DJ/DMA Floppy Disk Controller Technical Manual Revision 1

April 1982

MORROW DESIGNS

DJ/DMA Floppy Disk Controller

Technical Manual

Revision 1

Table of Contents

1. INTRODUCTION.. 1

2. PROGRAMMING SPECIFICATIONS.......................... 3
   2.1. The Channel Concept............................ 3
   2.2. The Start Channel Command...................... 4
   2.3. The Channel Command Address.................... 4
   2.4. Command Structure.............................. 5
   2.5. DJDMA Controller Commands...................... 5
   2.6. Controller Command Specifications.............. 6
        2.6.1. SET DMA ADDRESS......................... 6
        2.6.2. READ SECTOR............................. 7
        2.6.3. WRITE SECTOR............................ 9
        2.6.4. SENSE DRIVE STATUS 9
        2.6.5. SET INTERRUPT REQUEST................... 12
        2.6.6. SET ERROR RETRY COUNT................... 13
        2.6.7. SET LOGICAL DRIVE....................... 13
        2.6.8. SET HEAD UNLOAD/DRIVE DESELECT TIMEOUT....... 14
        2.6.9. READ TRACK.............................. 14
        2.6.1Ø. WRITE TRACK............................ 15
        2.6.11. OUTPUT TO SERIAL PORT.................. 16
        2.6.12. SERIAL INPUT ENABLE/DISABLE............ 16
        2.6.13. CONTROLLER HALT........................ 17
        2.6.14. BRANCH IN CHANNEL...................... 17
        2.6.15. SET CHANNEL ADDRESS.................... 17
        2.6.16. SET TRACK SIZE......................... 18
        2.6.17. READ CONTROLLER MEMORY 18
        2.6.18. WRITE CONTROLLER MEMORY................ 19
        2.6.19. EXECUTE CONTROLLER ROUTINE............. 19
   2.7. Command Summary................................ 2Ø
   2.8. Status Codes................................... 2Ø

3. IEEE 696 (S-1ØØ) BUS CONSIDERATIONS................. 21

4. INTERRUPTS.. 21

5. I/O CONNECTORS...................................... 22

6. JUMPERED SETTINGS................................... 22
   6.1. EPROM Replacement.............................. 22
   6.2. Bootstrap Program.............................. 23

Table of Contents, Cont.

7. BOOTSTRAP LOAD.. 23

8. BOOTING THE DJDMA...................................... 24

9. FORMATTING DISKETTES.................................... 24

Parts List.. P-1
Subject Index... I-1
Software Listing.. L-1
Component Layout/Schematic................................ S-1

List of Tables

2-1. Status Byte Codes.................................... 8
2-2. STATUS BYTE 1: Drive Characteristic Byte............. 1Ø
2-3. STATUS BYTE 2: Sector Length Code - Ø, 1, 2, or 3.... 1Ø
2-4. STATUS BYTE 3: Drive Status/Characteristic Byte...... 11
2-5. Supported Commands................................... 2Ø
2-6. Unsupported Commands................................. 2Ø
2-7. Status Code Summary.................................. 2Ø

7-1. 19-Byte Handshake Routine............................ 23

1. INTRODUCTION

The Disk Jockey/Direct Memory Access (DJDMA) Floppy Disk Controller is a single board S-1ØØ subsystem. It communicates with both 8 inch and 5 1/4 inch floppy disk drives. Up to eight drives may be connected to the controller - with the limitation that no more than four of each type can be accommodated.

Special programmable bipolar LSI logic makes it possible to read and write media with almost any format, be it hard or soft sectored. Presently, the controller supports soft-sectored IBM compatible 8 inch media and hard-sectored North Star compatible 5 1/4 inch media. In the spring of 1982, IBM and Radio Shack 5 1/4 inch soft-sectored media will also be supported. Existing controllers in the field can be upgraded by replacing two of the ICs on the unit. This is done at moderate cost to the user.

The controller has its own Z-8Ø 4MHz microprocessor which is used to supervise data transfers between the disk drive and the system memory without intervention of the main CPU. This relieves the main CPU of time consuming processes which include head positioning, rotational delays, and the usual byte-by-byte transfer of data from the diskette to main memory. As a result, transfers are faster and more efficient. Moreover, the main CPU has more time for data processing, and thus, supports more users and/or tasks.

The main advantage of the DJDMA controller over almost all the others is its "glitch free" direct memory access channel. This advanced channel concept allows the controller to communicate with S-1ØØ memory by "stealing" bus cycles from the main CPU. This idea of an intelligent I/O channel was first implemented by IBM on their famous 37Ø mainframes. Now for the first time, this powerful concept has been implemented on the S1ØØ bus.

The channel has the full 24-bits of memory addressing as described in the proposed IEEE standard for the S-1ØØ bus. Also, a great deal of care has been taken in the design of the interface circuitry so it conforms in every detail to this new standard and still allows the controller to work well with existing systems designed before the standardization effort was started.

The controller is a temporary bus master, meaning that it has the same access to memory as the CPU whenever it has control. It also features priority logic which allows it to contend with up to sixteen other "temporary" masters that may also want to "steal" bus cycles from the main CPU, or the "permanent" master.

The controller acts as a temporary master (TMA). A temporary master may take control of the bus to perform a DMA operation. This is possible because both the TMA and the CPU drive control lines. The CPU, as permanent master, monitors signals from the TMA. When the TMA wants control, it first asserts a HOLD/ signal to the CPU. Assuming the TMA has priority, the CPU acknowledges this signal upon completion of the present bus cycle by returning a processor hold acknowledge (pHLDA) signal. Upon receipt of this signal, the TMA enables its control line and asserts a control disable (CDSB) signal, disabling the CPU's control line. The TMA then disables the CPU's data-out, address and status lines using DODSB/, ADSB/ and SDSB/ signals. At that point the TMA has complete control to perform its DMA operation.

To return control to the CPU, the TMA first disables its own data-out, address and status lines, then re-enables the CPU's control lines, and simultaneously, its data-out, address and status lines. The TMA then releases its control line and makes false the HOLD/ signal, thus returning full control to the CPU.

So far, the process has been described as if only one temporary master wanted control of the bus. There can be up to 16 temporary masters on the bus. When there is more than one temporary master, they use the four DMA lines to decide who gets to assert HOLD/. Any device requesting the bus places its TMA priority level on the bus, and circuitry on the device decides if it has the highest priority. The device with the highest priority (ØF hex is highest) asserts HOLD/. It removes its priority from the DMA lines when it receives pHLDA from the permanent master.

The features associated with the intelligent channel on the controller make it exceptionally desirable in multi-tasking and multi-user applications. In fact, many were tailored to enhance the performance of Morrow Designs new, powerful DECISION I multiprocessing IEEE 696/S-1ØØ machine. The DJDMA is an integral part of this advanced microcomputer system which incorporates many of the concepts originally introduced by IBM in their famous 37Ø series mainframes.

The DJDMA can boot itself up on the bus and even has a primitive serial port which is intended for diagnostic purposes or possibly even integrating the controller into a larger S-1ØØ system that has I/O that the boot disk is not aware of. **Under no circumstances** can it be used as a general purpose serial port to the system, however, since it is inactive during disk activity.

All in all, there is nothing on the market in the way of an S-1ØØ bus floppy disk controller that comes anywhere near the performance and versatility of the DJDMA. For that matter, we here at Morrow Designs know of no other floppy disk controller on **any** bus that can match the DJDMA in price, power, performance, and flexibility.

Good luck with this product. One of the purposes of this document is to detail how the DJDMA controller can improve the speed and performance of your system. If we've missed anything, please let us know.

2. PROGRAMMING SPECIFICATIONS

2.1. The Channel Concept

The IBM 370 mainframe was the first computer system to make use of the channel concept. In the traditional setting, an I/O controller, even one with direct memory access ability, was normally sent commands one at a time. Status was then reported through I/O ports after a command had completed.

One of the things a Direct Memory Access Controller does (and should do well) is communicate with main memory. Having realized this, someone very clever at IBM reasoned that if a controller could communicate with memory all that easily, why shouldn't it pick up its commands from memory as well? For that matter, why not have it lay down its status information in the CPU's main memory also?

Once the idea of picking up one command from memory is accepted, it is only a small step to think about placing strings of commands in memory and having the controller begin treating memory in the same way as the CPU does itself! That is, memory should be used for both instructions and data.

There is one detail missing in the above discussion. How is the controller to be started and stopped? A CPU starts running when power is turned on and continues (in theory) forever. But then there is the situation of a device whose primary job it is to transfer information to and from main memory and a mass storage device of some kind; it should remain idle until the CPU tells it otherwise.

A possible solution to the problem above is to have the device sample a memory location for a start command. At power-up, however, solid state memory does not have a predictable pattern. A start command could be present before it was actually issued by the CPU. The only foolproof way to issue a start command is through an I/O port. But doesn't that put us right back where we started? Actually, no.

It takes very little I/O circuitry to issue a simple pulse which can serve as a start command. It is also a small price to pay in cost and circuit board real estate for the flexibility and efficiency that is obtained.

Stop commands are much easier. Simply build an instruction into the controller's command set that forces it back to the idle state it was in just prior to the initial start pulse issued by the CPU.

Obviously, a channel type of controller needs some kind of on-board intelligence. At the time that IBM first built this kind of device, it was expensive both in terms of dollars and in circuit board real estate to implement this intelligence. Today however, the situation is quite different. Microprocessors are inexpensive and take only a modest amount of space on a circuit board.

In theory, the only limitation to the power and flexibility of a channel driven controller is the size of the memory local to the resident microprocessor. Since memory is getting denser and cheaper, it would seem that time will favor the channel approach to I/O controllers.

2.2. The Start Channel Command

Just as in the general case discussed above, there is a single primitive I/O port on the DJDMA. It resides at location EF (hex) unless a custom unit has been ordered with a special I/O address. This port's only purpose is to send start pulses to the DJDMA controller. **Any output instruction to port EF (hex) starts the DJDMA.** It doesn't matter what value is sent nor does it matter what kind of device sends the data. **Any time** any output reference is made to this port by the main CPU permanent master, or even by a temporary master, the DJDMA begins fetching and executing commands. Where these commands come from and how they work is taken up below.

2.3. The Channel Command Address

When the DJDMA first powers up or is reset, there is a three-byte pointer initialized in its local memory. This pointer determines where the controller picks up its first command when a start pulse is issued via I/O port EF (hex).

There are actually two of these three-byte values the DJDMA maintains. The first points to where it should start its command sequence. The second points to where it should get its next command in the event that the current one is not a halt command. The user needs to be aware of both of these pointers as he sets up command sequences for the controller to execute.

The second pointer has the same function as the program counter of the main CPU: it always points to the next command that the controller will execute. The first pointer is similar to the value forced into the program counter (PC) of the main CPU when a reset signal is issued. In most cases, a reset signal forces a Ø into the PC. The processor commences to fetch instructions at this value.

The same is true for the DJDMA, except that the value is not zero. Also, unlike the CPU, this initial location can be changed by a sending the proper command to the controller. **The initial location that the DJDMA controller begins fetching commands from is 5Ø (hex).** The command that alters this starting location is described in the next section.

2.4. Command Structure

Commands to the DJDMA controller are at least two bytes long. The first byte is always the command code. Parameter lists follow the command byte (if needed) and the command status byte (if needed) comes at the end of the command string. The length of a command string varies with the command. Unless a branch in channel command is issued, commands must be arranged in memory one after the other with no gaps between the end of one command and the beginning of another. Sequences of commands must be terminated with either a controller halt command or a branch in channel command. If a sequence ends with a branch in channel command, another sequence of commands must be present at the location specified in the address parameter list of the branch in channel command.

2.5. DJDMA Controller Commands

The Disk Jockey DMA controller recognizes the following commands:

- SET DMA ADDRESS
- READ A SECTOR
- WRITE A SECTOR
- SENSE DRIVE STATUS
- SET INTERRUPT REQUEST
- SET ERROR RETRY COUNT
- READ TRACK
- WRITE TRACK
- OUTPUT SERIAL PORT
- SERIAL INPUT ENABLE/DISABLE
- CONTROLLER HALT
- BRANCH IN CHANNEL
- SET CHANNEL ADDRESS
- SET TRACK SIZE
- SET DRIVE DESELECT/HEAD UNLOAD TIMEOUT
- SET LOGICAL DRIVE
- READ CONTROLLER MEMORY
- WRITE CONTROLLER MEMORY
- BRANCH TO CONTROLLER ROUTINE

The last three commands require great care to use. They are used to format diskettes and will be used to support media formats which are not yet implemented. Improper use of any of the last three commands could produce unpredictable results and may cause the loss of information on write-enabled diskettes in drives connected to the controller. It could also cause the controller to be inoperative until a bus reset is performed.

Morrow Designs will have a separate document (at extra cost) that describes the firmware on the DJDMA controller. This information should be available at the end of first quarter 1982 or early second quarter. Thus, users with special applications will have a way to extend the command structure of the DJDMA controller. However, extended commands **will not be supported** by Morrow Designs and we cannot stress too strongly that efforts in this direction will require a great deal time and expertise to complete and debug.

2.6. Controller Command Specifications

Specifications for each of the controller commands are described in the following sections. In many instances, examples are given to fully illustrate use of the command.

2.6.1. SET DMA ADDRESS

Command code:	23 (hex)
Command length:	4 bytes
Command parameter list length:	3 bytes
Command status list length:	Ø bytes

The command length is four bytes. The first byte is the command code: 23 (hex). The next three bytes specify a 24-bit address in main memory where data is written to or read from during subsequent disk transfers. This field must be arranged so that the least significant byte of the address directly follows the command byte. The byte of next highest significance follows. The highest order byte of the address is last. The last byte specifies an extended page as defined in the proposed IEEE standard for the S-1ØØ bus and allows memory addressing to be extended to 16 million bytes.

In systems that do not support this new extended addressing, the value of this high order byte is not important. However, it **must** be present - whether it is used or not. Other commands which have three byte address fields in their parameter list require the same byte significance order as described above. The firmware that processes commands on the DJDMA expects all address fields to be three bytes long - even if only two of the three have effect on the address bus of the system.

The following example is a command that sets the DMA address of the controller to location 8Ø (hex) - the default disk data buffer of the popular CP/M operating system:

23 8Ø ØØ ØØ (hex).

2.6.2. READ SECTOR

Command code:	2Ø (hex)
Command length:	5 bytes
Command parameter list length:	3 bytes
Command status list length:	1 byte

The three-byte parameter field following the command code consists of

1. track
2. side/sector
3. drive

in that order. The side select is encoded in the high order bit of the sector field and merged together to form the second byte in the parameter list. The third byte determines which of eight possible drives are read. If the system has been booted up from a 5 1/4 inch drive, drives Ø through 3 specify this; drives 4 through 7 specify 8 inch drives. If the system has been booted from an 8 inch drive, the numbering is reversed with the first four being 8 inch drives and the last four being 5 1/4 inch. The following example is a command that reads data from sector 3 of track 5 on side 1 of drive Ø:

2Ø Ø5 83 ØØ ØØ

The last zero is provided so that the controller can fill in the status of the transfer after it has completed the read. Here is a second example that reads sector 2 from track 6 on side Ø of drive 1:

2Ø Ø6 Ø2 Ø1 ØØ

Again, the last byte is for status reporting **and it must be there.**

The length of the sector (and consequently a valid range of sector values) depends on what size drive is being addressed and how the media has been formatted. In the media currently supported, the following sector values and data field lengths are relevant:

5 1/4" hard sectored single density:	Ø - 9	256 bytes
5 1/4" hard sectored double density:	Ø - 9	512 bytes
8" soft sectored single density:	1 - 26	128 bytes
8" soft sectored double density:	1 - 26	256 bytes
8" soft sectored double density:	1 - 15	512 bytes
8" soft sectored double density:	1 - 8	1Ø24 bytes

The numbers in the above list are all decimal. The sector size, density, and valid range of values for the sector number are all determined automatically by the controller. The controller can inform the system of these parameters by executing the SENSE DRIVE STATUS command which is taken up below. These details are presented here because it is necessary to know how much space the controller will use when data is read from the disk into main memory. Also, an error occurs if incorrect values are specified for the sector, track, or drive.

All 8 inch drives presently have 77 tracks numbered 0 through 76. This is not the case with 5 1/4 inch drives. Some have 35 tracks numbered 0 through 34, others have 40 tracks numbered 0 through 39, and finally, the new double track density 5 1/4 inch drives have 80 tracks numbered 0 through 79. The default value for 5 1/4 inch drives on the DJDMA is 40. However, this value can be changed by executing a SET TRACK SIZE command which is discussed below.

The last byte in the read sector command is called the status byte. This byte should be filled with some value other than what the controller might use when it reports status after the command is completed. A 0 is ideal since the controller does not use this value. For that matter, it does not use FF either. Either of these values are handy since they can be tested easily. By testing the status byte, the system can determine when a read command (among others) has completed. Below is a list of status byte codes along with their meanings. All values are in hex.

Table 2-1. Status Byte Codes

40 -	normal completion - no errors
80 -	improper command code
81 -	illegal disk drive value
82 -	drive not ready
83 -	illegal track value
84 -	unreadable media
85 -	improper sector header - no sync byte
86 -	CRC error in sector header read
87 -	seek error
88-8D -	compare error in sector header scan
8E -	CRC error in data field
8F -	illegal sector value for current media
90 -	media is write protected (writing only)
91 -	lost data - DMA channel did not respond
92 -	lost command - channel did not respond

The above list is complete and applies to any command that that reports status in its last byte. Not all codes apply to all commands. For example, 90 (hex) never appears as the status reported by the READ SECTOR command.

