted diskette in it. When SYSGEN is finished, it will have copied the system tracks from the original to your backup copy.

To copy the rest of the programs to your backup diskette, use PIP to transfer from the system diskette to your backup. The command

A>PIP B:=*.*[V] (carriage return)

will copy and verify all files. At this point, you have a complete copy of the system diskette. The original should be put away in a safe place. You can follow the same procedure to make a further copy of the backup diskette before making further modifications. For example, you will probably want to use MOVCPM to increase the size of your CP/M system. You should use the command "MOVCPM * *" followed by "SYSGEN" to store a CP/M configured to your systems memory size. Please read the CP/M manual for instructions on modifying CP/M.

3.1.1. SPECIAL UTILITIES

There are several programs supplied with the Disk Jockey IO that are not part of a normal CP/M. The first of these, FMTIO#, has already been mentioned. Formatting is dependent on the hardware used, so it must be supplied by the system manufacturer. An ASM file for FMTIO# is also on the system diskette as an example.

The program REGENIO is a special purpose program. It is used to read diskettes that have been written in IBM 3740 format format and rewrite them. This is done by reading each track on a byte by byte basis, and rewriting it in the correct format.

The INSTALL program allows the disk CBIOS of the Disk Jockey to be patched into other CP/M operating systems. Information on the use of this program is contained in the file INSTALL.DOC.

3.2. CUSTOM INTERFACING

For those who are writing their own software to use the DJIO, there are three approaches available: to make use of the Jump Table that is part of CP/M; to use the Jump Table located in EPROM on the DJIO; to provide low-level drivers to directly interface with the disk controller. The suggested method is to use the Jump Table provided in CP/M. This requires the least amount of effort, and will be less difficult and time-consuming. It also means that software developed using the CP/M Jump Tables will be portable to other machines using the same operating system. Please refer to the documentation supplied that describes the use of the CP/M operating system. There are also several books which explain the workings of CP/M in greater (and clearer) detail.

The second and third approaches are available to those who need greater control over disk accesses. An example of this would be a data acquisition system for recording data in a time-critical environment. The Jump Table of the EPROM provides the next easiest method of approach. It bypasses the bookkeeping that operating systems provide, but still gives the user a debugged method of using the DJIO. A description of how to use the Jump Table follows. The last method is not described in detail, but by reading the section on using the EPROM, and by studying the section on the Hardware Level Registers, it is possible to write a series of low-level drivers for the DJIO. Examining the listings provided will also be of help.

3.2.1. ROM JUMP TABLE

The user should branch to the appropriate address in a jump table in the first few words of the system EPROM. Since each subroutine ends with a RET instruction, a CALL instruction should be used to branch to the subroutine. In order to read the EPROM, it must be banked selected, since a normal bootstrap turns off the EPROM. Please read the information on bank selecting in Hardware Registers.

The jump table contains jump instructions to the true address of the utility routines within the EPROM. Having a jump table allows the individual routines to be updated and moved around within the EPROM without having to change software that calls the routines. Let A represent the address of word 0 of the onboard EPROM. In boards with standard address decoding PROMS, A = F000H.

The address to call for the utility routines are then:

Table 3-1: EPROM JUMP TABLE

ADDRESS	VALUE Hex	NAME	FUNCTION
A	F000	BOOT	Bootstrap routine
A+3	F003	CIN	Serial port input
A+6	F006	COUT	Serial port output
A+9	F009	HOME	Recalibrate (seek to TRK0)
A+12	F00C	SEEK	Track select
A+15	F00F	SETSEC	Select sector
A+18	F012	SETDMA	Set DMA address
A+21	F015	READ	Read a sector of disk data
A+24	F018	WRITE	Write a sector of disk data
A+27	F01B	DRIVE	Select a disk drive
A+30	F01E	CPAN	Test for panic character
A+33	F021	CSTAT	Serial status input
A+36	F024	DMSTAT	Read current DMA address
A+39	F027	DSTAT	Read disk status
A+42	F02A	FLASH	Loop to flash LED
A+45	F02D	SETDEN	Set density
A+48	F030	SETSID	Set side for 2-headed drives

The specific function of each subroutine is described below.

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

3.2.2. SERIAL I/O

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

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

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

3.2.3. CIN

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

3.2.4. COUT

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

3.2.5. CPAN

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

3.2.6. CSTAT

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

3.3. DISK I/O

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

Information on the disk is organized into 77 concentric tracks. The disk read/write head can be moved to any track by a series of step-in or step-out commands. A step-in command moves the read/write head one track towards the center of the disk. A step-out command moves the head one track away from the center of the disk. The numbering of the tracks is arranged so that track zero is the farthest from the center of the disk. One of the responsibilities of the Western Digital 1791 controller is to know the current track number over which the read/write head is located and to calculate how many step-in or step-out commands are necessary to move the head to a new track. The controller chip only maintains the track information on the currently selected drive. This information must be changed whenever a new drive is selected or the 1791/8866 will report a seek error and home the head.

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

Table 3-2: PHYSICAL DISK FORMATS

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

In the header field which precedes the data field of a sector, the track number, the side, the sector number and the sector length are recorded. During read or write commands, this header is read before data transfers take place. Whenever a seek command is issued which causes the the read/write head to move to a new track the 1791/8866 on the DJ board performs a verify which

reads this sector header to make sure the head is positioned correctly and to determine if there is any change in the sector length or the density of the recorded information. If there is an error as to the track number, the 1791/8866 automatically issues a seek to track zero command to position the head over a known track.

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

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

Specify the drive if it has changed since the last disk access.

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

Check for error conditions.

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

Specify the side of the diskette in the case of two-sided drives.

Specify the starting memory address of block of data that is to be transferred to or form the disk.

Check for error conditions.

Actually perform the read or write operation.

Check for error conditions.

3.4. EPROM SUBROUTINES

SEEK - The value in the C register of the CPU specifies what track the read/write head will be positioned over when the next disk read or disk write operation is issued. A bounds check is made for a value greater than or equal to zero and less than or equal to 76. If the value in the C register is within these bounds, the contents of the C register is written into the RAM location TRACK. Otherwise no action is taken, the carry flag is set and the subroutine returns to the calling program.

SETSEC - The value in the C register of the CPU specifies what sector will be involved in the next disk read or write operation. If the C register contains a zero, the carry flag is set and the routine returns immediately. If the C register is non-zero, the low order five bits are transferred to the RAM location SECTOR, the carry flag is cleared and the routine returns to the calling program. Just prior to a disk transfer operation a comparison is made between the value in SECTOR and the maximum number of sectors on the track that the transfer is to take place on. If the value in SECTOR exceeds the maximum number of sectors, the transfer operation is aborted and error information is reported.

SETDMA - During disk transfer operations blocks of data are moved to and from the disk. These blocks can be 128, 256, 512, or 1024 bytes long. The starting address of a data block that will be involved in the next disk transfer operation is specified by the B-C register pair when the SETDMA subroutine is called. The contents of the B-C pair are written into the memory location specified by the label DMAADR. The carry flag is cleared and the routine ends.

DRIVE - The value of the C register determines which of 4 disk drives will be selected for the next disk transfer operation. Accordingly, the data in C is trimmed to the low order two bits and stored in the RAM location NDISK. The carry flag is cleared and the routine returns to the calling program.

SETSID - Double sided floppy disk drives have two read/write heads so that information can be stored and retrieved from both sides of the diskette. The two heads are positioned so that they are both on the same track one directly below the other. They also share common read/write electronics. Therefore only one of these heads can be selected at a time. Bit 0 of the C register is used to select which of the two heads on a double sided drive will be used during the next disk transfer operation. A zero in bit 0 will select the bottom head and a 1 will select the top head. Selecting a side and selecting a disk are independent operations. If side zero is selected then regardless of the disk selected, side zero will always be accessed until SETSID is called. Finally, if the selected disk is single sided, side zero will always be selected regardless of the results of the SETSID routine.

SETDEN - The 1791/8866 Floppy Disk Controller operates in two modes: single density FM (Frequency Modulation) mode or double density MFM (Modified Frequency Modulation) mode. Bit 0 of the C register determines what density the 1791 will operate in when the next disk transfer operation is

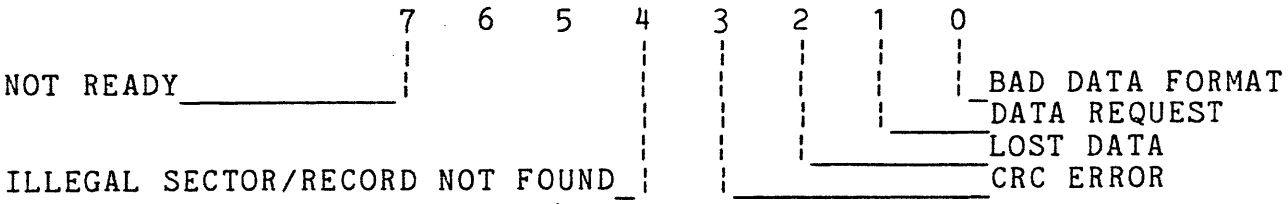
initiated (0=single,1=double). Care must be exercised in the use of this routine. Under certain conditions, if the density is changed in between disk transfers that occur on the same track, the micro-program that the 1791/8866 controller executes could fall into an error loop from which it could not recover. In such a case the system would have to be reset before further disk operations could be performed. The density mode of the 1791/8866 can safely be changed when a subsequent disk transfer operation will occur on a different track than the last. It should be noted that the firmware of the Disk Jockey has the ability to automatically set the density mode of the 1791/8866. Whenever a new drive is to be selected or whenever the head is not loaded, the Disk Jockey firmware performs a "read header" operation just after positioning the read/write head (if necessary) and just before attempting to perform a disk transfer. This "read header" operation is used to establish the density of the (possibly new) track and to determine the length of the sectors on this track. If the density has not changed from the last "read header" operation or if the calling program has set the density correctly through the use of SETDEN, the process of reading the sector header is slightly faster (by approximately one and a half diskette revolutions) than it would be if the initial assumption concerning the density was wrong.