2.6.3. WRITE SECTOR

Command code:	21 (hex)
Command length:	5 bytes
Command parameter list length:	3 bytes
Command status list length:	1 byte

The three-byte parameter field and the status byte have the same properties as those in the read sector command. All the items discussed in the read sector command apply to the write sector command with the exception that the write sector command can report a media write protect error (90 hex).

2.6.4. SENSE DRIVE STATUS

Command code:	22 (hex)
Command length:	6 bytes
Command parameter list length:	1 byte
Command status list length:	4 bytes

The single byte in the parameter list specifies a drive. Legal values range from 0 to 7. The last byte of the status list has codes which were listed above in the READ SECTOR command. The first three bytes of status are peculiar to a specific drive and are detailed below. However, **unless the last status byte contains a 40 (hex), the preceding three bytes do not accurately reflect the condition and characteristics of the drive whose status was supposed to be sensed.**

If any value other than 40 (hex) is present, nothing can be learned from the first three status bytes. When the final byte contains a 40 (hex), the first three describe characteristics and status concerning the drive specified in the parameter byte of the command.

Table 2-2. STATUS BYTE 1: Drive Characteristic Byte

Each bit in this byte describes a different characteristic of the drive specified in the parameter field of the command.

Bit Ø - Information internal to the controller.

Bit 1 - If the media is hard-sectored, this bit is a 1. When the media in the drive is soft-sectored this bit will be a Ø.

Bit 2 - If the drive is 5 1/4 inch, this bit is a 1. If the drive is 8 inch, the bit is a Ø.

Bit 3 - If the drive has a DC motor with an ON/OFF switch, this bit is a 1. If there is no ON/OFF switch, or if the drive motor is AC, this bit is a Ø.

Bit 4 - If the media in the drive is double density, this bit is a 1. It is Ø only if the media is single density.

Bit 5 - If this bit is a 1 there is no "drive ready" signal supplied by the drive. For drives with no "ready" signal, the DJDMA firmware tests for the presence of sector/index holes. If the drive has an active "ready" signal, this bit is a Ø.

Bit 6 - If there is no "head load" command line to the drive, the controller assumes that the head(s) are always loaded against the media and this bit is a 1. If there is a "head load" command line to the drive, this bit is a Ø.

Bit 7 - If the head(s) are currently loaded against the media, this bit is a 1. If the head(s) are not loaded, this bit is a Ø.

Table 2-3. STATUS BYTE 2: Sector Length Code - Ø, 1, 2, or 3

The Ø indicates a sector length of 128 bytes, 1 stands for a length of 256 bytes, 2 means that the length is 512 bytes, and 3 indicates that the sector is 1Ø24 bytes long. These are all decimal numbers.

Table 2-4. STATUS BYTE 3: Drive Status/Characteristic Byte

There is an input port on the controller which can examine status signals transmitted directly from the selected drive.

The third status byte is a direct image of this port.

Bit Ø - Used internally by the controller and is of no meaning to the system.

Bit 1 - Current status of the serial input line from an RS-232 device which may be attached to connector P3, the serial port of the controller.

Bit 2 - This bit indicates that a double-sided 8 inch drive is currently selected and that double-sided media is present in the drive. This line is not driven by 5 1/4 inch drives; thus, an indirect means must be employed to determine if a 5 1/4 inch drive is double-sided and has double-sided media in it.

Bit 3 - Currently not used.

Bit 4 - This is the index/sector hole indicator. If this bit is a 1, the drive has sensed the presence of either an index hole or a sector hole.

Bit 5 - If this bit is a 1, the head(s) of the drive are at Track Ø. If the head(s) are positioned over some other track, this bit is a Ø.

Bit 6 - This bit is a 1 if the media in the drive is write protected. A zero indicates that the media is not write protected and disk write commands do not produce "write protect" errors.

Bit 7 - This is the drive ready bit. Most 5 1/4 inch drives have no signal on this line; thus, it is not a good "drive ready" indicator in this case.

All 8 inch drives produce a "ready" signal at this bit. If the current drive is an 8 inch and this bit is 1, the drive is "ready" to accept read, write, or step commands. If it is a Ø, the 8 inch drive is not "ready" and will not respond to commands from the controller.

2.6.5. SET INTERRUPT REQUEST

Command code:	24 (hex)
Command length:	2 bytes
Command parameter list length:	Ø bytes
Command status list length:	1 byte

This command generates an interrupt to the system bus. There is a bus driver on the DJDMA circuit board whose output terminates at a jumper pad near the lower edge of the board (the exact location is described later in the manual). This jumper pad is arranged so that the driver can be connected to the main interrupt line of the system bus (PINT*) or any one of the eight vectored interrupt lines (VIØ*, VI1*, ... VI7*).

The controller is shipped from the factory with the driver uncommitted. If the DJDMA is to generate interrupts to the system, this driver must be connected to one of the nine interrupt lines. If the driver is not connected, the INTERRUPT REQUEST command causes the controller to pause until another start pulse is issued by the system. However, once an INTERRUPT REQUEST command is executed, the controller is put into a special state where the board responds differently to the start pulse than it usually does.

Normally a start pulse causes the controller to begin fetching commands at the location specified by the most recent channel command word address. When the DJDMA executes an INTERRUPT REQUEST, it activates the interrupt bus driver on the circuit board. It then pauses with this bus driver still active.

Upon receipt of the next start pulse, the controller turns off the bus driver generating the interrupt and fetches the command **which immediately follows the interrupt request command.** The controller thus treats the first start pulse issued after the interrupt request command has completed as an **INTERRUPT ACKNOWLEDGE handshake signal.** This is the only circumstance in which a start pulse to the controller does not cause the command pointer to be reset.

The system can test the status byte following the command code to determine when the command has completed. When the command completes, it fills the status byte with a 4Ø (hex). When the interrupt request bus driver is not connected, an interrupt request command causes the controller to pause until the next start pulse is received, at which time it resumes executing commands where it left off.

2.6.6. SET ERROR RETRY COUNT

Command Code:	28 (hex)
Command length:	2 bytes
Command parameter list length:	1 byte
Command status list length:	Ø bytes

This command specifies how many times a sector is read in the event that a CRC error occurs in the data field. At least one read always takes place, so the smallest value that should appear in the parameter byte is a 1. This value can be as high as 255 (decimal). The default value is 1Ø (decimal).

This command's main purpose is to ensure that the value can be made smaller for diagnostic purposes. It is also useful when a diskette becomes worn and data recovery becomes more difficult. In this case, the value is made larger.

2.6.7. SET LOGICAL DRIVE

Command code:	2E (hex)
Command length:	3 bytes
Command parameter list length:	1 byte
Command status list length:	1 byte

This command allows the user to change the logical numbering assigned to the 8 inch and 5 1/4 inch drives. The default values assigned the the 8 inch drives are Ø through 3, while the 5 1/4 inch drives are assigned values 4 through 7.

If a 4 appears in the parameter list of this command, the 5 1/4 inch drives are assigned drive values Ø through 3, while the 8 inch drives have their values changed to 4 through 7. A Ø in the parameter field reverses these values to the original default values. There is no status byte associated with this command and bit-2 in the parameter field is the only part of the byte examined by the command.

The status byte reported by the command reflects the logical value of the first physical 8 inch drive prior to the execution of the SET LOGICAL DRIVE command. If the status is 4Ø (hex), the previous logical value of the first physical 8 inch drive was Ø. If the status is 44 (hex), the old value was 4.

The logical values assigned to the drives are also affected by performing a bootstrap operation which is discussed later.

2.6.8. SET HEAD UNLOAD/DRIVE DESELECT TIMEOUT

Command Code:	2F (hex)
Command length:	2 bytes
Command parameter list length:	1 byte
Command status list length:	Ø bytes

In order to conserve power and maximize diskette life, during periods of disk inactivity the controller unloads the drive head(s) and deselects the drive after a certain number of revolutions of the diskette. Normally, the controller waits sixteen revolutions before it deselects a drive. This command allows the user to change this situation. The value in the parameter list determines how many revolutions occur after no disk activity before the head(s) are unloaded and the drive is deselected. A disk transfer operation requires more time if the drive is not selected and so, under certain conditions, it may be desirable to extend the time before a drive is deselected after a transfer occurs. This command makes it possible to affect this situation. The value in the parameter field should be between 1 and 255 (decimal). However, when the heads are loaded for extended periods of time with the motor running, diskette media life is shortened considerably.

2.6.9 READ TRACK

Command code:	29 (hex)
Command length:	8 bytes
Command parameter list length:	6 bytes
Command status list length:	1 byte

This command reads an entire track into main memory starting at the value specified by the most recent SET DMA ADDRESS command. The transfer begins with the first full sector encountered by the controller. Thus, the buffer may not fill from the beginning.

As an example, suppose that the diskette had eight 1Ø24 byte sectors and the first full sector of data encountered was Sector 6. In this case the last 3Ø72 bytes of the buffer would be filled with Sectors 6, 7, and 8. The DJDMA memory pointer would then be reset to the start of the track buffer and Sectors 1 through 5 would be transferred.

The first three bytes of the parameter list specify

1. track
2. side
3. drive

in that order. The side bit must appear in the most significant bit of the byte. Thus, the second byte in the parameter list is either Ø or 8Ø (hex). The last three bytes of the parameter list form a memory pointer to a sector table.

There must be an entry in this table for each sector on the track.

As an example, if the diskette in the selected drive had 512 byte sectors, there would be fifteen entries and the table length would also be fifteen. This table should be initialized with Øs, 8Øs (hex), or FFs (hex).

As a sector of the track is read, the controller fills the byte of the table corresponding to the sector with status information concerning that particular sector (assuming the initial entry was Ø). Thus, the system can determine error information individually, sector by sector.

If the controller encounters an FF (hex) entry in the sector table, it skips that sector which corresponds to the entry.

If a whole section of the table has FFs, the sectors corresponding to this section are not read.

If the controller encounters an entry in the table of 8Ø (hex), the READ TRACK command terminates at that point. An example should illustrate these ideas.

Suppose side 1 of track 23 (decimal) is to be read into a track buffer starting at location ØØEØØØ (hex) from drive 2 and that a set DMA address command with this value has already been executed. Suppose also that there are 1Ø24 byte sectors on the diskette and that the sector table is to immediately precede the track buffer in memory. The command to read the track would then appear as follows:

```
29 17 8Ø Ø2 F8 DF ØØ ØØ
```

The sector table address of ØØDFF8 (hex) has a value of eight less than ØØEØØØ (hex) since there are eight sectors on the track of the diskette. The last byte (indicated with a value of ØØ) is the overall status byte for the command. The status codes are the same as the READ SECTOR COMMAND where they are listed.

2.6.1Ø. WRITE TRACK

Command Code:	2A (hex)
Command length:	8 bytes
Command parameter list length:	6 bytes
Command status list length:	1 byte

The write track command is similar to the READ TRACK command. The six bytes of the parameter list are exactly the same and even the sector table entries work the same. Normally, the table has Øs as entries. Sectors that are not to be written (or rewritten) are marked with FFs (hex) while an 8Ø (hex) causes the command to terminate.

As with the read track command, the starting address of the track buffer is initialized with a SET DMA ADDRESS command.

2.6.11. OUTPUT TO SERIAL PORT

Command code:	2B (hex)
Command length:	3 bytes
Command parameter list length:	1 byte
Command status list length:	1 byte

This command communicates with the output portion of the bit serial port on the DJDMA. The parameter byte is filled with the ASCII value that is to be transmitted to the RS-232 device connected to the port. The status byte should be initialized to either Ø or FF (hex). The command fills the status byte with a 4Ø (hex) when all eight data bits and two stop bits have been transmitted.

The speed of this serial port is 96ØØ baud and cannot be changed. Also, it is vital that the system refrain from sending new start pulses to the controller until this command has completed. Otherwise, transmission of the serial stream is aborted before any or all of the bits have been sent.

The main purpose of the port in this subsystem is to allow a user to boot-up in a system where I/O devices are not defined on the boot diskette. This port is not adequate as a system consul port and will cause the controller to run less efficiently while the port is active (there is no disk activity while the serial port is engaged in data transmission). Input serial data can also be easily lost if the controller is supervising data transfer to or from a disk drive.

The input side of this serial port does not work the same as the output and is discussed in the next command.

2.6.12. SERIAL INPUT ENABLE/DISABLE

Command Code:	2C (hex)
Command length:	2 bytes
Command parameter list length:	1 byte
Command status list length:	Ø bytes

This command enables or disables input from the bit serial RS-232 port on the controller. Serial input operates in a slightly different manner than serial output. If the input side of the port is enabled, characters received by the port are deposited at location ØØØØ3E (hex).

After loading a new character at this location, the controller writes 4Ø (hex) at location ØØØØ3F (hex). This second location serves as a status flag for serial input and should be reset to some other value after reading the character.

In the enable/disable command, the value of the parameter byte determines whether the port is to be enabled or disabled. A Ø in this byte instructs the controller to turn off the port, while a 1 forces the DJDMA to enable input. At boot-up, input is enabled, but if there is no terminal connected to the board, it is automatically disabled.

2.6.13. CONTROLLER HALT

Command code:	25 (hex)
Command length:	2 bytes
Command parameter list length:	Ø bytes
Command status list length:	1 byte

This command is used to halt the DJDMA controller. There are no parameters. The status byte should be initialized to Ø or FF (hex). The controller fills this byte with a 4Ø (hex) when the command completes. As mentioned previously, this command resets the command pointer. Hence, the next start pulse causes the controller to begin fetching commands from the channel command word address which has an initial value of ØØØØ5Ø (hex). This value can be changed with a command that is described below.

2.6.14. BRANCH IN CHANNEL

Command code:	26 (hex)
Command length:	4 bytes
Command parameter list length:	3 bytes
Command status list length:	Ø bytes

The three parameter bytes specify a branch address for the controller. This address is the location from where the controller fetches its next command. The address bytes are arranged so that the low order byte immediately follows the command code, the middle order byte is next and the high order byte is last. There is no status code and immediately after execution, the controller picks up the next command from the branch address.

2.6.15. SET CHANNEL ADDRESS

Command code:	27 (hex)
Command length:	4 bytes
Command parameter list length:	3 bytes
Command status list length:	Ø bytes

The three parameter bytes of this command specify a memory address. After this command has executed, start pulses from the system cause the controller to fetch its first instruction at this address. The order of the bytes is the same as the branch in channel command. There is no status byte associated with this command.

2.6.16. SET TRACK SIZE

Command Code:	2D (hex)
Command length:	4 bytes
Command parameter list length:	2 bytes
Command status list length:	1 byte

This command allows the system to change the number of tracks that the controller assumes are on a disk drive. The first byte in the parameter list describes a drive and should have values between 0 and 7. Other values cause the command to return an error and not change the track value of any drive.

The second byte must contain a hex number which is **one larger** than the largest numerical track on the diskette. For 35 track drives, this value is 35 since the track numbering starts at zero. For the same reason, the value is 40 for 40 track drives, 77 for 77 track drives, and 80 for 80 track drives. (All the numbers used in this paragraph are decimal. They must be changed to hexadecimal when incorporated into the command string.)

It is possible to damage a drive if seeks are performed to tracks which extend beyond the boundaries of the seek mechanism. The controller has no way to determine if a particular value is improper for a given drive. **The user must exercise care in executing this command and Morrow Designs takes no responsibility for damage that occurs through its misuse.**

2.6.17. READ CONTROLLER MEMORY

Command Code:	A0 (hex)
Command length:	8 bytes
Command parameter list length:	7 bytes
Command status list length:	0 bytes

The first three bytes of the parameter list specify a main memory address with bytes in ascending order (just like the other commands that required a three-byte address field.)

The next two bytes specify a count which can have values anywhere between 0 and FFFF (hex). The last two bytes specify an address in the memory of the on-board Z-80A microprocessor. This command transfers local memory to main memory which allows the main CPU to read the controller's memory. It is not advisable to read locations 4001 (hex), 8001 (hex), A000 (hex), etc., since this type of reference causes the controller to hang waiting for data from a drive when none is selected. The only way to reliably recover from this fault is to issue a reset to the system. **Morrow Designs does not recommend use this command and does not support applications that make use of this command or the two that follow.** This command reports no status.

2.6.18. WRITE CONTROLLER MEMORY

Command Code:	A1 (hex)
Command length:	8 bytes
Command parameter list length:	7 bytes
Command status list length:	Ø bytes

The first three bytes of the parameter list specify a main memory address in ascending order (just like the other commands that required a three-byte address field.)

The next two specify a count that can range between Ø and FFFF (hex).

The last two bytes specify an address in the memory space of the on-board Z-8ØA microprocessor. This command transfers data from main memory to the memory of the controller. There are only 1Ø24 bytes of RAM on the controller board. This RAM starts at location 1ØØØ (hex). The only locations safe to write in are between 1Ø3Ø and 127F (hex). Writing in other locations produces unpredictable results and can lead to loss of data on diskettes which are not write protected and are inserted in drives connected to the controller. **Morrow Designs does not support the use of this command. This command is used in diskette format programs (included in this manual) but we strongly recommend that it not be used for other purposes).** There is no status byte associated with this command.

2.6.19. EXECUTE CONTROLLER ROUTINE

Command Code:	A2 (hex)
Command length:	3+ bytes
Command parameter list length:	2 bytes
Command status list length:	Ø+ bytes

The two bytes in the parameter list specify an address in the memory space of the on-board Z-8ØA microprocessor. This command forces the on-board processor to branch to and begin executing instructions at this address. As with the previous command, it is extremely dangerous and should not be used by anyone except those well versed with the inner workings of the controller. The status list length is given as Ø+ bytes because the length and type of status varies depending on the nature of the routine at the specified address. **As with the previous two commands, Morrow Designs does not support use of this command.**

2.7. Command Summary

The following tables summarize commands that are both supported and unsupported by the DJDMA.

Table 2-5. Supported Commands

- Set DMA (low, med, high)
- Read Sector (track, side/sector, drive, status)
- Write Sector (track, side/sector, drive, status)
- Sense Status (dstatl, dstat2, dstat3, status)
- Set Interrupt Request (status)
- Set Error Retry Count (count)
- Set Logical Drive (drive, type)
- Set Head Unload/Drive Deselect Timeout (revolution count)
- Read Track (track, side, drive, low, med, high, status)
- Write Track (track, side, drive, low, med, high, status)
- Serial Port Output (ASCII byte)
- Serial Input Enable/disable (control byte)
- Controller Halt (status)
- Branch in Channel (low, med, high)
- Set Channel Address (low, med, high)
- Set Track Size (drive, hitrack)

Table 2-6. Unsupported Commands

- Read CMemory (tlow, tmed, thigh, lcnt, hcnt, slow, shigh)
- Write CMemory (slow, smed, shigh, lcnt, hcnt, tlow, thigh)
- Execute Controller Routine (low, high, ..., ...)

2.8. Status Codes

The following table summarizes the DJDMA status codes.

Table 2-7. Status Code Summary

STATUS CODE	DESCRIPTION
4Ø	Normal completion - no error encountered
8Ø	Improper Command Code
81	Improper Disk Drive Value
82	Disk Drive Not Ready
83	Improper Track Value
84	Unreadable Media
85	Improper Sector Header - No Sync Byte(s)
86	CRC Error in Sector Header Scan
87	Seek Error
88 - 8D	Compare Error in Sector Header Scan
8E	CRC Error in Data Field
8F	Improper Sector Value
9Ø	Media Write Protected
91	Lost Data - DMA Channel did not respond
92	Lost Command - Channel did not respond

3. IEEE 696 (S-1ØØ) BUS CONSIDERATIONS

The DJDMA controller has been designed to meet the IEEE/696 proposed standard for the S-1ØØ bus and will operate properly in any S-1ØØ mainframe which meets this proposed standard and can accommodate temporary bus masters. In fact, the DJDMA runs in most existing S-1ØØ systems in operation today. However, we cannot guarantee that the controller will operate in a system unless it meets **all the specifications contained in the IEEE/696 document.**

In transferring data from a floppy disk directly into main memory, the DJDMA assumes that the permanent master in the system will respond to bus requests by the controller fast enough so that data will not be lost. If an 8 inch double density drive is connected to the controller, a byte of data is read or written every 16 microseconds.

The transfer rate for single density 8 inch drives and double density 5 1/4 inch drives is a byte every 32 microseconds.

Single density 5 1/4 inch drives have a transfer rate of one byte every 64 microseconds. If some device, such as a front panel, holds the READY line of the bus down for extended periods during disk transfers, data is lost and the controller cannot function properly.

Morrow Designs assumes that the user has made the proper determination concerning the ability of his system to respond to bus requests from the DJDMA so that data is not lost during disk transfers. **Morrow Designs is not responsible for operation of the controller in systems that cannot respond to bus requests at least as fast as those detailed above for the various types of floppy disk drives.**

4. INTERRUPTS

At the lower left area of the DJDMA circuit board, just above the edge connector fingers, is a jumper area designed so users can connect the board's interrupt request bus driver to one of the nine interrupt request lines: VIØ*, VI1*, VI2*, VI3*, VI4*, VI5*, VI6*, VI7*, or PINT* (See the component layout for an illustration of this area).

If the system does not use interrupts, there is no need to connect J3 to any of these lines. If J3 is not jumpered, it appears to the system that the controller has entered a pause state when it executes an interrupt request command. All activity stops (just as it does after a halt command). When the next start pulse is sent to the controller, it picks up its next instruction from the memory location immediately following the status byte of the interrupt request command (this is not the same as a halt command).

The DJDMA is shipped from the factory without any jumpering between J3 and the interrupt request lines. If the controller is to generate interrupt requests, the user must determine which of the nine possible connections is appropriate for his system. The DECISION I user reference manuals contain information about how the DJDMA communicates with the interrupt controller on the MULT-I/O and WUNDERBUSS I/O boards, and should serve as an example of how interrupts from the DJDMA could work in other systems.

5. I/O CONNECTORS

Refer to the component layout drawing included in this manual for a more complete understanding of the discussion in this section.

There are three I/O connectors at the top of the DJDMA circuit board: P1, P2, and P3.

P3 is at the top left-hand side of the board and is the connector for the bit serial RS-232 port. It has three pins, numbered 1 through 3 from left to right. Pin-1 is the RS-232 ground signal, pin-2 is the input and pin-3 is the RS-232 output signal.

To the right of P3 is P2. P2 has 34 pins and is used to connect 5 1/4 inch drives to the controller. The pins are arranged in two rows - the odd numbered pins being just above the even numbered ones. The pins are numbered 1 through 33, odd from right to left, and 2 through 34, even from right to left. All the odd numbered pins are connected to ground while the even numbered pins carry information to and from 5 1/4 inch floppy disk drives.

P1 is the right-most connector and has 5Ø pins. This connector is used to connect 8 inch drives to the controller and has pins arranged in two rows, the same as P2. The upper pins are odd and are numbered 1 through 49, right to left. The lower pins are even and are numbered 2 to 5Ø, right to left. As before, all odd pins are grounds while even pins carry signals between the controller and 8 inch drives.

6. JUMPERED SETTINGS

Refer to the component layout drawing included in this manual for a more complete understanding of the discussion in this section.

6.1. EPROM Replacement

The jumpered setting at J1 (located in the upper right hand corner of the board) is factory set B to C for a 2732 EPROM. It may be jumpered A to B, effectively replacing it with a 2716 EPROM. But please note that the **factory setting must be maintained** for proper system operation. The optional setting reduces the address space available and is only to be used in special, limited applications.

6.2. Bootstrap Program

J2 (located in the lower mid-section of the board) is jumpered B to C for conditional bootstrap operation. This mode is used for the Decision I and controllers are shipped from the factory with a jumper between these two pins.

J2 is jumpered A to B for non-bootstrap mode in systems which cannot allow a temporary master to hog the bus and intend to boot the DJDMA controller by external means.

7. BOOTSTRAP LOAD

The DJDMA performs an automatic bootstrap load at reset or power-on if J2 is jumpered B to C and a shunt jumper is placed between pins 1 and 2 of P3, or if a terminal is connected to P3. In either case, the controller halts the main CPU by taking control of the bus and reads the first 38 (hex) locations in main memory into its own local memory. Next it loads 0s into these first 38 (hex) bytes and places a short, 19 byte (decimal) handshake routine between 000038 and 00004A (hex). The bus is then released. When the main CPU executes the first part of the handshake routine, the controller restores the first 38 (hex) locations of main memory to its original state. Next, 80 (hex) bytes are loaded between 000080 and 0000FF (hex) from the first sector on Track 0 of the disk. Finally, the controller writes a control byte to the handshake routine which causes the main CPU to branch to location 000080 (hex). A listing of the 19-byte handshake routine is given below.

Table 7-1. 19-Byte Handshake Routine

```
000038    21 4A 00       START:    LXI   H,4A
00003B    36 00                    MVI   M,0
00003D    7E             LOOP:     MOV   A,M
00003E    B7                       ORA   A
00003F    CA 3D 00                 JZ    LOOP
000042    FE 40                    CPI   40H
000044    C2 3D 00                 JNZ   LOOP
000047    C3 80 00                 JMP   80H
00004A    FF                       DB    0FFH
```

The controller will boot from either the first drive connected to the 8 inch port or the first drive connected to the 5 1/4 inch port. The decision as to which port to choose is determined by testing for a "drive ready" signal. The 8 inch port is tested first. The controller will alternately continue to test for "drive ready" indefinitely to allow the user time to insert a diskette. This is evidenced by the indicator lights on the disk drives. They will alternately blink as the controller checks for the ready signal.

8. BOOTING THE DJDMA

The following is the proper procedure for booting the DJDMA:

1. Open the door of any drive the DJDMA could boot from.
2. Insert a bootstrap diskette in the boot drive WITHOUT closing the driver door.
3. Depress the RESET switch.
4. While the RESET switch is depressed, close the drive door.
5. Release the RESET switch.

It is possible that the above procedure will have to be repeated twice depending on the value of location Ø.

If a shunt jumper across pins 2 and 3 of P3 is not in place or if a terminal is not connected to P3, the controller powers itself up in normal "cycle steal" mode and waits for commands from the system.

9. FORMATTING DISKETTES

There are no firmware commands on the DJDMA to format diskettes for two reasons: **Formatting is a dangerous operation.** If a diskette is in a drive with valuable information written on it, an accidental format command could destroy this data. The controller is also capable of formatting a wide variety of diskettes and the EPROM is not large enough to accommodate both the command processor code and all of the desirable format routines.

For these reasons, the format routines are loaded from main memory using the WRITE CONTROLLER MEMORY command and executed using the EXECUTE CONTROLLER ROUTINE command. A listing of two format programs for IBM soft-sectored 8 inch diskettes and North Star hard-sectored 5 1/4 inch diskettes appears as an appendix to this manual. These programs are also available on diskettes for a modest cost for those who wish to avoid using controller commands not supported in the field.

When a CP/M operating system is shipped with either a lone DJDMA controller or a disk system which includes a DJDMA controller, there are built-in commands on the system diskette which will format both types of diskettes.

Parts List

Amount	Function	Description
1	PC board	DJDMA
5	Diode	1N914
1	Transistor	2N39Ø4
6	Transistor	2N39Ø6
2	Regulator	+5 volts
1	Regulator	+12 volts
1	Regulator	-12 volts
1	Resistor	1K Ohm 1/4W 5%
2	Resistor	1 Meg Ohm 1/4W 5%
1	Resistor	12K Ohm 1/4W 5%
1	Resistor	1.2K Ohm 1/4W 5%
1	Resistor	1.5K Ohm 1/4W 5%
1	Resistor	18Ø Ohm 1/4W 5%
2	Resistor	27K Ohm 1/4W 5%
4	Resistor	33Ø Ohm 1/4W 5%
11	Resistor	3.3K Ohm 1/4W 5%
1	Resistor	39Ø Ohm 1/4W 5%
3	Resistor	4.7K Ohm 1/4W 5%
1	Resistor	47K Ohm 1/4W 5%
1	Resistor	2.ØK Ohm 1/4W 1%
1	Resistor	2Ø.ØK Ohm 1/4W 1%
1	Resistor	28.ØK Ohm 1/4W 1%
1	SIP	18ØK 1/8W 5% (1Ø-pin)
1	SIP	3.3K 1/8W 5% (8-pin
1	Inductor	4.7uh
1	Capacitor	.ØØ1mf ceramic disk
13	Capacitor	.1uf mono cap
1	Capacitor	.Ø1 mylar cap
1	Capacitor	33pf silver/mica
2	Capacitor	47pf silver/mica
2	Capacitor	1ØØpf silver/mica
1	Capacitor	12ØØpf silver/mica
1	Capacitor	62Ø pf silver/mica
8	Capacitor	1uf dip. tant.
1	Crystal	4 MHz
1	PCB Header	SIN RT> NHD 3
1	PCB Header	DIN RT> HD 34
1	PCB Header	DIN RT> HD 5Ø
2	Slide Jumpers	
2	Screws	632 X 5/16 Pan Phil

Parts List, Cont.

2	Hex Nuts	632
2	Heat Sinks	Low Profile 3 Fin
2	Heat Sinks	Slimline 5 prong
1	IC Socket	Low Profile (8-pin)
13	IC Sockets	Low Profile (14-pin)
12	IC Sockets	Low Profile (16-pin)
2	IC Sockets	Low Profile (18-pin)
15	IC Sockets	Low Profile (20-pin)
1	IC Socket	Low Profile (24-pin)
1	IC Socket	Low Profile (28-pin)
1	IC Socket	Low Profile (40-pin)
1	IC	1458
2	IC	2114-3 RAM
1	IC	7404
1	IC	7406
1	IC	74LS02
1	IC	74LS04
1	IC	74LS08
1	IC	74LS10
2	IC	74LS138
1	IC	74LS139
1	IC	74LS153
3	IC	74LS221
1	IC	74LS244
2	IC	74LS273
1	IC	74LS279
1	IC	74LS299
4	IC	74LS373
4	IC	74LS374
1	IC	74LS38
1	IC	74LS393
3	IC	74LS74
1	IC	74LS75
1	IC	81LS95
1	IC	81LS96
1	IC	PAL
1	IC	FPLA
5	IC	PROM

Subject Index

B
BRANCH IN CHANNEL, 17
Board compatibility, 1

C
CONTROLLER HALT, 17
CP/M data buffer, 6
Command Pointer reset, 17
Command parameter lists, 5
Command status byte, 5
Controller
   DMA channel, 1
   microprocessor, 1
   supervision of data transfer, 1
Cycle steal mode, 24

D
DJDMA self boot capability, 2
DMA communication with main memory, 3
Dangers of formatting diskettes, 24
Data recovery, 13
Data transfer, 21
Drive values, 13

E
EXECUTE CONTROLLER ROUTINE, 19
Extended addressing, 6

I
IEEE standards and board compatibilty, 1
Intelligent I/O channel, 1
Interrupt request lines, 12

J
Jumpering interrupt request lines, 22

L
Listing of DJDMA Controller Commands, 5
Listing of status byte codes, 8
Listing of valid sector values, 7

Subject Index

M

Master
   permanent, 1, 4
   temporary, 1, 4

O

OUTPUT TO SERIAL PORT, 16

P

Permanent master, 1, 4
Port enable and terminal connection, 17
Power-up or reset pointer, 4
Primitive I/O port - DJDMA, 4
Program Counter, 4

R

READ CONTROLLER MEMORY, 18
READ SECTOR, 7
READ TRACK, 14

S

Sector transfer sample, 14
SENSE DRIVE STATUS, 9
SERIAL INPUT ENABLE/DISABLE, 16
SET CHANNEL ADDRESS, 17
SET DMA ADDRESS, 6
SET ERROR RETRY COUNT, 13
SET HEAD UNLOAD/DRIVE DESELECT TIMEOUT, 14
SET INTERRUPT REQUEST, 12
SET LOGICAL DRIVE, 13
SET TRACK SIZE, 18
Serial port communication, 16
Start command, 3
Status flag for serial input, 16
Stealing bus cycles, 1
Stop command, 3