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

READ - This subroutine transfers information from the diskette to memory. The first task is to select the proper disk drive. If the new drive is not the same as the current drive, the load head time-out flag is set and the current drive is updated to be the new drive. Next, the "head loaded" flag is tested. If the head is not loaded or if the current drive was not the same as the new

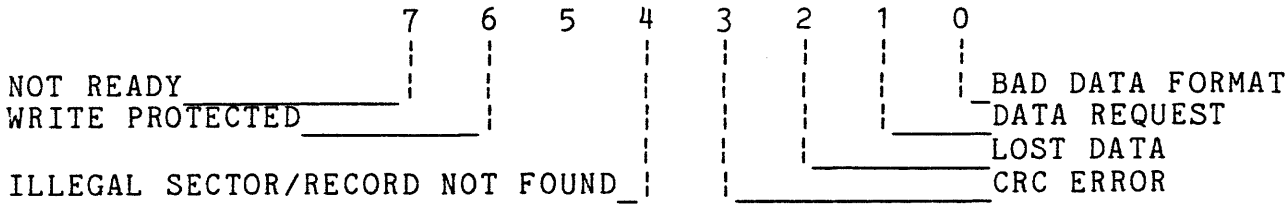
drive, the head load time-out flag is set. The firmware then merges the drive select bits with the head select bit and physically selects a drive, loads the head(s), and selects a side (if the drive is double sided). If the head load time-out bit is set, a 40 millesecond delay occurs to allow for the head to settle after loading. Next the "ready" line from the drive is tested. If the drive is not ready, the head is unloaded and the routine returns to the calling program with the carry bit set and an 80H in the A register. If the drive is ready, the head is positioned in accordance with the most recent seek operation. Head motion (including a head load) or a change of disk drive will cause the firmware to verify the track position by doing a "read header" operation. The correct density of the track is also determined during this operation and the density mode is changed if necessary. If the 1791 controller cannot read the header information in either density, its status is copied into the CPU's A register, the head of the drive is positioned over track zero, and the operation is terminated with the carry set. When the Disk Jockey firmware positions the head to a new track, it reads a header both to determine the proper density and to find out the length and number of the sectors on the newtrack. 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. After (possibly) positioning the head the firmware takes the sector address determined by the most recent set sector operation and compares it to the total number of sectors on the current track. If the desired sector is too large, the carry flag is set and the routine returns with a 10H in the A register. If the value is acceptable, the data from this sector is transferred to memory starting at the address specified by the most recent set DMA operation. The length of this transfer is determined by the length of the sectors on the current track. The last two bytes of data on the sector are not read into memory. These are the CRC check sum bytes and are used to detect data transfer errors. The 1791/8866 chip processes these bytes and then updates its status register. The last operation that the routine performs is to place the status information in the A register and conditionally set the carry flag. The details of these status bits are illustrated below.

Figure 3-1: "READ" REGISTER A ERROR BITS



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

Figure 3-2: "WRITE" REGISTER A ERROR BITS



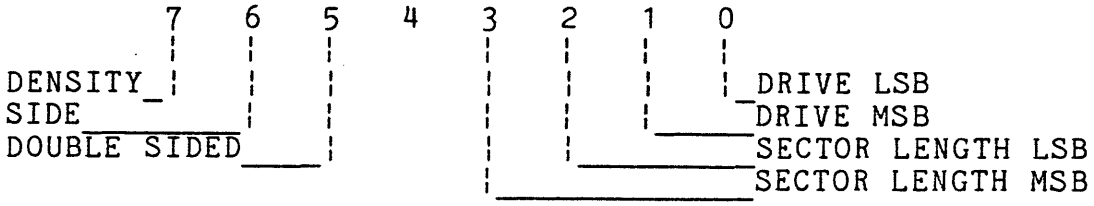
BOOT - Branching to this routine will initiate a bootstrap load operation from the floppy disk. 256 bytes of data will be read (single density mode) into the first page of memory (OH). The bootstrap routine terminates with a branchto OH. Typically, sector 1 and 2 of track zero will contain another bootstrap 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 FLASH utility which is described below. Before sector one is read into memory, various memory locations of the Disk Jockey RAM are initialized. Also BOOT goes through a several second delay to insure that the system

is stable. In order to effect an orderly start-up sequence, BOOT does not require that the drive have a diskette in place when it is called. If the drive is not ready when BOOT is called, it falls into a loop that turns on the LED at the top of the controller and slowly pulses the activity light at the front of the drive. This was done so that BOOT could be started before a diskette was inserted in the drive. When a diskette has been inserted, the door should be closed just AFTER the activity light has been pulsed.

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

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

Figure 3-3: A REGISTER BIT PATTERN



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

SIDE BIT	SIDE SELECTED
0	SIDE 0
1	SIDE 1

Figure 3-4: A REGISTER BIT PATTERNS (cont.)

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

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

Figure 3-5: RECAP OF REGISTER A ERROR BITS

"READ"	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									

"WRITE"	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									

3.4.1. DISKETTE INITIALIZATION

Before a new diskette can be successfully used, it must be initialized. Initializing a diskette destroys any information stored on the diskette. The process of initializing a diskette involves writing the header field of every sector of every track onto the diskette. None of the subroutines described in the section above can be used to write these header fields. This is a safety measure to ensure that an erroneous branch to the firmware EPROM cannot re-initialize a diskette, destroying all the data recorded on it. The initialization function for diskettes is typically provided by a command included in the Disk Operating System. CP/M diskettes from Morrow's/Thinker Toys contain a command called FMTIO# which allows the CP/M users the capability to format diskettes in any of the four IBM compatible formats.

The program REGENIO, provided by Morrow Designs, rewrites diskettes that were written in IBM 3740 format. Each track is read byte by byte, sector breaks and headers are located, and the entire track is rewritten with the correct format.

3.5. UTILIZING DISK JOCKEY FIRMWARE

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

Select the drive to be involved in the transfer. This is accomplished by calling the routine "DRIVE" with the proper drive number in register C. The drive need not be selected before every transfer. A drive once selected will remain selected until another drive is specified. For 2-headed drives, the side of a drive should be specified by calling the SETSID routine with the desired side number in the C register.

If the drive has not been accessed before, the read/write head of the drive is in an unknown position. To initialize the drive a call should be made to "HOME" in order to bring the head to track zero.

Set the DMA address. This involves calling the routine "SETDMA" with the correct value in the B-C register pair. It is not necessary to set the DMA address before every data transfer. If data is always being read into the same area of memory, then only one "SETDMA" call need be made.

Set the read/write head over the desired track. This involves a call to "SEEK" with the desired track number in register C. It is only necessary to call the "SEEK" routine when changing tracks. If the data transfer involves the same track as the previous transfer then no call to "SEEK" should be performed.

Set the desired sector number. The sector can be set by calling "SETSEC" with the correct sector number in register C. If the sector has not changed since the previous "SETSEC" call, as with a read-modify-write sequence, then this routine may be skipped.

Read or write the desired sector. The controller can now be commanded to read or write to the disk by calling "READ" or "WRITE".

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

4. HARDWARE LEVEL REGISTERS

Users desiring a greater level of control over the floppy disk or serial interface may wish to refer directly to the I/O device registers on the DJ from their 8080 or Z80 program. There are fourteen one-byte registers-- four of them read-only, five write

only and three-read/write. The registers have eight I/O addresses on the S-100 bus with a different register being selected during a read operation and a write operation when the addressed register is read-only or write-only. The standard base port address is 70H. This address will be referred to as BASE in the description that follows.

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

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

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

4.1. READABLE REGISTERS

Register 0 - The inverted UART data output register
Location: port 70H (BASE+0).

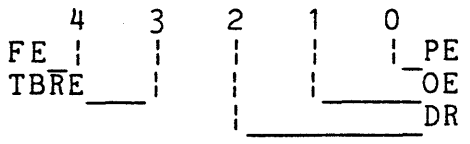
Data is stored in this register by the UART after it has been assembled from the serial data input stream. When a new character is assembled and transferred to this register, the UART sets the DR (Data Ready) flag (see register 1). When this register is read by the CPU, the DR flag is reset by the UART hardware.

HARDWARE LEVEL REGISTERS

Register 1 - The inverted UART status register
Location: port 71H (BASE+1).

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

Figure 4-1: INVERTED UART STATUS BITS



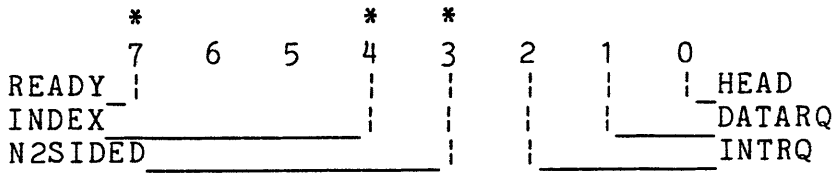
FE = Framing Error
TBRE = Transmitter Buffer Register Empty
DR = Data Ready
OE = Overrun Error
PE = Parity Error

EXAMPLE: When the 1602 has collected a character, the flag DR will be set to 1. When reading the status of this register, however, a byte is available when this bit becomes a zero.