T

Temporary master, 1, 4
Track numbering, 8

U

Undefined I/O devices, 16

W

WRITE CONTROLLER MEMORY, 19
WRITE SECTOR, 9
WRITE TRACK, 15

Z

Z-8ØA - memory transfer, 19

SOFTWARE LISTING

```
 0000'   31 059E'         START:  LD      SP,ECODE+30H           ;initialize the stack pointer
 0003'   21 1030                  LD      HL,1030H               ;initalize command addresss
 0006'   22 0161'                 LD      (DOTCMD+1),HL
 0009'   21 113A                  LD      HL,SDADVT
 000C'   22 0167'                 LD      (ATCMD+1),HL
 000F'   21 016F'                 LD      HL,SMESSG              ;start of program message
 0012'   CD 011E'                 CALL    OUTM                   ;send the message
 0015'   CD 012A'                 CALL    INPUT                  ;get response to drive number
 0018'   D2 0024'                 JP      NC,DATAOK              ;test for valid input
 001B'   21 01BA'         DEXIT:  LD      HL,BMESSG              ;invalid input message
 001E'   CD 011E'                 CALL    OUTM                   ;send the message
 0021'   C3 0000'                 JP      START                  ;go back to start of program
 0024'   32 045D'         DATAOK: LD      (SINGLE+1),A           ;store the drive number in code
 0027'   21 01ED'                 LD      HL,DMESSG              ;type of density message
 002A'   CD 011E'                 CALL    OUTM                   ;send the message
 002D'   CD 012A'                 CALL    INPUT                  ;wait for response
 0030'   DA 001B'                 JP      C,DEXIT                ;test for improper input
 0033'   E6 01                    AND     1                      ;density encoded in bit 0
 0035'   32 032A'                 LD      (DENSTY),A             ;save for later use
 0038'   CA 0065'                 JP      Z,SIDE                 ;skip sector size if single density
 003B'   21 0225'                 LD      HL,LMESSG              ;sector length message
 003E'   CD 011E'                 CALL    OUTM                   ;send the message
 0041'   CD 012A'                 CALL    INPUT                  ;wait for input
 0044'   DA 001B'                 JP      C,DEXIT                ;test for improper input
 0047'   FE 03                    CP      3                      ;futher test for improper input
 0049'   CA 001B'                 JP      Z,DEXIT                ;error exit
 004C'   16 00                    LD      D,0                    ;form offset into sector table
 004E'   5F                       LD      E,A
 004F'   3C                       INC     A                      ;adjust for sector length code
 0050'   32 03C5'                 LD      (DLCODE-DDFMT+DOUBLE),A ;store in format code
 0053'   21 016C'                 LD      HL,STABLE
 0056'   19                       ADD     HL,DE
 0057'   7E                       LD      A,M                    ;fetch number of sectors
 0058'   32 0407'                 LD      (DLAST-DDFMT+DOUBLE),A ;store in format code
 005B'   3E 20                    LD      A,20H                  ;sector length code is 80,100, or 0
 005D'   87               DCNST:  ADD     A,A
 005E'   1D                       DEC     E                      ;decrement the sector type
 005F'   F2 005D'                 JP      P,DCNST                ;test for cycle done
 0062'   32 03EF'                 LD      (DSIZE-DDFMT+DOUBLE),A ;store 1/4 length in format code
 0065'   21 0265'         SIDE:   LD      HL,HMESSG              ;double sided media message
 0068'   CD 011E'                 CALL    OUTM                   ;send the message
 006B'   CD 012A'                 CALL    INPUT                  ;wait for input
 006E'   DA 001B'                 JP      C,DEXIT                ;test for improper input
 0071'   E6 01                    AND     1                      ;discard all but bit 0
 0073'   32 041C'                 LD      (DDSBIT-DDFMT+DOUBLE),A ;store in format code double density
 0076'   32 0532'                 LD      (SDSBIT-SDFMT+SINGLE),A ;store in format code single density
 0079'   21 0151'         LOADC:  LD      HL,LSDCMD              ;load single density code command
 007C'   06 0A                    LD      B,0AH                  ;command length
 007E'   CD 00FB'                 CALL    LCMD                   ;load the code
 0081'   21 0160'                 LD      HL,DOTCMD              ;format track 0 command
 0084'   06 06                    LD      B,6                    ;command length
 0086'   CD 00FB'                 CALL    LCMD                   ;execute the command
 0089'   CA 00A8'                 JP      Z,PROCED               ;zero => no error
 008C'   21 029A'                 LD      HL,RMESSG              ;drive not ready message
 008F'   FE 82                    CP      82H                    ;drive not ready error code
 0091'   CA 0097'                 JP      Z,$+6                  ;test for drive not ready
 0094'   21 02D6'                 LD      HL,WMESSG              ;drive must be write protected
```

```
ØØ97'   CD Ø11E'            CALL    OUTM            ;send the message
ØØ9A'   CD Ø12A'            CALL    INPUT           ;wait for input
ØØ9D'   DA ØØ1B'            JP      C,DEXIT         ;test for improper input
ØØAØ'   E6 Ø1               AND     1               ;discard all but bit Ø
ØØA2'   CA ØØØØ'            JP      Z,START         ;zero => start the program over
ØØA5'   C3 ØØ79'            JP      LOADC           ;go back and do the command over
ØØA8'   21 Ø327'   PROCED:  LD      HL,CRLF         ;carriage return and line feed
ØØAB'   CD Ø11E'            CALL    OUTM            ;output the string
ØØAE'   21 1Ø5Ø             LD      HL,SDRDY        ;adjusted execution address of format
ØØB1'   3A Ø32A'            LD      A,(DENSTY)
ØØB4'   B7                  OR      A               ;test for double density
ØØB5'   CA ØØC9'            JP      Z,CONTUE        ;make no adjustments for single density
ØØB8'   21 Ø147'            LD      HL,LDDCMD       ;load double density format command
ØØBB'   Ø6 ØA               LD      B,ØAH           ;command length
ØØBD'   CD ØØFB'            CALL    LCMD            ;load the code into controller
ØØCØ'   21 1159             LD      HL,DDADVT       ;advance track execute address
ØØC3'   22 Ø167'            LD      (ATCMD+1),HL    ;update the command execute address
ØØC6'   21 1Ø3Ø             LD      HL,1Ø3ØH        ;format execute address
ØØC9'   22 Ø161'   CONTUE:  LD      (DOTCMD+1),HL   ;update track format execute address
ØØCC'   3E 2A               LD      A,"*"           ;send a star for a track done
ØØCE'   CD Ø114'            CALL    OUTPUT
ØØD1'   21 Ø166'            LD      HL,ATCMD        ;advance track command
ØØD4'   Ø6 Ø6               LD      B,6             ;command length
ØØD6'   CD ØØFB'            CALL    LCMD            ;load the command and execute
ØØD9'   FE 4D               CP      4DH             ;last track value (77 decimal)
ØØDB'   C2 ØØE7'            JP      NZ,FMTRCK       ;zero => formatting done
ØØDE'   21 Ø312'   ENDFMT:  LD      HL,FMESSG       ;send final message
ØØE1'   CD Ø11E'            CALL    OUTM
ØØE4'   C3 ØØØØ'            JP      START           ;go format another disk
ØØE7'   21 Ø16Ø'   FMTRCK:  LD      HL,DOTCMD       ;format a track command
ØØEA'   Ø6 Ø6               LD      B,6             ;command length
ØØEC'   CD ØØFB'            CALL    LCMD            ;load and execute the command
ØØEF'   CA ØØCC'            JP      Z,CONTUE+3      ;loop back for more tracks
ØØF2'   21 Ø29A'            LD      HL,RMESSG       ;drive has become not ready
ØØF5'   CD Ø11E'            CALL    OUTM
ØØF8'   C3 ØØDE'            JP      ENDFMT          ;stop the formatting

ØØFB'   11 ØØ5Ø    LCMD:    LD      DE,5ØH          ;start of command sequence
ØØFE'   7E                  LD      A,M             ;get command data
ØØFF'   12                  LD      (DE),A          ;load into command area
Ø1ØØ'   23                  INC     HL              ;advance the pointers
Ø1Ø1'   13                  INC     DE
Ø1Ø2'   Ø5                  DEC     B
Ø1Ø3'   C2 ØØFE'            JP      NZ,LCMD+3       ;test for transfer done

Ø1Ø6'   D3 EF      ECMD:    OUT     (ØEFH),A        ;start the controller
Ø1Ø8'   1B                  DEC     DE              ;pointer for status byte of halt cmd
Ø1Ø9'   1A                  LD      A,(DE)
Ø1ØA'   B7                  OR      A               ;test for command string done
Ø1ØB'   CA Ø1Ø9'            JP      Z,ECMD+3
Ø1ØE'   3A ØØ53             LD      A,(53H)         ;status byte for execute command
Ø111'   FE 4Ø               CP      4ØH             ;test for no error
Ø113'   C9                  RET

Ø114'   21 Ø15C'   OUTPUT:  LD      HL,SOCMD+1      ;data byte of serial output command
Ø117'   Ø6 Ø5               LD      B,5             ;serial output command string length
Ø119'   77                  LD      M,A             ;store the data
Ø11A'   2B                  DEC     HL              ;back up to pointer
Ø11B'   C3 ØØFB'            JP      LCMD            ;load the command and execute
```

```
 Ø11E'    7E              OUTM:   LD      A,M             ;get current byte of message
 Ø11F'    B7                      OR      A               ;test for end of message
 Ø12Ø'    C8                      RET     Z               ;return at end of message
 Ø121'    E5                      PUSH    HL              ;save the character pointer
 Ø122'    CD Ø114'                CALL    OUTPUT          ;output the character
 Ø125'    E1                      POP     HL              ;recover the character pointer
 Ø126'    23                      INC     HL              ;advance the character pointer
 Ø127'    C3 Ø11E'                JP      OUTM            ;go get the next character

 Ø12A'    21 ØØ3F         INPUT:  LD      HL,3FH          ;serial input status byte
 Ø12D'    3E 4Ø                   LD      A,4ØH           ;test value for status
 Ø12F'    96                      SUB     M               ;test for character ready
 Ø13Ø'    C2 Ø12D'                JP      NZ,INPUT+3      ;zero => new character ready
 Ø133'    77                      LD      M,A             ;zero out the status byte
 Ø134'    2B                      DEC     HL              ;back up pointer to the character
 Ø135'    7E                      LD      A,M             ;pickup the character
 Ø136'    F5                      PUSH    AF              ;save the data
 Ø137'    CD Ø114'                CALL    OUTPUT          ;echo the data
 Ø13A'    F1                      POP     AF
 Ø13B'    E6 7F                   AND     7FH             ;turn it into ASCII
 Ø13D'    FE 3Ø                   CP      3ØH             ;test for smaller than zero
 Ø13F'    D8                      RET     C
 Ø14Ø'    FE 34                   CP      34H             ;test for larger than three
 Ø142'    3F                      CCF
 Ø143'    D8                      RET     C
 Ø144'    E6 Ø3                   AND     3               ;change ASCII to binary
 Ø146'    C9                      RET
                                  PAGE
```

```
  Ø147'   A1                LDDCMD: DB      ØA1H                ;write controller memory command
  Ø148'   Ø32B'                     DW      DOUBLE              ;main memory address pointer
  Ø14A'   ØØ                        DB      Ø
  Ø14B'   Ø131                      DW      SINGLE-DOUBLE       ;byte count
  Ø14D'   1Ø3Ø                      DW      1Ø3ØH               ;controller memory address pointer
  Ø14F'   25                        DB      25H                 ;controller halt command
  Ø15Ø'   ØØ                        DB      Ø                   ;halt command status byte

  Ø151'   A1                LSDCMD: DB      ØA1H
  Ø152'   Ø45C'                     DW      SINGLE
  Ø154'   ØØ                        DB      Ø
  Ø155'   Ø112                      DW      ECODE-SINGLE
  Ø157'   1Ø3Ø                      DW      1Ø3ØH
  Ø159'   25                        DB      25H
  Ø15A'   ØØ                        DB      Ø

  Ø15B'   2B                SOCMD:  DB      2BH                 ;output character to controller cmd
  Ø15C'   ØØ                        DB      Ø                   ;output data
  Ø15D'   ØØ                        DB      Ø                   ;output character command status
  Ø15E'   25                        DB      25H                 ;controller halt command
  Ø15F'   ØØ                        DB      Ø                   ;halt command status byte

  Ø16Ø'   A2                DOTCMD: DB      ØA2H                ;execute controller routine command
  Ø161'   1Ø3Ø                      DW      1Ø3ØH               ;format a track address
  Ø163'   ØØ                        DB      Ø                   ;execute command status
  Ø164'   25                        DB      25H                 ;halt command
  Ø165'   ØØ                        DB      Ø                   ;status byte

  Ø166'   A2                ATCMD:  DB      ØA2H
  Ø167'   113A                      DW      SDADVT              ;advance the track value address
  Ø169'   ØØ                        DB      Ø
  Ø16A'   25                        DB      25H
  Ø16B'   ØØ                        DB      Ø

  Ø16C'   1B                STABLE: DB      1BH                 ;26 sectors per track (256 bytes)
  Ø16D'   1Ø                        DB      1ØH                 ;15 sectors per track (512 bytes)
  Ø16E'   Ø9                        DB      9                   ;8 sectors per track (1Ø24 bytes)
                                    PAGE
```

```
Ø16F'   ØDØA                    SMESSG: DW      CRLFS
Ø171'   49 42 4D 2Ø                     DB      "IBM Compatable 8 inch Format Program"
Ø175'   43 6F 6D 7Ø
Ø179'   61 74 61 62
Ø17D'   6C 65 2Ø 38
Ø181'   2Ø 69 6E 63
Ø185'   68 2Ø 46 6F
Ø189'   72 6D 61 74
Ø18D'   2Ø 5Ø 72 6F
Ø191'   67 72 61 6D
Ø195'   ØDØA                            DW      CRLFS
Ø197'   53 65 6C 65                     DB      "Select a Drive ( Ø, 1, 2, or 3 ): "
Ø19B'   63 74 2Ø 61
Ø19F'   2Ø 44 72 69
Ø1A3'   76 65 2Ø 28
Ø1A7'   2Ø 3Ø 2C 2Ø
Ø1AB'   31 2C 2Ø 32
Ø1AF'   2C 2Ø 6F 72
Ø1B3'   2Ø 33 2Ø 29
Ø1B7'   3A 2Ø
Ø1B9'   ØØ                              DB      Ø
Ø1BA'   ØDØA                    BMESSG: DW      CRLFS
Ø1BC'   49 6D 7Ø 72                     DB      "Improper input - returning to start of program"
Ø1CØ'   6F 7Ø 65 72
Ø1C4'   2Ø 69 6E 7Ø
Ø1C8'   75 74 2Ø 2D
Ø1CC'   2Ø 72 65 74
Ø1DØ'   75 72 6E 69
Ø1D4'   6E 67 2Ø 74
Ø1D8'   6F 2Ø 73 74
Ø1DC'   61 72 74 2Ø
Ø1EØ'   6F 66 2Ø 7Ø
Ø1E4'   72 6F 67 72
Ø1E8'   61 6D
Ø1EA'   ØDØA                            DW      CRLFS
Ø1EC'   ØØ                              DB      Ø
Ø1ED'   ØDØA                    DMESSG: DW      CRLFS
Ø1EF'   53 65 6C 65                     DB      "Select double density ( 1 ) or single density ( Ø ): "
Ø1F3'   63 74 2Ø 64
Ø1F7'   6F 75 62 6C
Ø1FB'   65 2Ø 64 65
Ø1FF'   6E 73 69 74
Ø2Ø3'   79 2Ø 28 2Ø
Ø2Ø7'   31 2Ø 29 2Ø
Ø2ØB'   6F 72 2Ø 73
Ø2ØF'   69 6E 67 6C
Ø213'   65 2Ø 64 65
Ø217'   6E 73 69 74
Ø21B'   79 2Ø 28 2Ø
Ø21F'   3Ø 2Ø 29 3A
Ø223'   2Ø
Ø224'   ØØ                              DB      Ø
Ø225'   ØDØA                    LMESSG: DW      CRLFS
Ø227'   53 65 6C 65                     DB      "Select the byte length of a sector ( Ø=256, 1=512, 2=1Ø24 ): "
Ø22B'   63 74 2Ø 74
Ø22F'   68 65 2Ø 62
Ø233'   79 74 65 2Ø
```

```
 0237'   6C 65 6E 67
 023B'   74 68 20 6F
 023F'   66 20 61 20
 0243'   73 65 63 74
 0247'   6F 72 20 28
 024B'   20 30 3D 32
 024F'   35 36 2C 20
 0253'   31 3D 35 31
 0257'   32 2C 20 32
 025B'   3D 31 30 32
 025F'   34 20 29 3A
 0263'   20
 0264'   00                            DB      0
 0265'   0D0A                  HMESSG: DW      CRLFS
 0267'   53 65 6C 65                   DB      "Select single ( 0 ) or double ( 1 ) sided media : "
 026B'   63 74 20 73
 026F'   69 6E 67 6C
 0273'   65 20 28 20
 0277'   30 20 29 20
 027B'   6F 72 20 64
 027F'   6F 75 62 6C
 0283'   65 20 28 20
 0287'   31 20 29 20
 028B'   73 69 64 65
 028F'   64 20 6D 65
 0293'   64 69 61 20
 0297'   3A 20
 0299'   00                            DB      0
 029A'   0D0A                  RMESSG: DW      CRLFS
 029C'   44 72 69 76                   DB      "Drive not ready - restart program? ( 0 ) or cycle ( 1 ): "
 02A0'   65 20 6E 6F
 02A4'   74 20 72 65
 02A8'   61 64 79 20
 02AC'   2D 20 72 65
 02B0'   73 74 61 72
 02B4'   74 20 70 72
 02B8'   6F 67 72 61
 02BC'   6D 3F 20 28
 02C0'   20 30 20 29
 02C4'   20 6F 72 20
 02C8'   63 79 63 6C
 02CC'   65 20 28 20
 02D0'   31 20 29 3A
 02D4'   20
 02D5'   00                            DB      0
 02D6'   0D0A                  WMESSG: DW      CRLFS
 02D8'   57 72 69 74                   DB      "Write protected - restart program? ( 0 ) or cycle ( 1 ): "
 02DC'   65 20 70 72
 02E0'   6F 74 65 63
 02E4'   74 65 64 20
 02E8'   2D 20 72 65
 02EC'   73 74 61 72
 02F0'   74 20 70 72
 02F4'   6F 67 72 61
 02F8'   6D 3F 20 28
 02FC'   20 30 20 29
 0300'   20 6F 72 20
 0304'   63 79 63 6C
 0308'   65 20 28 20
```

```
  Ø3ØC'   31 2Ø 29 3A
  Ø31Ø'   2Ø
  Ø311'   ØØ                      DB      Ø
  Ø312'   ØDØA            FMESSG: DW      CRLFS
  Ø314'   46 6F 72 6D             DB      "Formatting finished"
  Ø318'   61 74 74 69
  Ø31C'   6E 67 2Ø 66
  Ø32Ø'   69 6E 69 73
  Ø324'   68 65 64
  Ø327'   ØDØA            CRLF:   DW      CRLFS
  Ø329'   ØØ                      DB      Ø
  Ø32A'   ØØ              DENSTY: DB      Ø
                                  PAGE
```

```
 Ø32B'                          DOUBLE  EQU     $
                                        .PHASE  1Ø3ØH
 1Ø3Ø    21 4ØØ3                DDFMT:  LD      HL,STATUS
 1Ø33    CB 7E                          BIT     7,M             ;check that the drive is ready
 1Ø35    3E 82                  NREXIT: LD      A,82H           ;drive not ready error code
 1Ø37    C8                             RET     Z               ;error exit
 1Ø38    CB 76                          BIT     6,M             ;test for write protected
 1Ø3A    3E 9Ø                          LD      A,9ØH           ;write protected error code
 1Ø3C    CØ                             RET     NZ              ;error exit
 1Ø3D    DD 36 ØB ØØ                    LD      (IX+ØBH),Ø      ;reset index counter
 1Ø41    3A 1ØC4                        LD      A,(DTRCK)       ;get the new track value
 1Ø44    FD BE Ø1                       CP      (IY+1)          ;compare with current track
 1Ø47    F5                             PUSH    AF              ;save the track
 1Ø48    C4 ØØA3                        CALL    NZ,SEEK         ;move the head(s) if needed
 1Ø4B    21 4ØØ1                        LD      HL,DISKD        ;pointer to disk shift register
 1Ø4E    11 4ØØ7                        LD      DE,CONTRL       ;pointer to control port
 1Ø51    F1                             POP     AF              ;recover the tack
 1Ø52    FE 2B                          CP      2BH             ;compare with track 43
 1Ø54    3E Ø4                          LD      A,4             ;no write precompensation
 1Ø56    38 Ø2                          JR      C,LOADPC        ;carry => track is less than 43
 1Ø58    3E 14                          LD      A,14H           ;write precompensation bit set
 1Ø5A    32 1Ø81                LOADPC: LD      (PRECMP),A      ;setup the write precompensation byte
 1Ø5D    9F                             SBC     A,A             ;push carry bit throughout accumulator
 1Ø5E    F6 FE                          OR      ØFEH            ;low current bit now set
 1Ø6Ø    FD A6 Ø2                       AND     (IY+2)          ;merge with drive pattern
 1Ø63    F6 Ø2                          OR      2               ;select side Ø
 1Ø65    FD 77 Ø2                       LD      (IY+2),A        ;restore drive pattern
 1Ø68    F6 ØC                          OR      ØCH             ;turn off step command
 1Ø6A    32 4ØØ5                        LD      (4ØØ5H),A       ;update the drive register
 1Ø6D    Ø6 5Ø                          LD      B,5ØH           ;preamble length
 1Ø6F    3A 4ØØ3                DDLBL1: LD      A,(STATUS)
 1Ø72    E6 1Ø                          AND     INDEX           ;look for index pulse
 1Ø74    2Ø F9                          JR      NZ,DDLBL1       ;wait for no index pulse present
 1Ø76    3A 4ØØ3                DDLBL2: LD      A,(STATUS)
 1Ø79    E6 1Ø                          AND     INDEX
 1Ø7B    28 F9                          JR      Z,DDLBL2        ;wait for leading edge of new indes pulse
 1Ø7D    3E 9Ø                          LD      A,9ØH           ;control byte - normal write/no CRC
 1Ø7F    12                             LD      (DE),A          ;initialize control port
 1Ø8Ø    3E ØØ                          LD      A,Ø
 1Ø81                           PRECMP  EQU     $-1             ;write precompensation & controller start
 1Ø82    32 4ØØ6                        LD      (4ØØ6H),A       ;start the controller
 1Ø85    36 4E                  DDLBL3: LD      M,4EH
 1Ø87    1Ø FC                          DJNZ    DDLBL3          ;write the preamble
 1Ø89    Ø6 ØC                          LD      B,ØCH           ;zero preamble length
 1Ø8B    36 ØØ                  DDLBL4: LD      M,Ø
 1Ø8D    1Ø FC                          DJNZ    DDLBL4          ;write the zero preamble
 1Ø8F    3E 8Ø                          LD      A,8ØH           ;control byte for 16 bit write
 1Ø91    12                             LD      (DE),A          ;change mode
 1Ø92    36 52                          LD      M,52H           ;first half of C2
 1Ø94    36 24                          LD      M,24H           ;second half of C2
 1Ø96    36 52                          LD      M,52H           ;another C2
 1Ø98    36 24                          LD      M,24H
 1Ø9A    36 52                          LD      M,52H           ;the third C2
 1Ø9C    3E 9Ø                          LD      A,9ØH           ;control byte 8 bit write
 1Ø9E    12                             LD      (DE),A          ;change mode
 1Ø9F    36 24                          LD      M,24H           ;finish the sync bytes
 1ØA1    36 FC                          LD      M,ØFCH          ;index mark
```

```
10A3    Ø6 32                   LD      B,32H           ;postamble length
10A5    36 4E           DDLBL5: LD      M,4EH
10A7    1Ø FC                   DJNZ    DDLBL5          ;write the postamble

10A9    Ø6 ØC           DMLOOP: LD      B,ØCH           ;zero preamble length
10AB    36 ØØ           DDLBL6: LD      M,Ø
10AD    1Ø FC                   DJNZ    DDLBL6          ;write the preamble
10AF    3E 81                   LD      A,81H           ;16 bit write mode w/CRC
10B1    12                      LD      (DE),A          ;change mode
10B2    36 44                   LD      M,44H           ;first half of A1
10B4    36 89                   LD      M,89H           ;second half of A1
10B6    36 44                   LD      M,44H           ;second A1
10B8    36 89                   LD      M,89H
10BA    36 44                   LD      M,44H           ;third A1
10BC    3E 91                   LD      A,91H           ;8 bit write mode w/CRC
10BE    12                      LD      (DE),A          ;change mode
10BF    36 89                   LD      M,89H           ;finish sync bytes
10C1    36 FE                   LD      M,ØFEH          ;sector header ID byte
10C3    36 ØØ                   LD      M,Ø             ;write the track number
10C4                    DTRCK   EQU     $-1
10C5    36 ØØ                   LD      M,Ø             ;write the side
10C6                    DSIDE   EQU     $-1
10C7    36 Ø1                   LD      M,1             ;write the sector number
10C8                    DSECT   EQU     $-1
10C9    36 Ø1                   LD      M,1             ;sector length code
10CA                    DLCODE  EQU     $-1
10CB    3E A1                   LD      A,ØA1H          ;mode to write CRC bytes
10CD    12                      LD      (DE),A          ;change mode
10CE    77                      LD      M,A
10CF    77                      LD      M,A             ;write the CRC bytes
10DØ    3E 9Ø                   LD      A,9ØH           ;reset CRC generator
10D2    12                      LD      (DE),A          ;change mode
10D3    Ø6 16                   LD      B,16H           ;4E postamble length
10D5    36 4E           DDLBL7: LD      M,4EH
10D7    1Ø FC                   DJNZ    DDLBL7          ;write the postamble
10D9    Ø6 ØC                   LD      B,ØCH           ;data field preamble
10DB    36 ØØ           DDLBL8: LD      M,Ø
10DD    1Ø FC                   DJNZ    DDLBL8          ;write the preamble
10DF    3E 81                   LD      A,81H           ;16 bit write w/CRC
10E1    12                      LD      (DE),A          ;change mode
10E2    36 44                   LD      M,44H           ;first half of A1
10E4    36 89                   LD      M,89H           ;second half of A1
10E6    36 44                   LD      M,44H           ;second A1
10E8    36 89                   LD      M,89H
10EA    36 44                   LD      M,44H           ;third A1
10EC    3E 91                   LD      A,91H           ;8 bit write w/CRC
10EE    12                      LD      (DE),A          ;change mode
10EF    36 89                   LD      M,89H           ;finish the 3 sync bytes
10F1    36 FB                   LD      M,ØFBH          ;data header ID byte
10F3    Ø6 4Ø                   LD      B,4ØH           ;sector length divided by four
10F4                    DSIZE   EQU     $-1
10F5    36 E5           DDLBL9: LD      M,ØE5H          ;empty sector data byte
10F7    36 E5                   LD      M,ØE5H
10F9    36 E5                   LD      M,ØE5H
10FB    36 E5                   LD      M,ØE5H          ;write four fill bytes
10FD    1Ø F6                   DJNZ    DDLBL9          ;test for data field write done
10FF    3E A1                   LD      A,ØA1H          ;CRC control byte
11Ø1    12                      LD      (DE),A          ;change mode
11Ø2    77                      LD      M,A             ;write the CRC bytes
```

```
 11Ø3    77                      LD      M,A
 11Ø4    3E 9Ø                   LD      A,9ØH           ;turn off the CRC generator
 11Ø6    12                      LD      (DE),A          ;change mode
 11Ø7    3A 1ØC8                 LD      A,(DSECT)       ;get the sector number
 11ØA    3C                      INC     A
 11ØB    FE 1B                   CP      1BH             ;test for last sector +1
 11ØC                    DLAST   EQU     $-1
 11ØD    36 4E                   LD      M,4EH           ;first byte of postamble
 11ØF    2Ø Ø2                   JR      NZ,$+4          ;zero => all sectors written
 1111    3E Ø1                   LD      A,1
 1113    32 1ØC8                 LD      (DSECT),A       ;update the sector number
 1116    Ø6 35                   LD      B,35H           ;postamble length less one
 1118    36 4E           DDLBLA: LD      M,4EH
 111A    1Ø FC                   DJNZ    DDLBLA          ;write the postamble
 111C    2Ø 8B                   JR      NZ,DMLOOP
 111E    36 4E                   LD      M,4EH           ;first fill byte
 112Ø    Ø6 ØØ                   LD      B,Ø             ;double sided bit test
 1121                    DDSBIT  EQU     $-1
 1122    3A 1ØC6                 LD      A,(DSIDE)
 1125    A8                      XOR     B               ;conditionally switch the side byte
 1126    32 1ØC6                 LD      (DSIDE),A       ;update the side byte
 1129    36 4E                   LD      M,4EH           ;second fill byte
 112B    Ø6 4F                   LD      B,4FH           ;preamble length less one
 112D    Ø8                      EX      AF,AF'          ;save the double sided status
 112E    36 4E           DLBLB:  LD      M,4EH           ;write a fill byte
 113Ø    3A 4ØØ3                 LD      A,(STATUS)
 1133    E6 1Ø                   AND     INDEX           ;wait for the index pulse
 1135    28 F7                   JR      Z,DLBLB
 1137    Ø8                      EX      AF,AF'          ;recover the double sided status
 1138    28 ØF                   JR      Z,DDLBLC        ;zero => track write is done
 113A    FD 7E Ø2                LD      A,(IY+2)        ;drive pattern
 113D    F6 ØC                   OR      ØCH             ;turn off the step command
 113F    E6 FD                   AND     ØFDH            ;change read/write heads
 1141    32 4ØØ5                 LD      (4ØØ5H),A       ;update the command register
 1144    36 4E                   LD      M,4EH           ;first preamble byte
 1146    C3 1Ø85                 JP      DDLBL3          ;format the other side
 1149    36 4E           DDLBLC: LD      M,4EH           ;trailing fill byte
 114B    36 4E                   LD      M,4EH           ;trailing fill byte
 114D    36 4E                   LD      M,4EH           ;trailing fill byte
 114F    AF                      XOR     A
 115Ø    12                      LD      (DE),A          ;turn off the write gate
 1151    3E Ø6                   LD      A,6
 1153    32 4ØØ6                 LD      (4ØØ6H),A       ;turn off the controller
 1156    3E 4Ø                   LD      A,4ØH           ;status code
 1158    C9                      RET
 1159    3A 1ØC4         DDADVT: LD      A,(DTRCK)       ;get the current track value
 115C    3C                      INC     A               ;increment
 115D    32 1ØC4                 LD      (DTRCK),A       ;restore the new value
 116Ø    C9                      RET                     ;return with current track value
                                 .DEPHASE
                                 PAGE
 Ø45C'                      SINGLE  EQU     $
                                    .PHASE  1Ø3ØH
 1Ø3Ø   3E ØØ               SDFMT:  LD      A,Ø             ;second byte filled with proper drive number
 1Ø32   CD ØØA6                     CALL    SDRIVE          ;select the new drive
 1Ø35   CØ                          RET     NZ              ;return if wrong value
 1Ø36   FD 7E Ø2                    LD      A,(IY+2)        ;get the drive pattern
 1Ø39   F6 ØF                       OR      ØFH             ;side Ø and no step command
 1Ø3B   32 4ØØ5                     LD      (4ØØ5H),A       ;update drive control register
 1Ø3E   21 ØØØØ                     LD      HL,Ø            ;delay for the head load
 1Ø41   2B                  SDWAIT: DEC     HL
 1Ø42   7C                          LD      A,H
 1Ø43   B5                          OR      L
 1Ø44   2Ø FB                       JR      NZ,SDWAIT
 1Ø46   DD 77 ØB                    LD      (IX+ØBH),A      ;reset the index counter
 1Ø49   CD ØØAØ             SDTRKØ: CALL    HOME            ;calibrate the head(s)
 1Ø4C   CB 6E                       BIT     5,M             ;test for track zero
 1Ø4E   28 Ø5                       JR      Z,SNREXT
 1Ø5Ø   21 4ØØ3             SDRDY:  LD      HL,STATUS
 1Ø53   CB 7E                       BIT     7,M             ;test for the drive ready
 1Ø55   3E 82               SNREXT: LD      A,82H           ;drive not ready code
 1Ø57   C8                          RET     Z               ;error exit
 1Ø58   CB 76                       BIT     6,M             ;write protect bit
 1Ø5A   3E 9Ø                       LD      A,9ØH           ;write protect error code
 1Ø5C   CØ                          RET     NZ
 1Ø5D   DD 36 ØB ØØ                 LD      (IX+ØBH),Ø      ;reset the index counter
 1Ø61   3A 1ØB6                     LD      A,(STRCK)       ;get the new track
 1Ø64   FD BE Ø1                    CP      (IY+1)          ;compare with current track
 1Ø67   C4 ØØA3                     CALL    NZ,SEEK         ;do track seek if necessary
 1Ø6A   21 4ØØ1                     LD      HL,DISKD        ;controller data register
 1Ø6D   11 4ØØ7                     LD      DE,CONTRL       ;control register
 1Ø7Ø   Ø6 28                       LD      B,28H           ;preamble length
 1Ø72   3A 4ØØ3             SDLBL1: LD      A,(STATUS)
 1Ø75   E6 1Ø                       AND     INDEX
 1Ø77   2Ø F9                       JR      NZ,SDLBL1       ;wait for no index pulse
 1Ø79   3A 4ØØ3             SDLBL2: LD      A,(STATUS)
 1Ø7C   E6 1Ø                       AND     INDEX
 1Ø7E   28 F9                       JR      Z,SDLBL2        ;wait for leading edge of new index pulse
 1Ø8Ø   3E 9Ø                       LD      A,9ØH           ;clear the CRC register & turn on write gate
 1Ø82   12                          LD      (DE),A          ;change modes
 1Ø83   3E 44                       LD      A,44H           ;single density & start bit
 1Ø85   32 4ØØ6                     LD      (4ØØ6H),A       ;start the controller
 1Ø88   36 FF               SDLBL3: LD      M,ØFFH
 1Ø8A   1Ø FC                       DJNZ    SDLBL3          ;write the preamble
 1Ø8C   3E 8Ø                       LD      A,8ØH           ;16 bit write mode
 1Ø8E   12                          LD      (DE),A          ;change modes
 1Ø8F   Ø6 ØC                       LD      B,ØCH           ;zero preamble length
 1Ø91   36 AA               SDLBL4: LD      M,ØAAH          ;half a zero cell
 1Ø93   1Ø FC                       DJNZ    SDLBL4          ;write the zero preamble
 1Ø95   36 F7                       LD      M,ØF7H          ;first half of FC
 1Ø97   3E 9Ø                       LD      A,9ØH           ;8 bit write mode
 1Ø99   12                          LD      (DE),A          ;change modes
 1Ø9A   36 7A                       LD      M,7AH           ;second half of FC
 1Ø9C   Ø6 1A                       LD      B,1AH           ;postamble length
 1Ø9E   36 FF               SDLBL5: LD      M,ØFFH
 1ØAØ   1Ø FC                       DJNZ    SDLBL5          ;write the postamble

 1ØA2   3E 8Ø               SMLOOP: LD      A,8ØH           ;16 bit write mode
 1ØA4    12                   LD      (DE),A          ;change modes
 1ØA5    Ø6 ØC                LD      B,ØCH           ;sector header preamble length
 1ØA7    36 AA        SDLBL6: LD      M,ØAAH          ;half a zero cell
 1ØA9    1Ø FC                DJNZ    SDLBL6          ;write the preamble
 1ØAB    3E 81                LD      A,81H           ;enable CRC & 16 bit write
 1ØAD    12                   LD      (DE),A          ;change modes
 1ØAE    36 F5                LD      M,ØF5H          ;first half of FE
 1ØBØ    3E 91                LD      A,91H           ;enable CRC & 8 bit write
 1ØB2    12                   LD      (DE),A          ;change modes
 1ØB3    36 7E                LD      M,7EH           ;second half of FE
 1ØB5    36 ØØ                LD      M,Ø             ;write the track
 1ØB6                 STRCK   EQU     $-1
 1ØB7    36 ØØ                LD      M,Ø             ;write the side byte
 1ØB8                 SSIDE   EQU     $-1
 1ØB9    36 Ø1                LD      M,1             ;write the sector number
 1ØBA                 SSECT   EQU     $-1
 1ØBB    36 ØØ                LD      M,Ø             ;write the sector length code
 1ØBD    3E A1                LD      A,ØA1H
 1ØBF    12                   LD      (DE),A          ;change modes
 1ØCØ    77                   LD      M,A
 1ØC1    77                   LD      M,A             ;write the CRC bytes
 1ØC2    3E 9Ø                LD      A,9ØH           ;reset the CRC
 1ØC4    12                   LD      (DE),A          ;change modes
 1ØC5    Ø6 ØB                LD      B,ØBH           ;sector header postamble length
 1ØC7    36 FF        SDLBL7: LD      M,ØFFH
 1ØC9    1Ø FC                DJNZ    SDLBL7          ;write the postamble
 1ØCB    3E 8Ø                LD      A,8ØH           ;16 bit write mode
 1ØCD    12                   LD      (DE),A          ;change modes
 1ØCE    Ø6 ØC                LD      B,ØCH           ;data field preamble length
 1ØDØ    36 AA        SDLBL8: LD      M,ØAAH          ;half a zero cell
 1ØD2    1Ø FC                DJNZ    SDLBL8          ;write the preamble
 1ØD4    3E 81                LD      A,81H           ;enable CRC & 16 bit write
 1ØD6    12                   LD      (DE),A          ;change modes
 1ØD7    36 F5                LD      M,ØF5H          ;first half of FB
 1ØD9    3E 91                LD      A,91H           ;8 bit write
 1ØDB    12                   LD      (DE),A          ;change modes
 1ØDC    36 6F                LD      M,6FH           ;second half of FB
 1ØDE    Ø6 8Ø                LD      B,8ØH           ;sector data field length
 1ØEØ    36 E5        SDLBL9: LD      M,ØE5H
 1ØE2    1Ø FC                DJNZ    SDLBL9          ;write the data field
 1ØE4    3E A1                LD      A,ØA1H
 1ØE6    12                   LD      (DE),A          ;change modes
 1ØE7    77                   LD      M,A
 1ØE8    77                   LD      M,A             ;write the CRC bytes
 1ØE9    3E 9Ø                LD      A,9ØH           ;reset the CRC
 1ØEB    12                   LD      (DE),A          ;change modes
 1ØEC    3A 1ØBA              LD      A,(SSECT)       ;get the current sector
 1ØEF    3C                   INC     A               ;advance
 1ØFØ    FE 1B                CP      1BH             ;compare with 27
 1ØF2    36 FF                LD      M,ØFFH          ;first postamble byte
 1ØF4    2Ø Ø2                JR      NZ,$+4          ;zero => all sectors written
 1ØF6    3E Ø1                LD      A,1
 1ØF8    32 1ØBA              LD      (SSECT),A       ;update the sector
 1ØFB    Ø6 1A                LD      B,1AH           ;postamble length less one
 1ØFD    36 FF        SDLBLA: LD      M,ØFFH
 1ØFF    1Ø FC                DJNZ    SDLBLA          ;write the postamble
 11Ø1    2Ø 9F                JR      NZ,SMLOOP       ;test for more sectors to format
 11Ø3    36 FF                LD      M,ØFFH          ;first fill byte
 11Ø5    Ø6 ØØ                LD      B,Ø             ;side bit
 11Ø6                       SDSBIT  EQU     $-1
 11Ø7    3A 1ØB8                    LD      A,(SSIDE)       ;get the current side
 11ØA    A8                         XOR     B               ;conditionally switch side bits
 11ØB    32 1ØB8                    LD      (SSIDE),A       ;update the side byte
 11ØE    36 FF                      LD      M,ØFFH          ;write second fill byte
 111Ø    Ø6 19                      LD      B,19H           ;preamble length less one
 1112    Ø8                         EX      AF,AF'          ;save the double sided status
 1113    36 FF              SDLBLB: LD      M,ØFFH          ;write a fill byte
 1115    3A 4ØØ3                    LD      A,(STATUS)
 1118    E6 1Ø                      AND     INDEX
 111A    28 F7                      JR      Z,SDLBLB        ;wait for the index hole
 111C    Ø8                         EX      AF,AF'          ;recover the double sided status
 111D    28 ØF                      JR      Z,SDLBLC        ;zero => single sided
 111F    FD 7E Ø2                   LD      A,(IY+2)        ;get the drive pattern
 1122    F6 ØC                      OR      ØCH             ;turn off the step command
 1124    E6 FD                      AND     ØFDH            ;turn on head one
 1126    32 4ØØ5                    LD      (4ØØ5H),A       ;update drive control register
 1129    36 FF                      LD      M,ØFFH          ;write first preamble byte
 112B    C3 1Ø88                    JP      SDLBL3          ;go format the other side
 112E    36 FF              SDLBLC: LD      M,ØFFH          ;trailing byte
 113Ø    AF                         XOR     A
 1131    12                         LD      (DE),A          ;turn off write gate
 1132    3E Ø6                      LD      A,6
 1134    32 4ØØ6                    LD      (4ØØ6H),A       ;turn off the controller
 1137    3E 4Ø                      LD      A,4ØH           ;status code
 1139    C9                         RET
 113A    3A 1ØB6            SDADVT: LD      A,(STRCK)       ;get the current track
 113D    3C                         INC     A               ;advance track value
 113E    32 1ØB6                    LD      (STRCK),A       ;update the track value
 1141    C9                         RET                     ;return with track value
                                    .DEPHASE
 Ø56E'                      ECODE   EQU     $
                                    END
 ØØØØ'    31 Ø4A2'              START:  LD      SP,ECODE+3ØH
 ØØØ3'    21 1Ø3Ø                       LD      HL,1Ø3ØH
 ØØØ6'    22 Ø177'                      LD      (DOTCMD+1),HL
 ØØØ9'    3E 2Ø                         LD      A,2ØH
 ØØØB'    32 Ø4Ø4'                      LD      (DATA-NSFMT+FORMAT),A
 ØØØE'    32 Ø4Ø6'                      LD      (CPDATA-NSFMT+FORMAT),A
 ØØ11'    AF                            XOR     A
 ØØ12'    32 Ø47Ø'                      LD      (TRACK-NSFMT+FORMAT),A
 ØØ15'    21 Ø18C'                      LD      HL,SMESSG
 ØØ18'    CD Ø13E'                      CALL    OUTM
 ØØ1B'    CD Ø14A'                      CALL    INPUT
 ØØ1E'    D2 ØØ2A'                      JP      NC,DATAOK
 ØØ21'    21 Ø1E2'              DEXIT:  LD      HL,BMESSG
 ØØ24'    CD Ø13E'                      CALL    OUTM
 ØØ27'    C3 ØØØØ'                      JP      START
 ØØ2A'    32 Ø385'              DATAOK: LD      (FORMAT+1),A
 ØØ2D'    21 Ø24D'                      LD      HL,LMESSG
 ØØ3Ø'    CD Ø13E'                      CALL    OUTM
 ØØ33'    CD Ø14A'                      CALL    INPUT
 ØØ36'    DA ØØ21'                      JP      C,DEXIT
 ØØ39'    FE Ø3                         CP      3
 ØØ3B'    CA ØØ21'                      JP      Z,DEXIT
 ØØ3E'    16 ØØ                         LD      D,Ø
 ØØ4Ø'    5F                            LD      E,A
 ØØ41'    21 Ø182'                      LD      HL,STABLE
 ØØ44'    19                            ADD     HL,DE
 ØØ45'    7E                            LD      A,M
 ØØ46'    32 Ø3DE'                      LD      (STRACK-NSFMT+FORMAT),A
 ØØ49'    D5                            PUSH    DE
 ØØ4A'    21 Ø215'                      LD      HL,DMESSG
 ØØ4D'    CD Ø13E'                      CALL    OUTM
 ØØ5Ø'    CD Ø14A'                      CALL    INPUT
 ØØ53'    D1                            POP     DE
 ØØ54'    DA ØØ21'                      JP      C,DEXIT
 ØØ57'    E6 Ø1                         AND     1
 ØØ59'    Ø6 51                         LD      B,Ø51H
 ØØ5B'    CA ØØ65'                      JP      Z,STOREO
 ØØ5E'    F5                            PUSH    AF
 ØØ5F'    ØF                            RRCA
 ØØ6Ø'    83                            ADD     A,E
 ØØ61'    5F                            LD      E,A
 ØØ62'    F1                            POP     AF
 ØØ63'    Ø6 D1                         LD      B,ØD1H
 ØØ65'    32 Ø3D7'              STOREO: LD      (DEN1-NSFMT+FORMAT),A
 ØØ68'    78                            LD      A,B
 ØØ69'    32 Ø41Ø'                      LD      (DEN2-NSFMT+FORMAT),A
 ØØ6C'    D5                            PUSH    DE
 ØØ6D'    21 Ø2BF'                      LD      HL,HMESSG
 ØØ7Ø'    CD Ø13E'                      CALL    OUTM
 ØØ73'    CD Ø14A'                      CALL    INPUT
 ØØ76'    D1                            POP     DE
 ØØ77'    DA ØØ21'                      JP      C,DEXIT
 ØØ7A'    E6 Ø1                         AND     1
 ØØ7C'    32 Ø45Ø'                      LD      (DFLAG-NSFMT+FORMAT),A
 ØØ7F'    CA ØØ86'                      JP      Z,DATAC
 ØØ82'    Ø7                            RLCA
 ØØ83'    Ø7                            RLCA
```

```
ØØ84'   83                      ADD     A,E
ØØ85'   5F                      LD      E,A
ØØ86'   D5              DATAC:  PUSH    DE
ØØ87'   21 Ø282'                LD      HL,NMESSG
ØØ8A'   CD Ø13E'                CALL    OUTM
ØØ8D'   CD Ø14A'                CALL    INPUT
ØØ9Ø'   D1                      POP     DE
ØØ91'   DA ØØ21'                JP      C,DEXIT
ØØ94'   E6 Ø1                   AND     1
ØØ96'   CA ØØAE'                JP      Z,LOADC
ØØ99'   7B                      LD      A,E
ØØ9A'   E6 8Ø                   AND     8ØH
ØØ9C'   3E 1Ø                   LD      A,1ØH
ØØ9E'   CA ØØA6'                JP      Z,STORED
ØØA1'   21 Ø1Ø5'                LD      HL,TYPE-8ØH
ØØA4'   19                      ADD     HL,DE
ØØA5'   7E                      LD      A,M
ØØA6'   32 Ø4Ø6'        STORED: LD      (CPDATA-NSFMT+FORMAT),A
ØØA9'   3E E5                   LD      A,ØE5H
ØØAB'   32 Ø4Ø4'                LD      (DATA-NSFMT+FORMAT),A
ØØAE'   21 Ø167'        LOADC:  LD      HL,LFDCMD
ØØB1'   Ø6 ØA                   LD      B,ØAH
ØØB3'   CD Ø11B'                CALL    LCMD
ØØB6'   21 Ø176'                LD      HL,DOTCMD
ØØB9'   Ø6 Ø6                   LD      B,6
ØØBB'   CD Ø11B'                CALL    LCMD
ØØBE'   CA ØØDD'                JP      Z,PROCED
ØØC1'   21 Ø2F4'                LD      HL,RMESSG
ØØC4'   FE 82                   CP      82H
ØØC6'   CA ØØCC'                JP      Z,$+6
ØØC9'   21 Ø33Ø'                LD      HL,WMESSG
ØØCC'   CD Ø13E'                CALL    OUTM
ØØCF'   CD Ø14A'                CALL    INPUT
ØØD2'   DA ØØ21'                JP      C,DEXIT
ØØD5'   E6 Ø1                   AND     1
ØØD7'   CA ØØØØ'                JP      Z,START
ØØDA'   C3 ØØAE'                JP      LOADC
ØØDD'   21 Ø381'        PROCED: LD      HL,CRLF
ØØEØ'   CD Ø13E'                CALL    OUTM
ØØE3'   21 1Ø4F                 LD      HL,ENTRY
ØØE6'   22 Ø177'                LD      (DOTCMD+1),HL
ØØE9'   3E 2A           CONTUE: LD      A,"*"
ØØEB'   CD Ø134'                CALL    OUTPUT
ØØEE'   21 Ø17C'                LD      HL,ATCMD
ØØF1'   Ø6 Ø6                   LD      B,6
ØØF3'   CD Ø11B'                CALL    LCMD
ØØF6'   47                      LD      B,A
ØØF7'   3A Ø3DE'                LD      A,(STRACK-NSFMT+FORMAT)
ØØFA'   B8                      CP      B
ØØFB'   C2 Ø1Ø7'                JP      NZ,FMTRCK
ØØFE'   21 Ø36C'        ENDFMT: LD      HL,FMESSG
Ø1Ø1'   CD Ø13E'                CALL    OUTM
Ø1Ø4'   C3 ØØØØ'                JP      START
Ø1Ø7'   21 Ø176'        FMTRCK: LD      HL,DOTCMD
Ø1ØA'   Ø6 Ø6                   LD      B,6
Ø1ØC'   CD Ø11B'                CALL    LCMD
Ø1ØF'   CA ØØE9'                JP      Z,CONTUE
Ø112'   21 Ø2F4'                LD      HL,RMESSG
Ø115'   CD Ø13E'                CALL    OUTM
```

```
  Ø118'    C3 ØØFE'                 JP      ENDFMT

  Ø11B'    11 ØØ5Ø          LCMD:   LD      DE,5ØH
  Ø11E'    7E                       LD      A,M
  Ø11F'    12                       LD      (DE),A
  Ø12Ø'    23                       INC     HL
  Ø121'    13                       INC     DE
  Ø122'    Ø5                       DEC     B
  Ø123'    C2 Ø11E'                 JP      NZ,LCMD+3

  Ø126'    D3 EF            ECMD:   OUT     (ØEFH),A
  Ø128'    1B                       DEC     DE
  Ø129'    1A                       LD      A,(DE)
  Ø12A'    B7                       OR      A
  Ø12B'    CA Ø129'                 JP      Z,ECMD+3
  Ø12E'    3A ØØ53                  LD      A,(53H)
  Ø131'    FE 4Ø                    CP      4ØH
  Ø133'    C9                       RET

  Ø134'    21 Ø172'         OUTPUT: LD      HL,SOCMD+1
  Ø137'    Ø6 Ø5                    LD      B,5
  Ø139'    77                       LD      M,A
  Ø13A'    2B                       DEC     HL
  Ø13B'    C3 Ø11B'                 JP      LCMD

  Ø13E'    7E               OUTM:   LD      A,M
  Ø13F'    B7                       OR      A
  Ø14Ø'    C8                       RET     Z
  Ø141'    E5                       PUSH    HL
  Ø142'    CD Ø134'                 CALL    OUTPUT
  Ø145'    E1                       POP     HL
  Ø146'    23                       INC     HL
  Ø147'    C3 Ø13E'                 JP      OUTM

  Ø14A'    21 ØØ3F          INPUT:  LD      HL,3FH
  Ø14D'    3E 4Ø                    LD      A,4ØH
  Ø14F'    96                       SUB     M
  Ø15Ø'    C2 Ø14D'                 JP      NZ,INPUT+3
  Ø153'    77                       LD      M,A
  Ø154'    2B                       DEC     HL
  Ø155'    7E                       LD      A,M
  Ø156'    F5                       PUSH    AF
  Ø157'    CD Ø134'                 CALL    OUTPUT
  Ø15A'    F1                       POP     AF
  Ø15B'    E6 7F                    AND     7FH
  Ø15D'    FE 3Ø                    CP      3ØH
  Ø15F'    D8                       RET     C
  Ø16Ø'    FE 34                    CP      34H
  Ø162'    3F                       CCF
  Ø163'    D8                       RET     C
  Ø164'    E6 Ø3                    AND     3
  Ø166'    C9                       RET
                                    PAGE
```

```
 Ø167'   A1              LFDCMD: DB      ØA1H
 Ø168'   Ø384'                   DW      FORMAT
 Ø16A'   ØØ                      DB      Ø
 Ø16B'   ØØEE                    DW      ECODE-FORMAT
 Ø16D'   1Ø3Ø                    DW      1Ø3ØH
 Ø16F'   25                      DB      25H
 Ø17Ø'   ØØ                      DB      Ø

 Ø171'   2B              SOCMD:  DB      2BH
 Ø172'   ØØ                      DB      Ø
 Ø173'   ØØ                      DB      Ø
 Ø174'   25                      DB      25H
 Ø175'   ØØ                      DB      Ø

 Ø176'   A2              DOTCMD: DB      ØA2H
 Ø177'   1Ø3Ø                    DW      1Ø3ØH
 Ø179'   ØØ                      DB      Ø
 Ø17A'   25                      DB      25H
 Ø17B'   ØØ                      DB      Ø

 Ø17C'   A2              ATCMD:  DB      ØA2H
 Ø17D'   1114                    DW      ADVTRK
 Ø17F'   ØØ                      DB      Ø
 Ø18Ø'   25                      DB      25H
 Ø181'   ØØ                      DB      Ø

 Ø182'   23              STABLE: DB      35
 Ø183'   28                      DB      4Ø
 Ø184'   5Ø                      DB      8Ø

 Ø185'   9Ø              TYPE:   DB      9ØH
 Ø186'   AØ                      DB      ØAØH
 Ø187'   CØ                      DB      ØCØH
 Ø188'   ØØ                      DB      Ø
 Ø189'   FØ                      DB      ØFØH
 Ø18A'   DØ                      DB      ØDØH
 Ø18B'   EØ                      DB      ØEØH
                                 PAGE
 Ø18C'   ØDØA                      SMESSG: DW      CRLFS
 Ø18E'   4E 6F 72 74                       DB      "North Star Compatable 5 1/4 inch Format Program"
 Ø192'   68 2Ø 53 74
 Ø196'   61 72 2Ø 43
 Ø19A'   6F 6D 7Ø 61
 Ø19E'   74 61 62 6C
 Ø1A2'   65 2Ø 35 2Ø
 Ø1A6'   31 2F 34 2Ø
 Ø1AA'   69 6E 63 68
 Ø1AE'   2Ø 46 6F 72
 Ø1B2'   6D 61 74 2Ø
 Ø1B6'   5Ø 72 6F 67
 Ø1BA'   72 61 6D
 Ø1BD'   ØDØA                              DW      CRLFS
 Ø1BF'   53 65 6C 65                       DB      "Select a Drive ( Ø, 1, 2, or 3 ): "
 Ø1C3'   63 74 2Ø 61
 Ø1C7'   2Ø 44 72 69
 Ø1CB'   76 65 2Ø 28
 Ø1CF'   2Ø 3Ø 2C 2Ø
 Ø1D3'   31 2C 2Ø 32
 Ø1D7'   2C 2Ø 6F 72
 Ø1DB'   2Ø 33 2Ø 29
 Ø1DF'   3A 2Ø
 Ø1E1'   ØØ                                DB      Ø
 Ø1E2'   ØDØA                      BMESSG: DW      CRLFS
 Ø1E4'   49 6D 7Ø 72                       DB      "Improper input - returning to start of program"
 Ø1E8'   6F 7Ø 65 72
 Ø1EC'   2Ø 69 6E 7Ø
 Ø1FØ'   75 74 2Ø 2D
 Ø1F4'   2Ø 72 65 74
 Ø1F8'   75 72 6E 69
 Ø1FC'   6E 67 2Ø 74
 Ø2ØØ'   6F 2Ø 73 74
 Ø2Ø4'   61 72 74 2Ø
 Ø2Ø8'   6F 66 2Ø 7Ø
 Ø2ØC'   72 6F 67 72
 Ø21Ø'   61 6D
 Ø212'   ØDØA                              DW      CRLFS
 Ø214'   ØØ                                DB      Ø
 Ø215'   ØDØA                      DMESSG: DW      CRLFS
 Ø217'   53 65 6C 65                       DB      "Select double density ( 1 ) or single density ( Ø ): "
 Ø21B'   63 74 2Ø 64
 Ø21F'   6F 75 62 6C
 Ø223'   65 2Ø 64 65
 Ø227'   6E 73 69 74
 Ø22B'   79 2Ø 28 2Ø
 Ø22F'   31 2Ø 29 2Ø
 Ø233'   6F 72 2Ø 73
 Ø237'   69 6E 67 6C
 Ø23B'   65 2Ø 64 65
 Ø23F'   6E 73 69 74
 Ø243'   79 2Ø 28 2Ø
 Ø247'   3Ø 2Ø 29 3A
 Ø24B'   2Ø
 Ø24C'   ØØ                                DB      Ø
 Ø24D'   ØDØA                      LMESSG: DW      CRLFS
 Ø24F'   53 65 6C 65                       DB      "Select the number of tracks ( Ø=35, 1=4Ø, 2=8Ø ): "
```

```
  Ø253'    63 74 2Ø 74
  Ø257'    68 65 2Ø 6E
  Ø25B'    75 6D 62 65
  Ø25F'    72 2Ø 6F 66
  Ø263'    2Ø 74 72 61
  Ø267'    63 6B 73 2Ø
  Ø26B'    28 2Ø 3Ø 3D
  Ø26F'    33 35 2C 2Ø
  Ø273'    31 3D 34 3Ø
  Ø277'    2C 2Ø 32 3D
  Ø27B'    38 3Ø 2Ø 29
  Ø27F'    3A 2Ø
  Ø281'    ØØ                            DB      Ø
  Ø282'    ØDØA                 NMESSG:  DW      CRLFS
  Ø284'    53 65 6C 65                   DB      "Select North Star ( Ø ) or CP/M ( 1 ) data compatibility: "
  Ø288'    63 74 2Ø 4E
  Ø28C'    6F 72 74 68
  Ø29Ø'    2Ø 53 74 61
  Ø294'    72 2Ø 28 2Ø
  Ø298'    3Ø 2Ø 29 2Ø
  Ø29C'    6F 72 2Ø 43
  Ø2AØ'    5Ø 2F 4D 2Ø
  Ø2A4'    28 2Ø 31 2Ø
  Ø2A8'    29 2Ø 64 61
  Ø2AC'    74 61 2Ø 63
  Ø2BØ'    6F 6D 7Ø 61
  Ø2B4'    74 69 62 69
  Ø2B8'    6C 69 74 79
  Ø2BC'    3A 2Ø
  Ø2BE'    ØØ                            DB      Ø
  Ø2BF'    ØDØA                 HMESSG:  DW      CRLFS
  Ø2C1'    53 65 6C 65                   DB      "Select single ( Ø ) or double ( 1 ) sided media : "
  Ø2C5'    63 74 2Ø 73
  Ø2C9'    69 6E 67 6C
  Ø2CD'    65 2Ø 28 2Ø
  Ø2D1'    3Ø 2Ø 29 2Ø
  Ø2D5'    6F 72 2Ø 64
  Ø2D9'    6F 75 62 6C
  Ø2DD'    65 2Ø 28 2Ø
  Ø2E1'    31 2Ø 29 2Ø
  Ø2E5'    73 69 64 65
  Ø2E9'    64 2Ø 6D 65
  Ø2ED'    64 69 61 2Ø
  Ø2F1'    3A 2Ø
  Ø2F3'    ØØ                            DB      Ø
  Ø2F4'    ØDØA                 RMESSG:  DW      CRLFS
  Ø2F6'    44 72 69 76                   DB      "Drive not ready - restart program? ( Ø ) or cycle ( 1 ): "
  Ø2FA'    65 2Ø 6E 6F
  Ø2FE'    74 2Ø 72 65
  Ø3Ø2'    61 64 79 2Ø
  Ø3Ø6'    2D 2Ø 72 65
  Ø3ØA'    73 74 61 72
  Ø3ØE'    74 2Ø 7Ø 72
  Ø312'    6F 67 72 61
  Ø316'    6D 3F 2Ø 28
  Ø31A'    2Ø 3Ø 2Ø 29
  Ø31E'    2Ø 6F 72 2Ø
  Ø322'    63 79 63 6C
  Ø326'    65 2Ø 28 2Ø
  Ø32A'    31 2Ø 29 3A
  Ø32E'    2Ø
  Ø32F'    ØØ                               DB      Ø
  Ø33Ø'    ØDØA                     WMESSG: DW      CRLFS
  Ø332'    57 72 69 74                      DB      "Write protected - restart program? ( Ø ) or cycle ( 1 ): "
  Ø336'    65 2Ø 7Ø 72
  Ø33A'    6F 74 65 63
  Ø33E'    74 65 64 2Ø
  Ø342'    2D 2Ø 72 65
  Ø346'    73 74 61 72
  Ø34A'    74 2Ø 7Ø 72
  Ø34E'    6F 67 72 61
  Ø352'    6D 3F 2Ø 28
  Ø356'    2Ø 3Ø 2Ø 29
  Ø35A'    2Ø 6F 72 2Ø
  Ø35E'    63 79 63 6C
  Ø362'    65 2Ø 28 2Ø
  Ø366'    31 2Ø 29 3A
  Ø36A'    2Ø
  Ø36B'    ØØ                               DB      Ø
  Ø36C'    ØDØA                     FMESSG: DW      CRLFS
  Ø36E'    46 6F 72 6D                      DB      "Formatting finished"
  Ø372'    61 74 74 69
  Ø376'    6E 67 2Ø 66
  Ø37A'    69 6E 69 73
  Ø37E'    68 65 64
  Ø381'    ØDØA                     CRLF:   DW      CRLFS
  Ø383'    ØØ                               DB      Ø
                                            PAGE
```