Register 2 - Disk Jockey status register
Location: port 72H (BASE+2).

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

Figure 4-2: DISK JOCKEY STATUS REGISTER



Bits marked with an asterisk reflect the current state of the status lines from the currently selected floppy disk drive. For a detailed specification of these signals see the documentation that accompanies the floppy disk drive. If no drive is currently selected or if the head is not loaded READY and INDEX are low and N2SIDED is high.

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

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

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

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

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

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

Register 3 - Not currently used
Location: port 73H (BASE+3).

Register 4 - 1791/8866 controller status register
Location: port 74H (BASE+4).

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

4.2. WRITE-ONLY REGISTERS

Register 0 - The inverted UART data input register
location: port 70H (BASE+0).

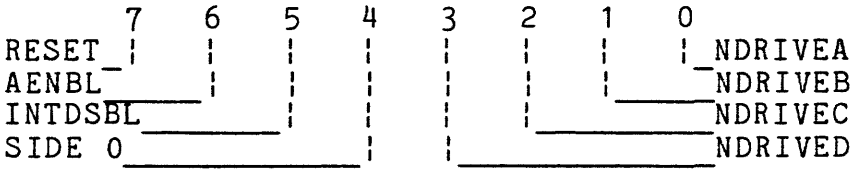
Inverted data is stored is this register by the CPU for serial output by the UART. The UART transfers the data from this register to an internal parallel load serial output register where a start bit, optional parity bit and the stop bits are added to the data. Whenever the UART empties register 0, the TBRE status bit (see read register 1) is raised to inform the CPU that it is possible to output more data to the UART. Data sent to the UART must be complemented in software.

HARDWARE LEVEL REGISTERS

Register 1 - Disk Jockey drive control register
Location: port 71H (BASE+1).

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

Figure 4-3: DRIVE CONTROL REGISTER



RESET- When a one is stored in this bit, the master reset pin of the 1791/8866 is active and the controller chip is in a reset condition and will not accept any commands. The Voltage Controlled Oscillator of the Phase Lock Loop is also disabled and the Phase Lock Loop will not process any data to produce data windows for the 1791/8866. This bit is used to reinitialize the 1791/8866 in the event that the micro-program in the controller chip becomes confused and gets lost trying to read bad data. When a zero is stored in this bit (after a one value) the VCO of the Phase Lock Loop will properly start and the 1791/8866 will execute a home command and place itself in a state to accept commands.

AENBL- When the CPU references the 1791/8866's data register during a data transfer, the PREADY line (S-100 bus line 72) is brought low which puts the processor in a wait state. The CPU remains in this state until the 1791/8866 raises its DATA REQUEST line. This mode of operation dispenses with the usual status test during data transfers and makes it possible for the Disk Jockey to run at double density speeds without having to use a DMA channel. However, there are times when the CPU needs access to the data register even though the DATA REQUEST LINE is low and will stay low (just before a seek command is issued, for example). When the AENBL bit is a one, the stall logic that usually governs accesses to the 1791/8866's data register is disabled. This allows the CPU to have access to this register as if it were a normal IO location. However, before the Disk Jockey can

move data to or from the floppy disk drive, this bit must be a zero so that the CPU can synchronize its data transfers to the 1791/8866 controller.

INTDSBL-When this bit is a zero, the interrupt request line of the 1791/8866 controller is enabled to request interrupts on the S-100 system bus. When this bit is a one, no interrupts can be generated by the controller. The user should consult the 1791/8866 data sheet for a thorough understanding of the chip's interrupt request line.

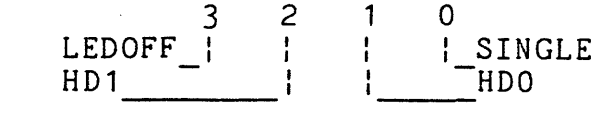
SIDE 0- When a double headed drive is connected to the Disk Jockey, a zero in this bit will enable head 1 whenever the drive is selected. A one will enable head 0. If a single headed drive is selected, this bit has no effect on the drive.

NDRIVED-By setting one of the four low order bits to zero, a
to drive will be selected when a head load command is given.
NDRIVEA Bit 0 represents the first drive, and bit 3 represents drive 4. Only one of the four low order bits of this register should ever be a zero. If more than one of these bits are zero, loading the head will select more than one drive and cause data errors during reads and possible head position errors on seeks.

Register 2 - The Disk Jockey function register
Location: port 72H (BASE+2).

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

Figure 4-4: 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 DJIO board will read and write data to and from the disk in single density. When this bit is a zero, reads and writes are performed in double density.

HD0,HD1-These two bits control the loading of the read/write head. Their functional characteristics are detailed in the table that follows.

Figure 4-5: HEAD CONTROL BITS

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

Register 3 - Bank Select
Location: port 73H (BASE+3).

By outputting a zero or a one to this location, the memory mapped portion of the DJ IO can be switched on or off. A one enables the bank select, a zero disables. This does not affect the I/O ports.

When the bootstrap on the DJ IO is used, the on-board memory will be selected. After the first page of normal RAM has been loaded (0H-100H), the DJ IO EPROM and RAM are deselected by the standard bootstrap routine. For an example of this, look at the SDBOOT routine in the CBIOS listings.

If you want to use the jump table in the EPROM, the board must be selected and any memory in the same address space must be disabled.

Register 4 - 1791/8866 controller command register
Location: port 74H (BASE+4).

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

4.3. READ-WRITE REGISTERS

Register 5 - 1791/8866 track register
Location: port 75H (BASE+5).

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

HARDWARE LEVEL REGISTERS

Register 6 - 1791/8866 sector register
Location: port 76H (BASE+6).

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

Register 7 - 1791/8866 data register
Location: port 77H (BASE+7).

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

FINAL NOTE

The Disk Jockey firmware contains numerous examples illustrating the use of the hardware registers listed above. A comprehensive study of the two Western Digital (for the 1791 and 1602) or the Fujitsu 8866 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. Please refer to the listings in the back of this manual of the EPROM and the CBIOS for examples.

5. HARDWARE FEATURES GUIDE

The DJ IO disk controller is designed to allow simultaneous access to the entire 64k address space of S-100 systems and floppy disks. The flexibility to install the DJ IO in most 8080 Z-80 systems is provided for by switch selectable features. Most switches will only need to be set to initially configure the Disk Jockey IO to your system's architecture. The Disk Jockey IO also allows the option of responding to the full 16 megabyte address space of the IEEE 696 specifications for S-100 bus systems.

There are five banks of switches on the board, and one set of strappable pads. These six areas are described in the sections that follow and identified below:

Power on Jump Switch Bank

Reset Options/Port Select Switch Bank

BAUD Rate Switch Bank

Phantom Options Switch Bank

Extended Addressing Switch Bank

Interrupt Strapping Pads

Following this, the normal bootstrap operation is described, and construction and connection of cables for the DJ IO.

5.1. POWER ON JUMP SWITCH BANK

This switch is located at 9C, in the center of the third row of chips, just above the words "DISK JOCKEY / IO". Its primary purpose is to enable/disable the board's Power on Jump capability, and to set the address for this feature. It also controls the extended addressing feature.

The DJ IO board has the ability to place a sequence of addresses on the S-100 bus. This task is normally the province of the CPU. During a Power on Jump, the DJ IO sets true the Address Disable* (pin 22) S-100 signal, which should turn off the CPU's external address bus. Then a sequence of eight consecutive addresses are placed on the bus. This allows the addressed memory to be read by the CPU. The sixth of these eight bytes read by the Power on Jump sequence should be a Jump Instruction, which loads the Program Counter of the CPU with the next two bytes. When the last byte is read, Address Disable is made false, and the CPU takes over by jumping to the location contained in the last two bytes.

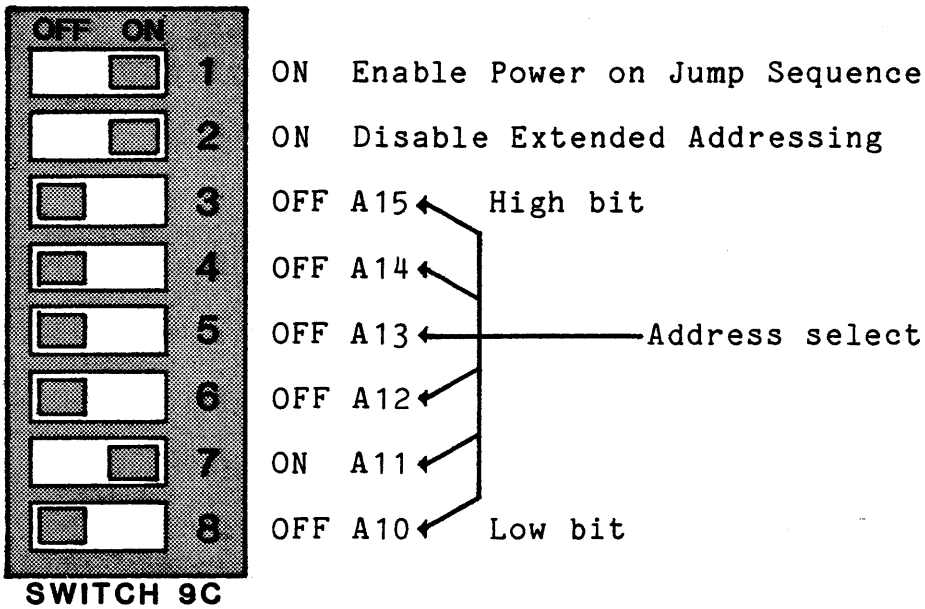
The first switch enables the Power on Jump feature when it is turned on. If two devices are simultaneously attempting a Power on Jump, the system will not boot. If you wish the Disk Jockey IO to perform the bootstrap operation, switch 1 should be on.