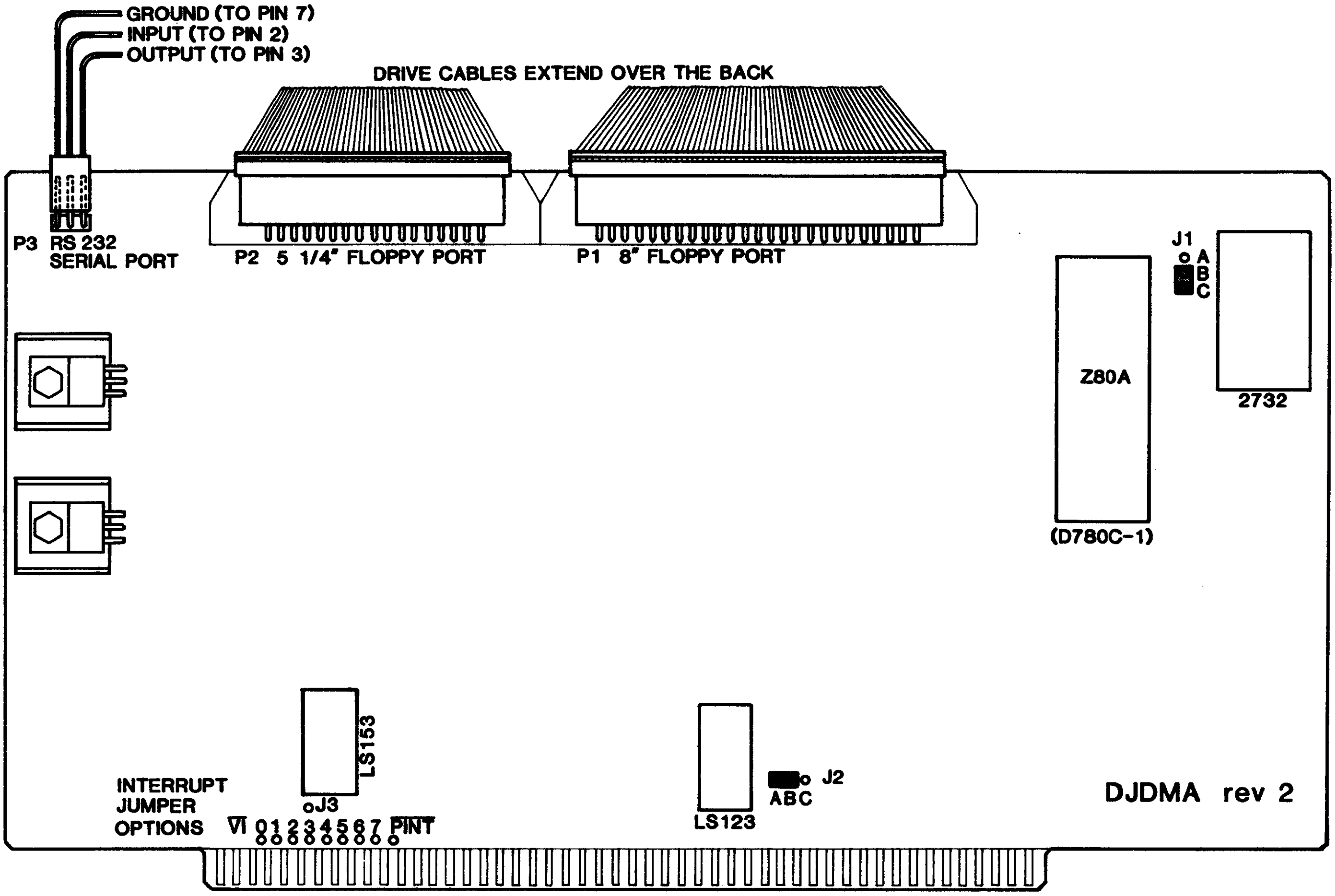
```
0384'                          FORMAT  EQU     $
                                       .PHASE  1030H
1030    3E 00                  NSFMT:  LD      A,0
1032    CD 00A6                        CALL    SDRIVE
1035    C0                             RET     NZ
1036    DD 36 0B 00                    LD      (IX+0BH),0
103A    FD 7E 02                       LD      A,(IY+2)
103D    F6 0E                          OR      0EH
103F    32 4004                        LD      (4004H),A
1042    CD 00A9                        CALL    HSYNC
1045    3E 82                  NREXIT: LD      A,82H
1047    C8                             RET     Z
1048    CD 00A0                TRACK0: CALL    HOME
104B    CB 6E                          BIT     5,M
104D    28 F6                          JR      Z,NREXIT
104F    DD 36 0B 00            ENTRY:  LD      (IX+0BH),0
1053    3A 111C                        LD      A,(TRACK)
1056    FD BE 01                       CP      (IY+1)
1059    C4 00A3                        CALL    NZ,SEEK
105C    3A 4003                        LD      A,(4003H)
105F    E6 40                          AND     40H
1061    3E 90                          LD      A,90H
1063    C0                             RET     NZ
1064    DD 36 0A 80                    LD      (IX+0AH),80H
1068    CD 00A9                WSECT0: CALL    HSYNC
106B    28 D8                          JR      Z,NREXIT
106D    AF                             XOR     A
106E    DD BE 0A                       CP      (IX+0AH)
1071    20 F5                          JR      NZ,WSECT0
1073    3E 90                          LD      A,90H
1075    32 4007                        LD      (CONTRL),A
1078    21 4001                        LD      HL,DISKD
107B    0E 00                          LD      C,0
107D    DD 71 09                       LD      (IX+9),C
1080    06 11                          LD      B,11H
1082    3E 00                          LD      A,0
1083                           DEN1    EQU     $-1
1084    1F                             RRA
1085    3E 64                          LD      A,64H
1087    30 0F                          JR      NC,CSTART
1089    3E 18                          LD      A,18H
108A                           STRACK  EQU     $-1
108B    1F                             RRA
108C    C6 05                          ADD     A,5
108E    FD BE 01                       CP      (IY+1)
1091    9F                             SBC     A,A
1092    E6 10                          AND     10H
1094    F6 24                          OR      24H
1096    06 20                          LD      B,20H
1098    32 4006                CSTART: LD      (4006H),A

109B    36 00                  ZEROW:  LD      M,0
109D    E3                             EX      (SP),HL
109E    E3                             EX      (SP),HL
109F    10 FA                          DJNZ    ZEROW
10A1    3A 1083                        LD      A,(DEN1)
10A4    B7                             OR      A
```

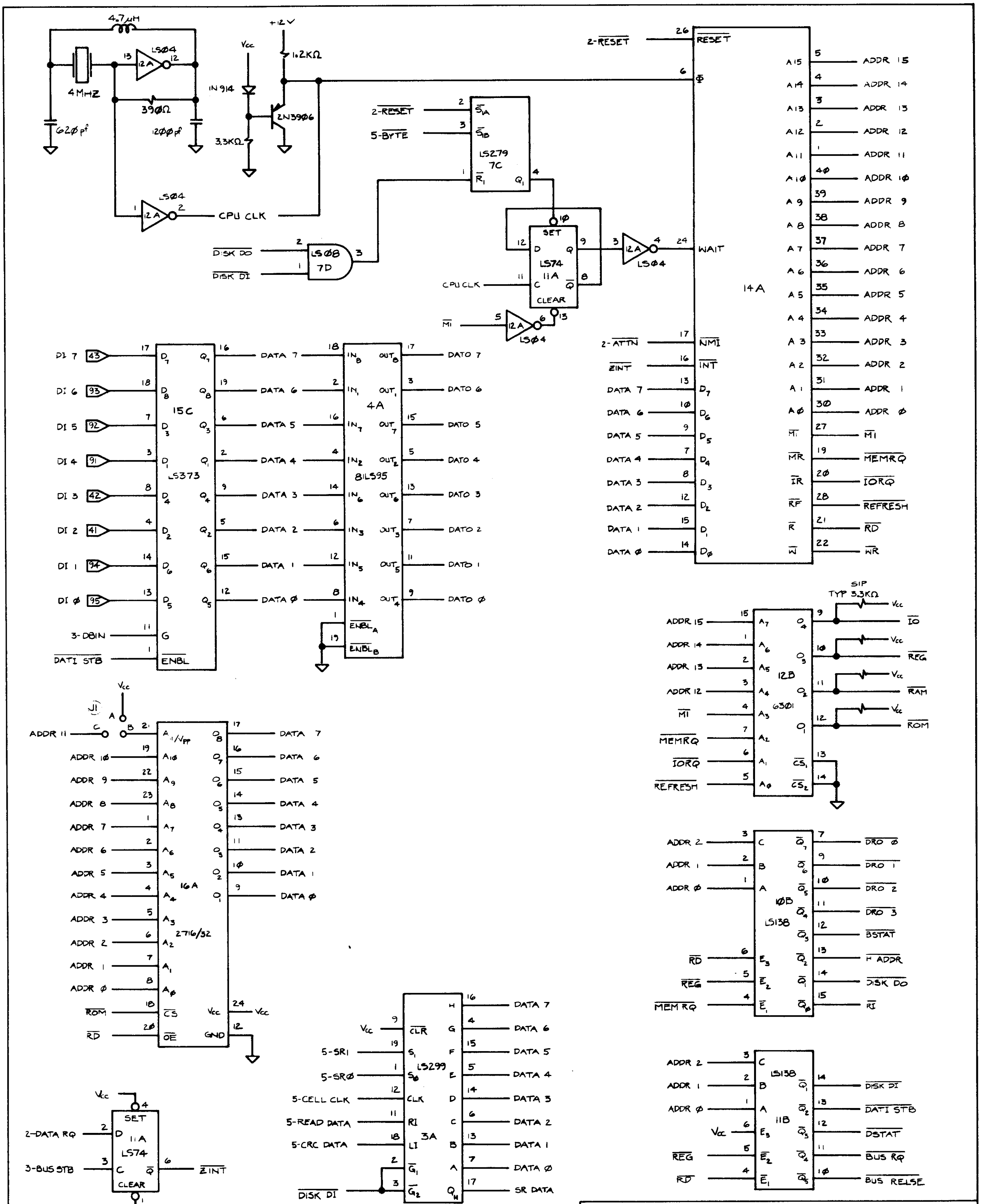
```
1ØA5    28 Ø4                   JR      Z,LASTS
1ØA7    36 FB                   LD      M,ØFBH
1ØA9    E3                      EX      (SP),HL
1ØAA    E3                      EX      (SP),HL
1ØAB    36 FB           LASTS:  LD      M,ØFBH
1ØAD    Ø6 5C                   LD      B,5CH
1ØAF    1E 2Ø                   LD      E,2ØH
1ØBØ                    DATA    EQU     $-1
1ØB1    16 2Ø                   LD      D,2ØH
1ØB2                    CPDATA  EQU     $-1
1ØB3    AF                      XOR     A
1ØB4    E3              D1LOOP: EX      (SP),HL
1ØB5    E3                      EX      (SP),HL
1ØB6    73                      LD      M,E
1ØB7    AB                      XOR     E
1ØB8    Ø7                      RLCA
1ØB9    1Ø F9                   DJNZ    D1LOOP
1ØBB    Ø6 51                   LD      B,51H
1ØBC                    DEN2    EQU     $-1
1ØBD    E3                      EX      (SP),HL
1ØBE    E3                      EX      (SP),HL
1ØBF    72                      LD      M,D
1ØCØ    AA                      XOR     D
1ØC1    Ø7                      RLCA
1ØC2    Ø8                      EX      AF,AF'
1ØC3    7B                      LD      A,E
1ØC4    32 1ØB2                 LD      (CPDATA),A
1ØC7    Ø8                      EX      AF,AF'
1ØC8    E3                      EX      (SP),HL
1ØC9    E3                      EX      (SP),HL
1ØCA    73                      LD      M,E
1ØCB    AB                      XOR     E
1ØCC    Ø7                      RLCA
1ØCD    E3              D2LOOP: EX      (SP),HL
1ØCE    E3                      EX      (SP),HL
1ØCF    73                      LD      M,E
1ØDØ    AB                      XOR     E
1ØD1    Ø7                      RLCA
1ØD2    E3                      EX      (SP),HL
1ØD3    E3                      EX      (SP),HL
1ØD4    73                      LD      M,E
1ØD5    AB                      XOR     E
1ØD6    Ø7                      RLCA
1ØD7    1Ø F4                   DJNZ    D2LOOP
1ØD9    E3                      EX      (SP),HL
1ØDA    E3                      EX      (SP),HL
1ØDB    77                      LD      M,A
1ØDC    3A 1Ø83                 LD      A,(DEN1)
1ØDF    B7                      OR      A
1ØEØ    Ø6 11                   LD      B,11H
1ØE2    28 Ø2                   JR      Z,$+4
1ØE4    Ø6 2Ø                   LD      B,2ØH
1ØE6    E3              ILOOP:  EX      (SP),HL
1ØE7    E3                      EX      (SP),HL
1ØE8    73                      LD      M,E
1ØE9    3A 4ØØ3                 LD      A,(STATUS)
1ØEC    E6 1Ø                   AND     INDEX
1ØEE    28 F6                   JR      Z,ILOOP
1ØFØ    ØC                      INC     C
```

```
  1ØF1     3E ØA                      LD      A,ØAH
  1ØF3     B9                         CP      C
  1ØF4     2Ø A5                      JR      NZ,ZEROW
  1ØF6     ØE ØØ                      LD      C,Ø
  1ØF8     3A 111D                    LD      A,(DSIDE)
  1ØFB     EE ØØ                      XOR     Ø
  1ØFC                        DFLAG   EQU     $-1
  1ØFD     32 111D                    LD      (DSIDE),A
  11ØØ     28 ØC                      JR      Z,FTDONE
  11Ø2     FD 7E Ø2                   LD      A,(IY+2)
  11Ø5     F6 ØE                      OR      ØEH
  11Ø7     E6 FD                      AND     ØFDH
  11Ø9     32 4ØØ4                    LD      (4ØØ4H),A
  11ØC     18 8D                      JR      ZEROW
  11ØE     32 4ØØ7            FTDONE: LD      (CONTRL),A      ;turn off write gate
  1111     3E 4Ø                      LD      A,4ØH
  1113     C9                         RET
  1114     3A 111C            ADVTRK: LD      A,(TRACK)       ;get the current track
  1117     3C                         INC     A               ;advance track value
  1118     32 111C                    LD      (TRACK),A       ;update the track value
  111B     C9                         RET                     ;return with track value
  111C     ØØ                 TRACK:  Ø
  111D     ØØ                 DSIDE:  Ø
                                      .DEPHASE
  Ø472'                       ECODE   EQU     $
                                      END
```