The second switch disables extended addressing capabilities of the DJ IO when it is on. By turning this switch off, the DJ IO will attempt to decode the extended address lines (A16-A23) for board memory enable. If you are not using extended addressing, this switch should remain on and the chip at 4D, 25LS21, should be removed and saved for future use.

The DJ IO does not produce a 24 bit address for bootstrapping. If you are using extended addressing, a different board must produce a 24 bit address to allow the bootstrap in EPROM to be accessed.

Switch 3 represents the most significant bit (A15) and switch 8 the least significant settable bit (A10) of the Power on Jump address. A switch that is OFF represents a 1, and a switch that is ON represents a 0. For example: the location of the boot sequence for a standard DJ IO is F7F0H or 11110111 11110000 Binary. Only the higher order 6 bits are selected by the switch: 3,4,5,6,8 are off (for 1's), and 7 is on (for a 0). The DJ IO's boot sequence is 5 NOP's, followed by a jump to F000. This sequence is located in the second to last eight bytes in the EPROM.

Figure 5-1: POWER ON JUMP SWITCH BANK (at 9C)



SWITCH 9C

5.2. RESET OPTIONS/PORT SELECT SWITCH BANK

This switch bank is located at 8B, in the second row of chips and in the middle of the board. It controls the selection of port address, reset options and a wait state.

When the DJ IO is used in systems with a clock greater than 2 megahertz, switch 1 should be on. Turning this switch on will set the PRDY* line true while the EPROM is being accessed, for one wait state. This is done because the 2716 EPROM has a slower memory fetch cycle time than ordinary memory.

The next two switches control the memory mapped portion of the board during RESET* and POC*. When switch 3 is on and 2 is off, the DJ IO will wake-up bank selected when the system is reset or a Power on Jump occurs. If you intend to use the Power on Jump feature, that is you have turned on switch 1 of the Power on Jump Switch Bank, then switch 3 of Reset Options/Port Select MUST be on.

If some other device is handling the bootstrap, than switch 3 should be off, and switch 2 should be on. When switch 2 is on, the DJ IO's memory will be deselected after a RESET* or a POC*. This allows full use of 64k of read/write memory, and does not interfere with the use of the board.

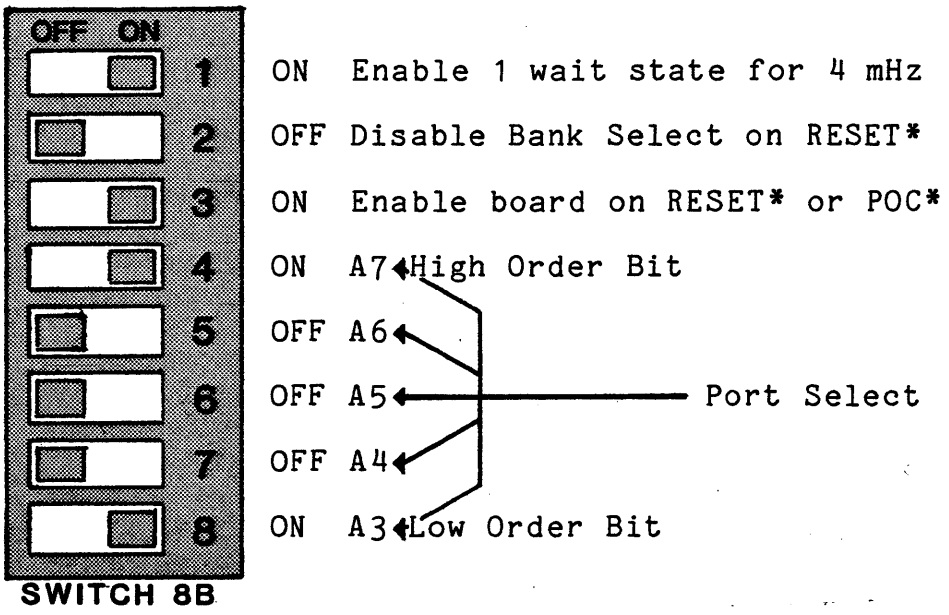
> NOTE: Either switch 2 OR switch 3 should be on, NOT BOTH. They are mutually exclusive states, and having both on or both off will leave the board in an unknown (confused!) state until a bank select command is issued.

Switches 4 through 8 represent the I/O port address of the board. These switches allow you to change which eight consecutive port addresses the DJ IO will respond to. However, the software will not be changed by changing these switches. The bootstrap routine in EPROM, and the standard operating system expect to find the Disk Jockey IO at ports 70H-77H. If these switches are changed, and the software is not, the board will not respond.

So, assuming you have assembled your own software switches 4-8 allow you to set the board to any consecutive group of port address that is evenly divisible by 8. This is because only the 5 high order bits can be changed. Like the Power on Jump switches, turning a switch ON represents a zero, and OFF sets the bit to one.

> EXAMPLE: For the standard DJ IO, switches 4 and 8 should be ON (for zeros), and switches 5, 6 and 7 OFF, (set to ones), for 01110(000)Binary or 70H. To set the port base address to F0H, switches 4,5,6 and 7 would be OFF, and switch 8 would be ON (11110(000)Binary=F0H).

Figure 5-2: RESET OPTIONS/PORT SELECT SWITCH BANK (at 8B)



5.3. BAUD RATE SWITCH BANK

This switch is located at 14C, in the third row of chips, near the right edge of the board and the 1602 UART chip that it controls. It is used to set the options for using the serial port of the DJ IO. The software provided with the Disk Jockey IO expects to use this serial port for the console. So, unless you modify your software to address a different port, the settings of this switch should be set to agree with your terminal.

The first four switches select the format for serial transmission. Switch 1 selects parity type: when on, the parity will be odd, when off, the parity bit will be even. The parity bit will not be used unless switch 4 is turned on. When switch 4 is turned off, no parity will be selected.

Switch 2 sets the word length: when it is on, 7 bits will be transmitted; when off, eight bits will be sent (or expected to be received).

The number of stop bits is selected by switch 3. When switch 3 is on, 1 stop bit will be sent; when it is off, 2 stop bits are transmitted (or looked for in the incoming serial stream). Normally, devices are fairly tolerant of the number of stop bits.

> EXAMPLE: A customary format for serial ports (which will be called "Normal Use" in the diagram) is to use 8 data bits, no parity and two stop bits. This translates as switches 1-4 being OFF.

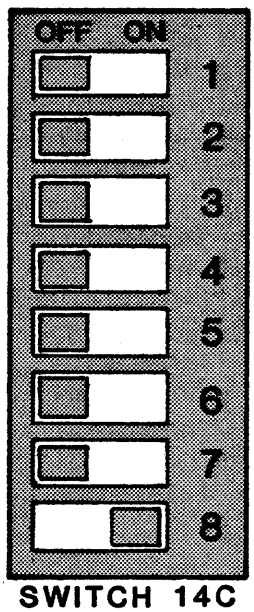
The baud rate is selected using switches 5-8. For CRT terminals, 9600 baud is often the highest rate available. Using the highest possible rate will improve system performance. For 9600 baud,

switches 5,6 and 7 would be OFF, and switch 8 ON. A complete list of possible baud rates follows:

Table 5-1: BAUD RATE SWITCH SETTINGS

SW-5	SW-6	SW-7	SW-8	BAUD RATE
on	on	on	on	50
on	on	on	off	75
on	on	off	on	110
on	on	off	off	134.5
on	off	on	on	150
on	off	on	off	300
on	off	off	on	600
on	off	off	off	1200
off	on	on	on	1800
off	on	on	off	2000
off	on	off	on	2400
off	on	off	off	3600
off	off	on	on	4800
off	off	on	off	7200
off	off	off	on	9600
off	off	off	off	19200

Figure 5-3: BAUD RATE CONTROL SWITCH BANK (at 14C)



OFF Parity select: on=odd, off=even

OFF Word length: on=7 bits, off= 8 bits

OFF Stop bits: on=1 stop bits, off=2 stop

OFF Parity enable: on=parity, off=no parity

OFF ←
OFF ←
OFF ←
ON ←

Baud Rate Switches
This example is for 9600 baud,
8 bits, 2 stop bits, no parity

5.4. PHANTOM OPTION SWITCH BANK

This switch bank, of only four switches, is located at 4B, in the second row of chips, near the left of the board. Of these four, switches 1 and 2 are not used. Switches 3 and 4 control what the DJ IO will do with the Phantom* line (pin 67).

The Phantom* line, pin 67 of an S-100 bus, is used to disable memory mapped devices. When Phantom* goes true, any board which uses the Phantom* signal is deselected from the address space. This is handy because it allows both the DJ IO RAM and EPROM to be present in memory along with system RAM. the DJ IO board's memory mapped portion must appear in the address space during on-board bootstrapping.

The Disk Jockey IO can either generate a Phantom* signal, or respond to it. When switch 4 is turned on, the DJ IO will set Phantom* true when it is bank selected AND its memory is addressed. Thus, during a bootstrap procedure, the conflicting system RAM could be turned off by Phantom* while the EPROM is being read. The EPROM reads the first two sectors off the diskette into low memory, and transfers control to the program there. In the case of standard DJ IO software, the board is then deselected, and Phantom* is no longer generated. This allows high memory to be loaded. It may be necessary to change the configuration of the memory board affected, so that it will respond to Phantom*.

The DJ IO can also respond to the Phantom* signal. If switch three is on, the Phantom* signal will disable the the DJ IO board's memory. This does not affect the I/O ports.

> NOTE: ONLY ONE of these two switches can be on at a time. Otherwise, the DJ IO will attempt to deselect itself when it generates the Phantom* signal. Both can be left off safely.

Switches 1 and 2 are not used.

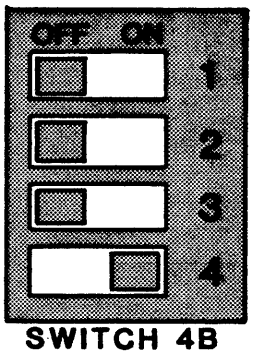


Figure 5-4: PHANTOM OPTION SWITCH BANK (at 4B)

1	OFF	Not used
2	OFF	Not used
3	OFF	Disable DJ IO when Phantom* true
4	ON	Generate Phantom* signal when DJ IO is both selected and addressed

5.5. EXTENDED ADDRESS SWITCH BANK

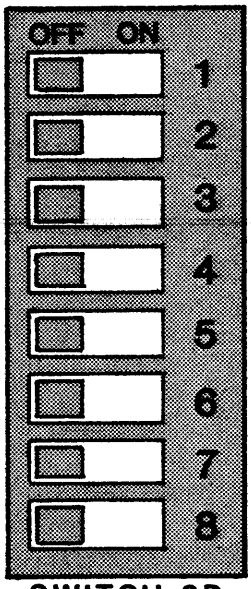
This switch is located at 3D, down in the fourth row of chips, in the lower left corner of the board. It allows the Disk Jockey IO to respond to memory request in the 16 megabyte address space of the IEEE 696 S-100 bus specifications. What this means is if your system is capable of using the eight address lines in addition to the customary 16, the DJ IO can decode these addresses.

To use this feature, switch 2 of the Power on Jump Switch Bank (at 9C) must be turned OFF. Turning this switch ON disables extended address capabilities.

The Extended Address Switch Bank is arranged so that switch 1 represents the high bit (A23), and switch 8 the low bit (A16). To set a bit to a 1, turn the switch OFF, and to a 0, turn it ON.

> EXAMPLE: For the DJ IO RAM to be located at AAF800, switches 1,3,5 and 7 would be turned OFF, and switches 2,4,6 and 8 would be turned ON. And, of course, switch 2 of the POJ Switch Bank (at 9C) would be turned OFF.

Figure 5-5: EXTENDED ADDRESS SWITCH BANK (at 3D)



SWITCH 3D

OFF A23 High Order Bit

OFF A22

OFF A21 off=1

OFF A20 on=0

OFF A19

OFF A18 Switch 2 of POJ (at 9C) must be OFF to enable this switch.

OFF A17

OFF A16 Low Order Bit

5.6. INTERRUPT VECTOR STRAPPING PAD

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

5.6.1. INTERRUPT LOGIC

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

The Disk Jockey interrupt logic is implemented by installing a jumper at the lower left hand area of the circuit board. The jumper should originate at the open pad just above J1 and should connect to ONLY ONE of the pads below the symbols VI 0 through 9 or PINT. Unless there is a vectored interrupt controller on the bus or on the system's CPU board, the jumper connection should be made to PINT. After the interrupt jumper is installed, interrupts from the 1791/8866 can be enabled or disabled by outputting a 0 or 1 in bit 5 of the Disk Jockey drive control register (write only register 1). For the details please refer to the section on Hardware Level Registers. The jumper pad layout for installing interrupts on the DJ IO board are shown below:

```
                                                    *
                                                   J1

VI  0  1  2  3  4  5  6  7  PINT
    *  *  *  *  *  *  *  *  *
```

5.7. NORMAL BOOTSTRAP PROCEDURES

Just to the left of P1, the right angle header connector for the disk drive,is the error LED. This LED (light emitting diode) will slowly flash on and off if the DBOOT routine cannot load the bootstrap from the diskette. Since the boot routine does not use any of the terminal I/O logic, this LED is helpful in determining whether the DJ IO was successfully accessed for the beginning of a bootstrap operation, but unable to read the first two sectors of the boot diskette.

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

when it goes off, close the drive door. The system will boot the next time the drive activity light goes on.

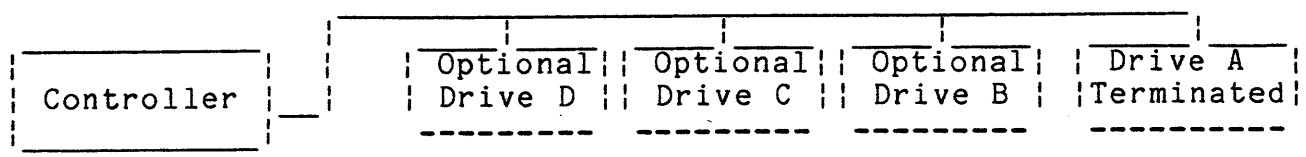
Whenever the bootstrap load DBOOT routine is called, the head on Drive A will not load (as evidenced by the drive activity LED at the front of the drive) for a second or two. There is a built in delay in DBOOT to make sure that all components of the system are stable and have finished any reset processes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

EPROM Boot Function

When a jump is made to the first location in the EPROM, a cold boot routine is initiated. As described above, first there is a several second delay. Then, if there is no diskette in drive A, the LED on the drive
will slowly flash. Once a diskette has been inserted, the head will be loaded, and a seek home command will be issued. The boot routine expects the first two sectors of the diskette to be in single density, 128 bytes/sector. These sectors should contain a program to read in enough of the operating system to take over control. The routine on the first two sectors will be loaded into memory from 0H to 100H, and a Jump to 0 will terminate the EPROM boot routine. If the DJ IO is unable to read sectors one or two, it will leave the error LED flashing.

5.8. CABLE CONNECTIONS

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



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

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

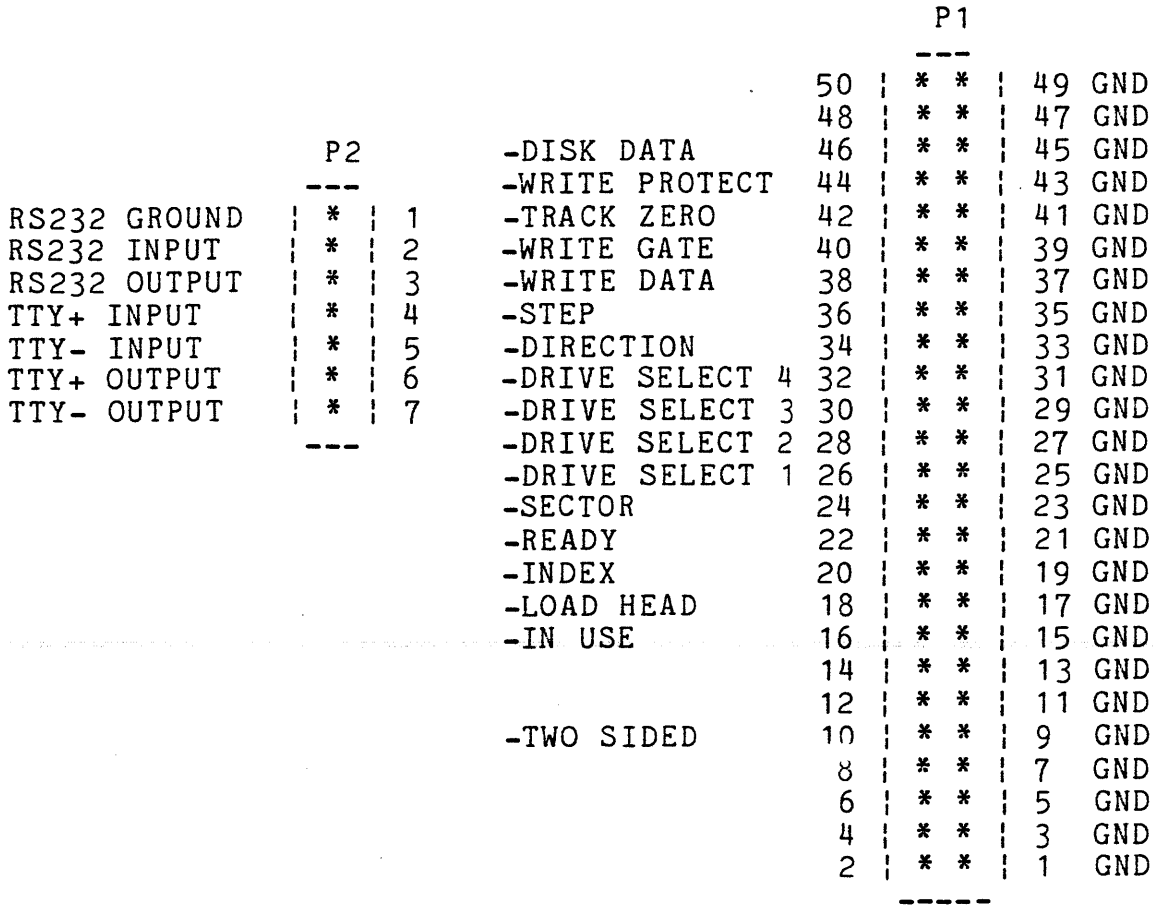
Four different daisy chain cables are available for one, two, three or four drive systems. A daisy chain cable is simply a parallel cable. Not all available connectors on a multiple drive cable need be filled for the system to function. Also, a dual system with drives addressed, say, as "A" and "C" would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

The following rule applies to all cable configurations supplied by Thinker Toys:

> The 50 pin flat ribbon cable provided with the Discus system should be connected to the Disk Jockey controller board so that the cable extends out over the solder side of the PC board-- not the component side.
>
> Whichever end of the 50 pin flat ribbon cable is chosen to plug into the controller board, that side of the cable which is on the LEFT (closer to the heat sink) as it connects to the controller should be UP as it connects to each and every drive on the system. Thus, J1 pin 50 on the DJ controller board should come in to each disk drive via the top part of the male 50 pin connector attached to the cabinet of each drive. If the LED on the front of the drive comes on upon power up, the cable is on backwards and should be reversed. The LED on the front of the drive should light up only when a command has been issued to load the head.
>
> Any visual "key" such as an arrow or triangle on a connector should be used solely as an aid in implementing the connection scheme described above.

5.8.1. I/O CONNECTORS P1 AND P2

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



6. PARTS LIST

DESCRIPTION	MORROW DESIGN PART #	QUANTITY
Diode 1N751	028-1N751	1
Diode 1N914 [1N4454][1N3600]	028-1N914	8
Transister 2N3904	028-2N3904	2
Transister 2N3906	028-2N3906	2
Regulator +5 volts	028-7805	2
Regulator 12 volt [340T-12]	028-7812	1
Regulator -12 volts	028-79L12	1
LED, single lens	028-LEDRL209	1
Resister 1K ohm 1/4W 5%	030-C0205-102	8
Resister 10K ohm 1/4W 5%	030-C0205-103	2
Resister 1meg ohm 1/4W 5%	030-C0205-105	4
Resister 1.5K ohm 1/4W 5%	030-C0205-152	1
Resister 18.2Kohm 1/4W 1%	030-M0201-18.20	1
Resister 240 ohm 1/4W 5%	030-C0205-241	1
Resister 27K ohm 1/4W 5%	030-C0205-273	2
Resister 3.3K ohm 1/4W 5%	030-C0205-332	7
Resister 390 ohm 1/4W 5%	030-C0205-391	1
Resister 4.7K ohm 1/4W 5%	030-C0205-472	4
Resister 47K ohm 1/4W 5%	030-C0205-473	1
Resister 560 ohm 1/4W 5%	030-C0205-561	2
Resister 750 ohm 1/2W 5%	030-C0205-751	2
Resister 5.36Kohm 1/4W 1%	030-M0201-05.36	2
Resister 20.5Kohm 1/4W 1%	030-M0201-20.50	1
Resister 54.9Kohm 1/4W 1%	030-M0201-54.90	1
Resister 86.6Kohm 1/4W 1%	030-M0201-86.60	1
SIP 180 1/8W 5% 10 Pins	030-S0105-181-10	1
SIP 3.3K 1/8W 5% 8 Pins	030-S0105-332-08	3
SIP 4.7K 1/8W 5% 10 Pins	030-S0105-472-10	2
Capacitor .001 mf cer. disk	033-CE001A	1
Capacitor .01 mf cer. disk	033-CE001B	23
Capacitor .01 mylar cap.	033-MY001B	1
Capacitor 20 pf silv mica	033-SM020	2
Capacitor 47 pf silv mica	033-SM047	2
Capacitor 56 pf silv mica	033-SM056	1
Capacitor 100 pf silv mica	033-SM100	3
Capacitor 2.2 uf dip tant	033-TD022-20	11
Crystal 5.0688 mega Hz.	037-MZ05.0688	1
Crystal 10 mega Hz.	037-MZ10	1
Inductor 2.2 uH axial	039-IND2.2	1
Switch, 4 position DIP	041-DS04	1
Switch, 8 position DIP	041-DS08	4
PCB header (serial) 7 pin	043-07SRF	1
PCB header (drive) 50 pin	043-50DRH	1
Screw 6/32 x 5/16 pan phil.	096-06X516PP	3
Hex Nut 6/32	098-0632HN	3
Heatsink, slim line 5 prong	094-S0521	2
IC socket, 8 pins, low prof.	039-SOCLP-08	1
IC socket, 14 pin, low prof.	039-SOCLP-14	13
IC socket, 16 pin, low prof.	039-SOCLP-16	11

DISK JOCKEY IO PARTS LIST

DESCRIPTION	MORROW DESIGNS PART #	QUANTITY
IC socket, 18 pin, low prof.	039-SOCLP-18	3
IC socket, 20 pin, low prof.	039-SOCLP-20	7
IC socket, 24 pin, low prof.	039-SOCLP-24	1
IC socket, 40 pin, low prof.	039-SOCLP-40	2
IC 1863 UART [1602]	026-IC1863	1
IC 2114-3 RAM	026-IC2114-3	2
IC 25LS2521	026-IC25LS2521	1
IC 2941	026-IC2941	1
IC 7404	026-IC7404	1
IC 74LS00	026-IC74LS00	1
IC 74LS02	026-IC74LS02	1
IC 74LS04	026-IC74LS04	1
IC 74LS08	026-IC74LS08	1
IC 74LS10	026-IC74LS10	1
IC 74LS138	026-IC74LS138	1
IC 74LS161	026-IC74LS161	1
IC 74LS175	026-IC74LS175	1
IC 74LS221	026-IC74LS221	1
IC 74LS240	026-IC74LS240	1
IC 74LS244	026-IC74LS244	2
IC 74LS273	026-IC74LS273	1
IC 74LS367	026-IC74LS367	1
IC 74LS368	026-IC74LS368	4
IC 74LS373	026-IC74LS373	1
IC 74LS38	026-IC74LS38	1
IC 74LS390	026-IC74LS390	1
IC 74LS74	026-IC74LS74	4
IC 8131	026-IC8131	1
IC 81LS96	026-IC81LS96	1
IC 8866 floppy disk contr.	026-IC8866	1
IC 96LS02	026-IC96LS02	1

B

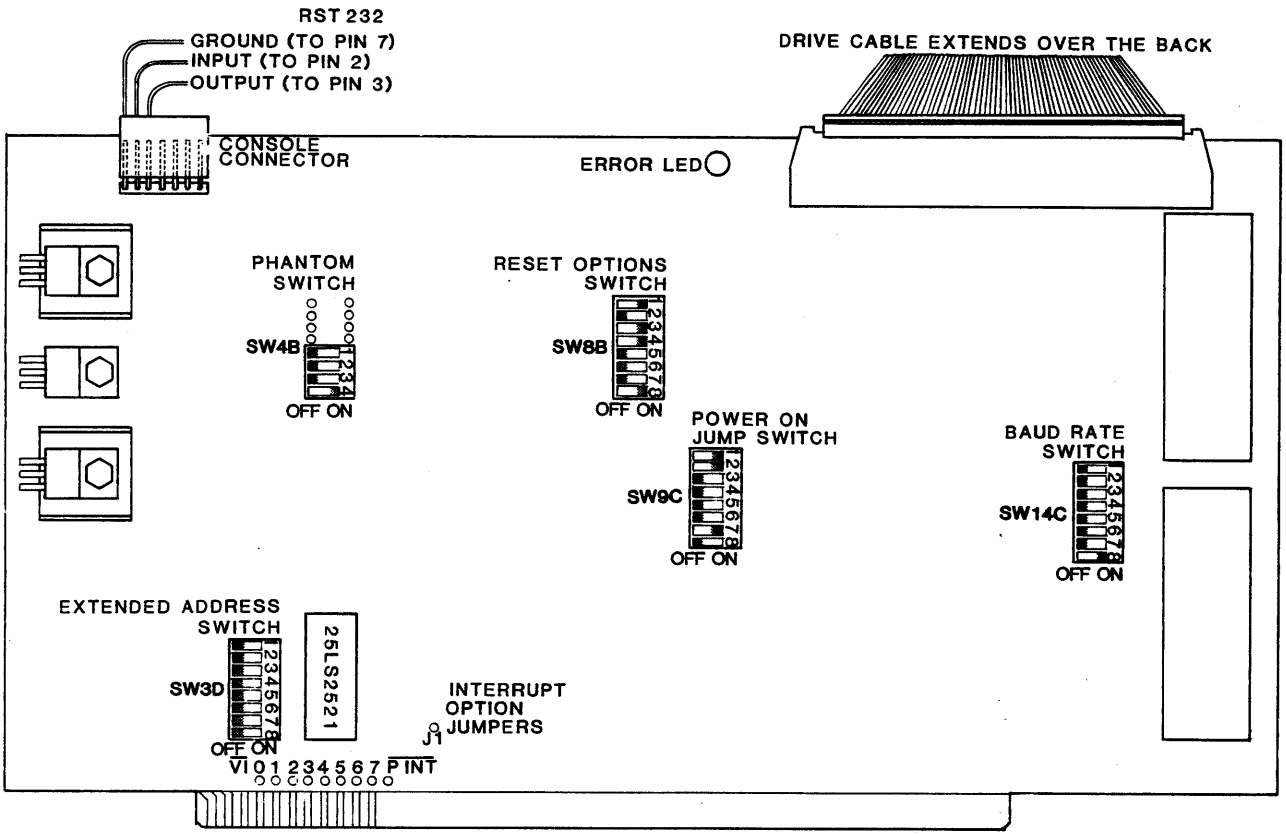
C

D

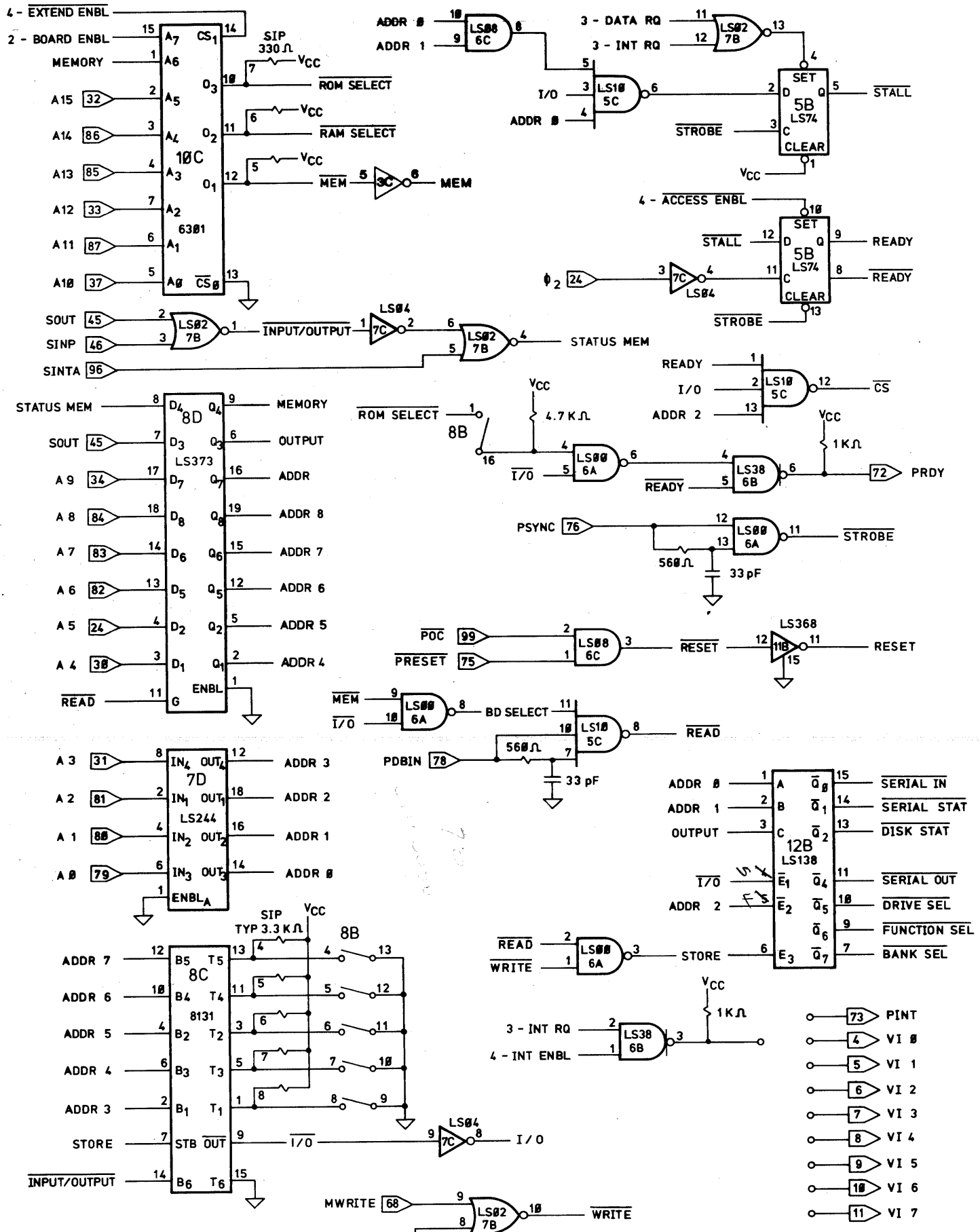
E

F

I

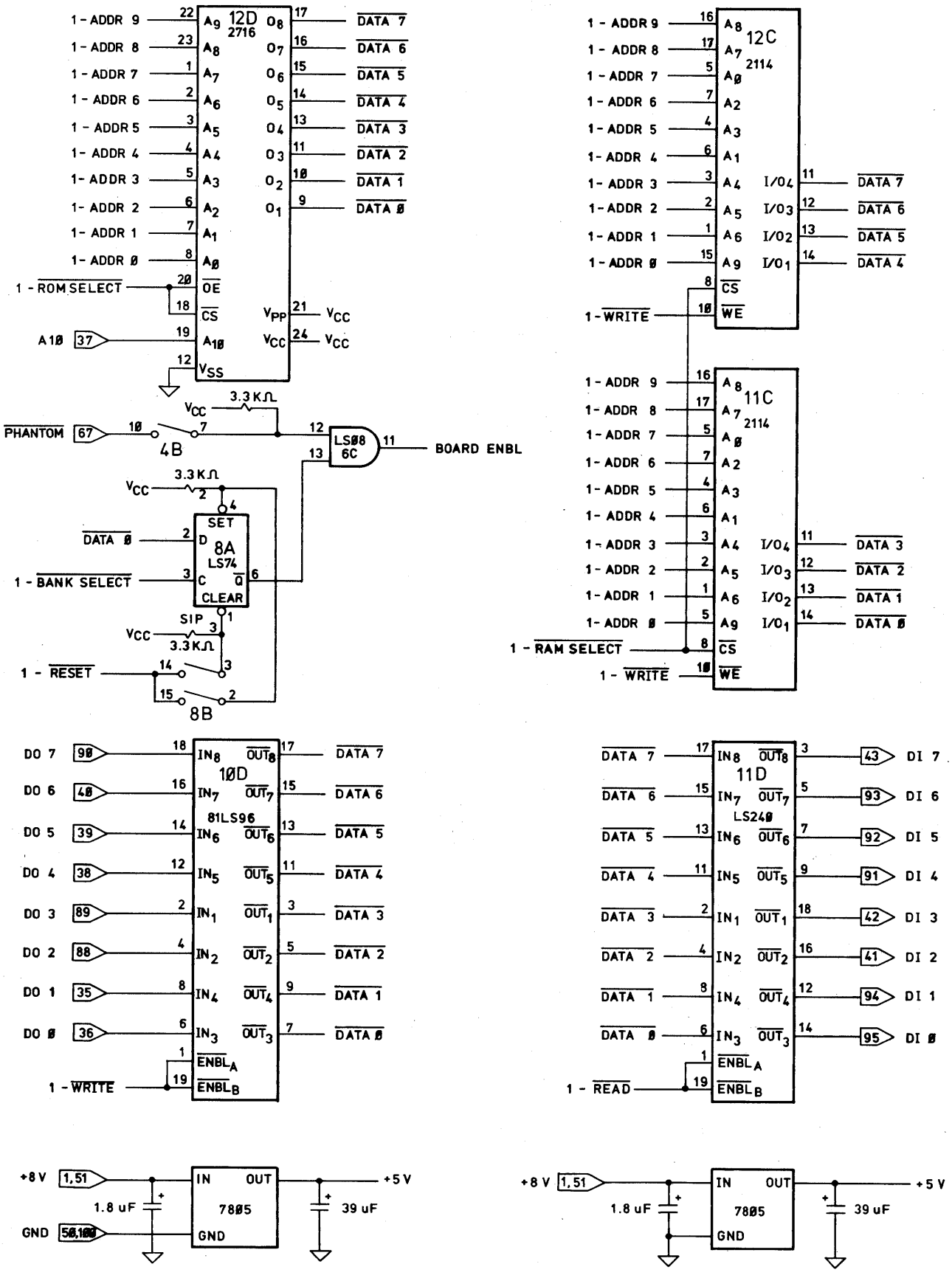


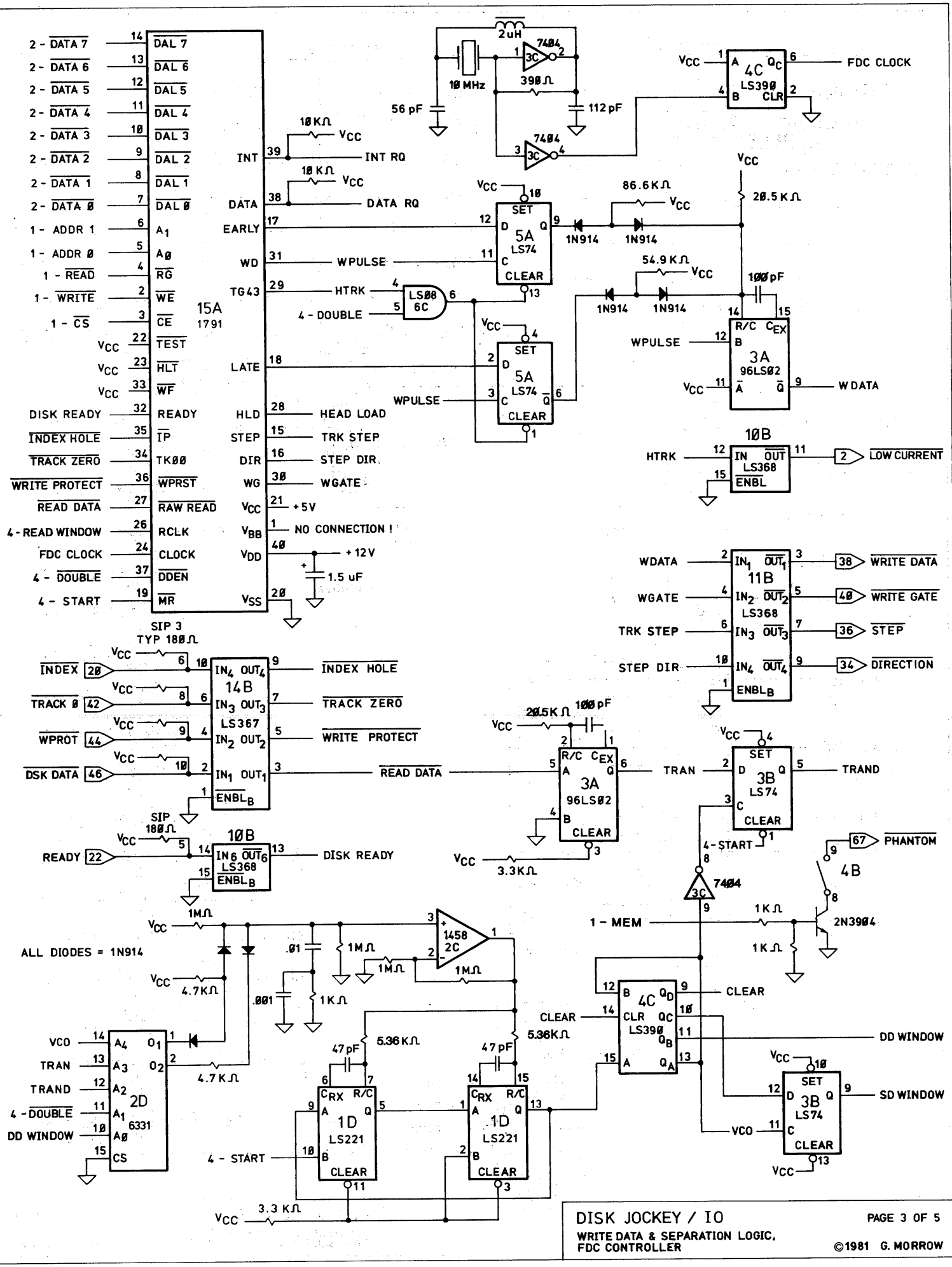
Disk Jockey 2/DIO Component Layout



DISK JOCKEY / IO

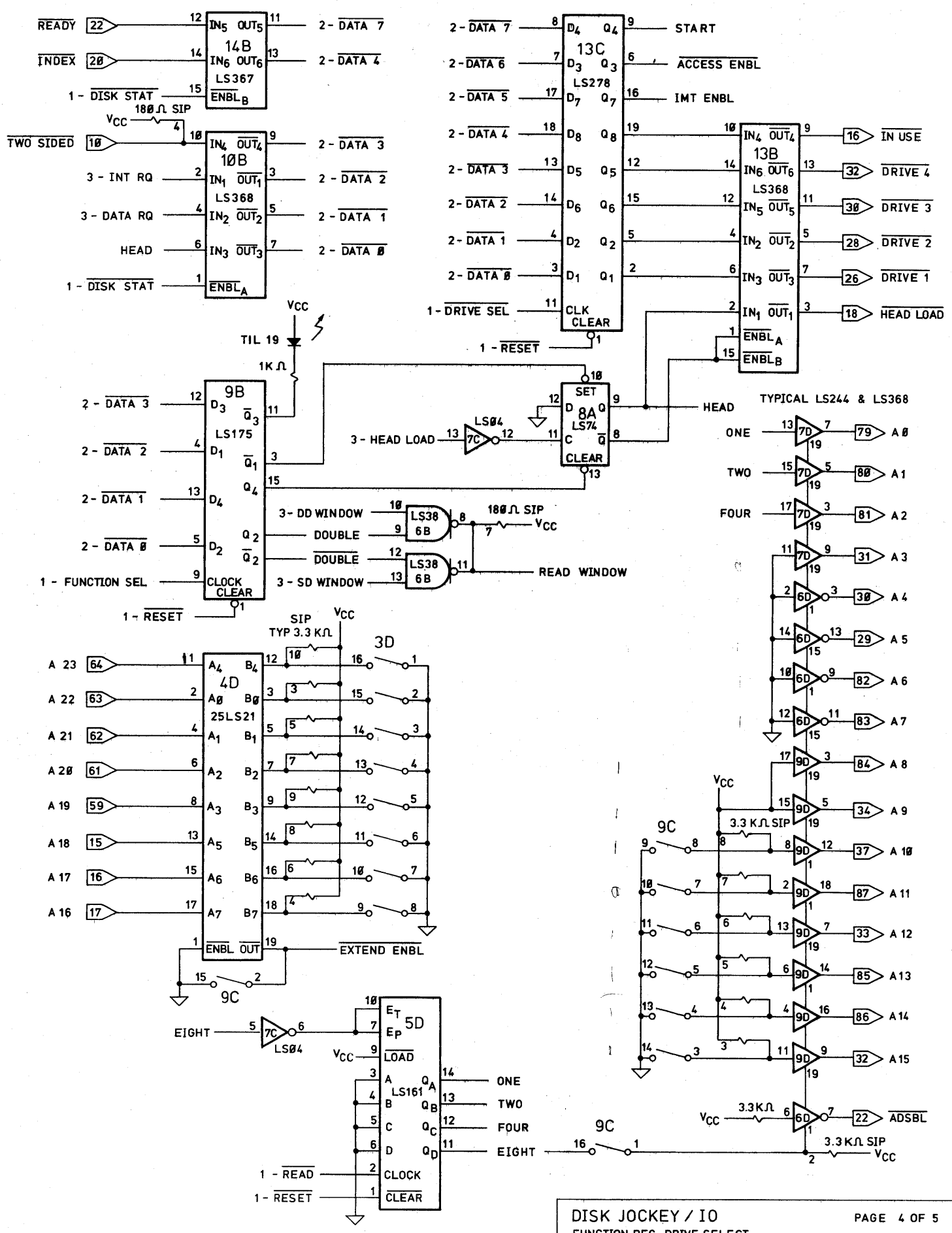
PAGE 1 OF 5





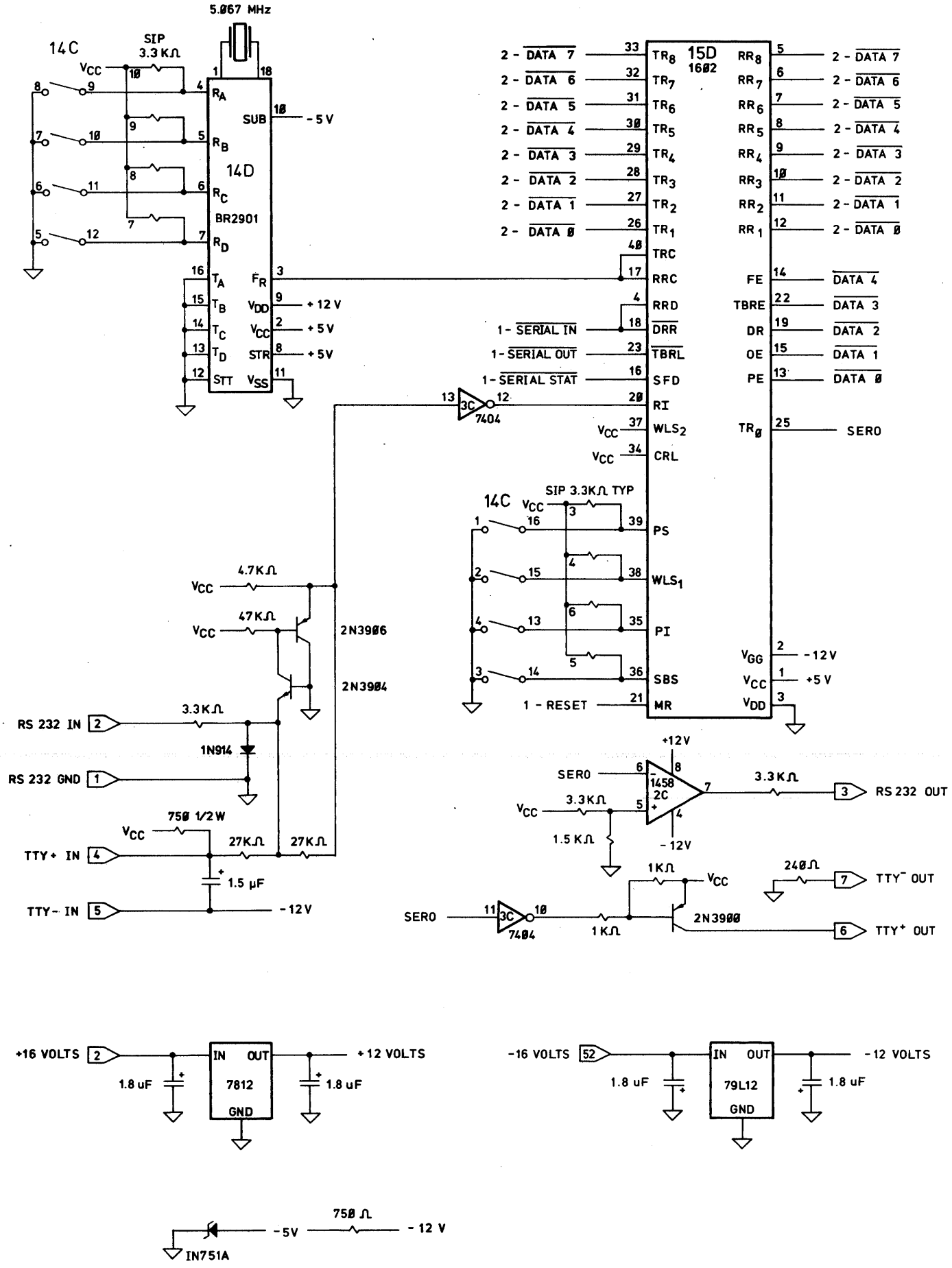
DISK JOCKEY / IO

WRITE DATA & SEPARATION LOGIC, FDC CONTROLLER



DISK JOCKEY / IO

FUNCTION REG., DRIVE SELECT, BANK SELECT, & POWER UP LOGIC



DISK JOCKEY / IO

PAGE 5 OF 5

SERIAL INTERFACE LOGIC