COMPONENT LAYOUT/SCHEMATIC

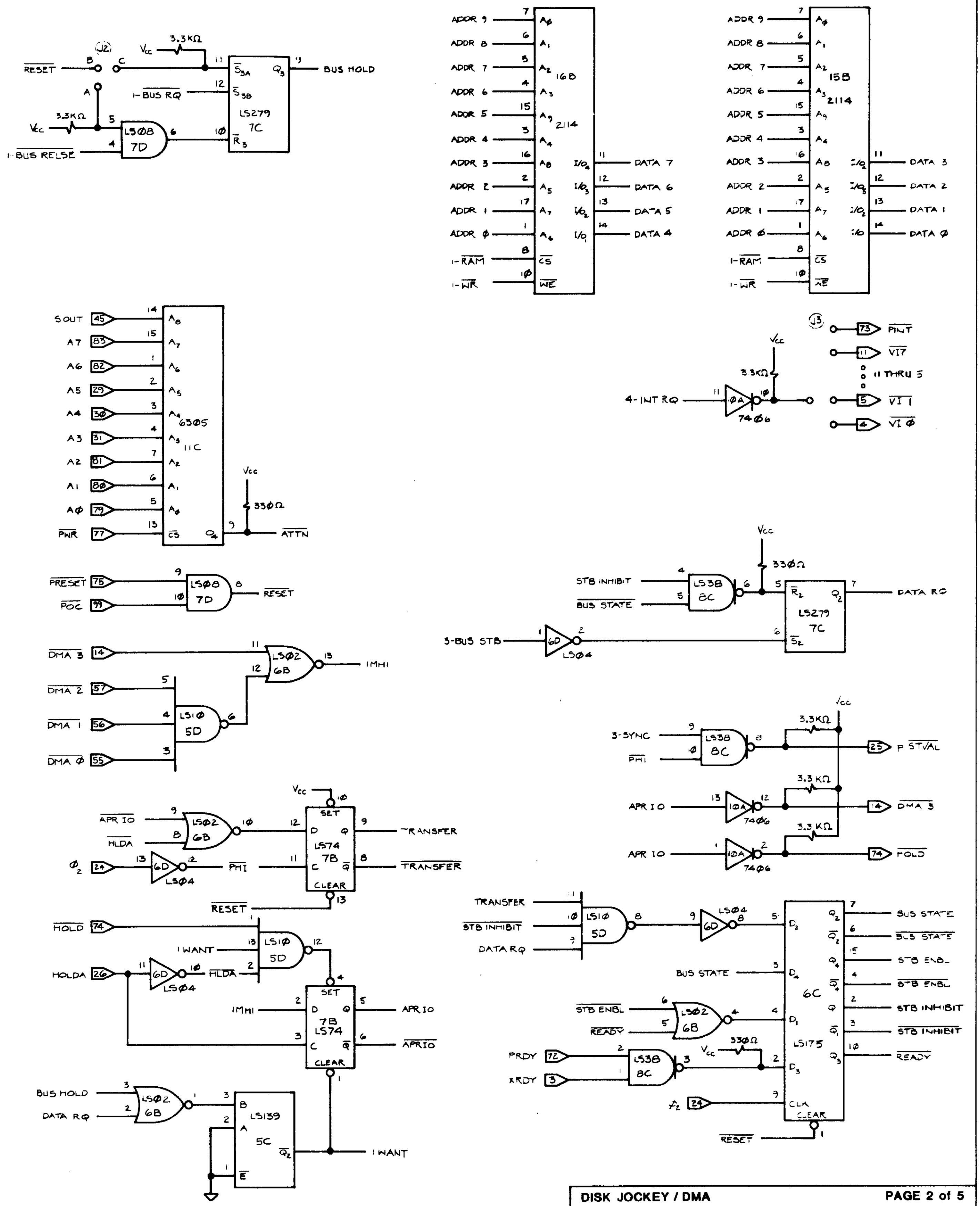


Disk Jockey / DMA Component Layout



DISK JOCKEY / DMA **PAGE 1 of 5**

CPU, BUFFERS, ROM & DECODE LOGIC

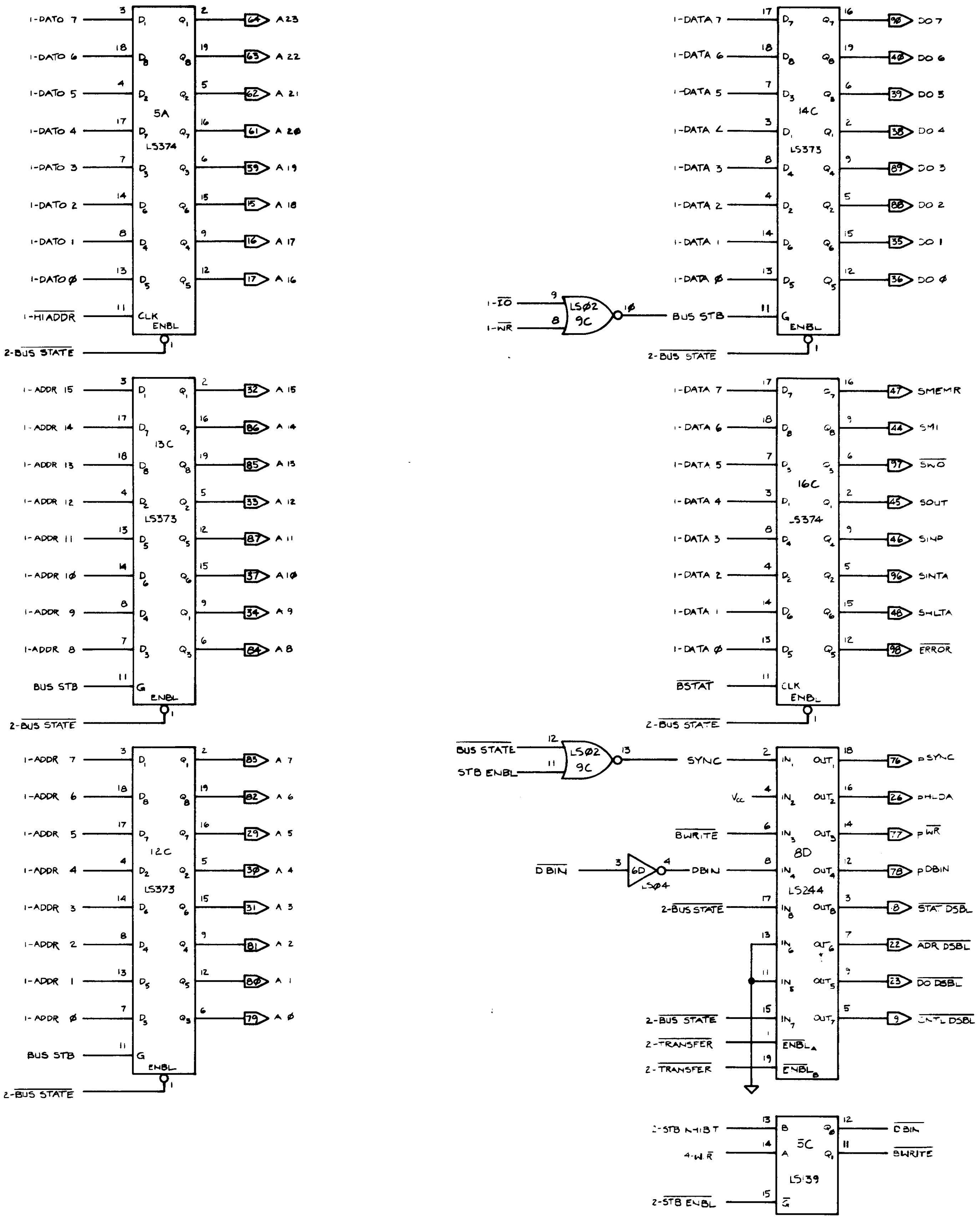


DISK JOCKEY / DMA

PAGE 2 of 5

BUS CONTROL LOGIC, BOOTSTRAP, I/O

© 1980 G. MORROW

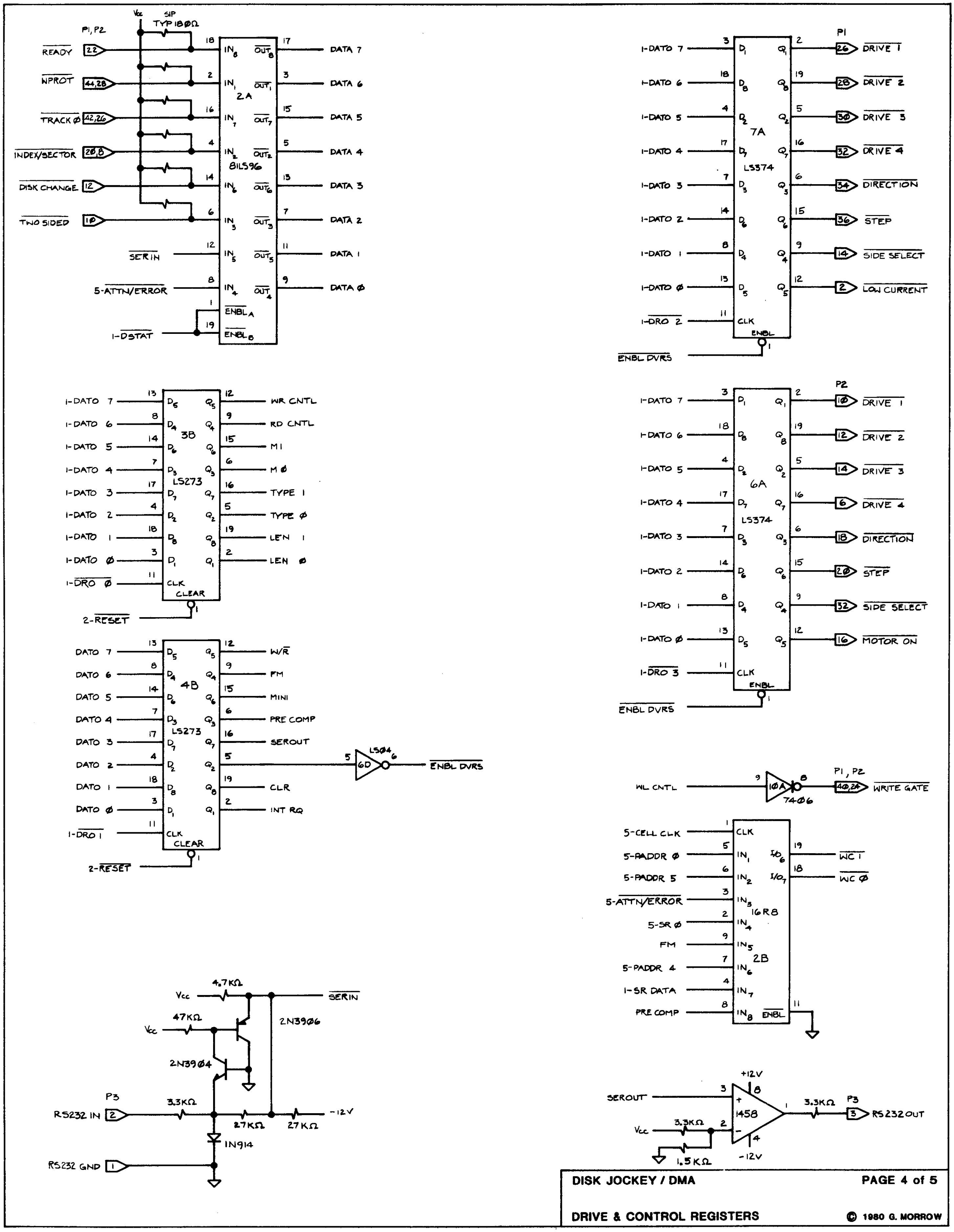


DISK JOCKEY / DMA

PAGE 3 of 5

BUS BUFFERS

© 1980 G. MORROW

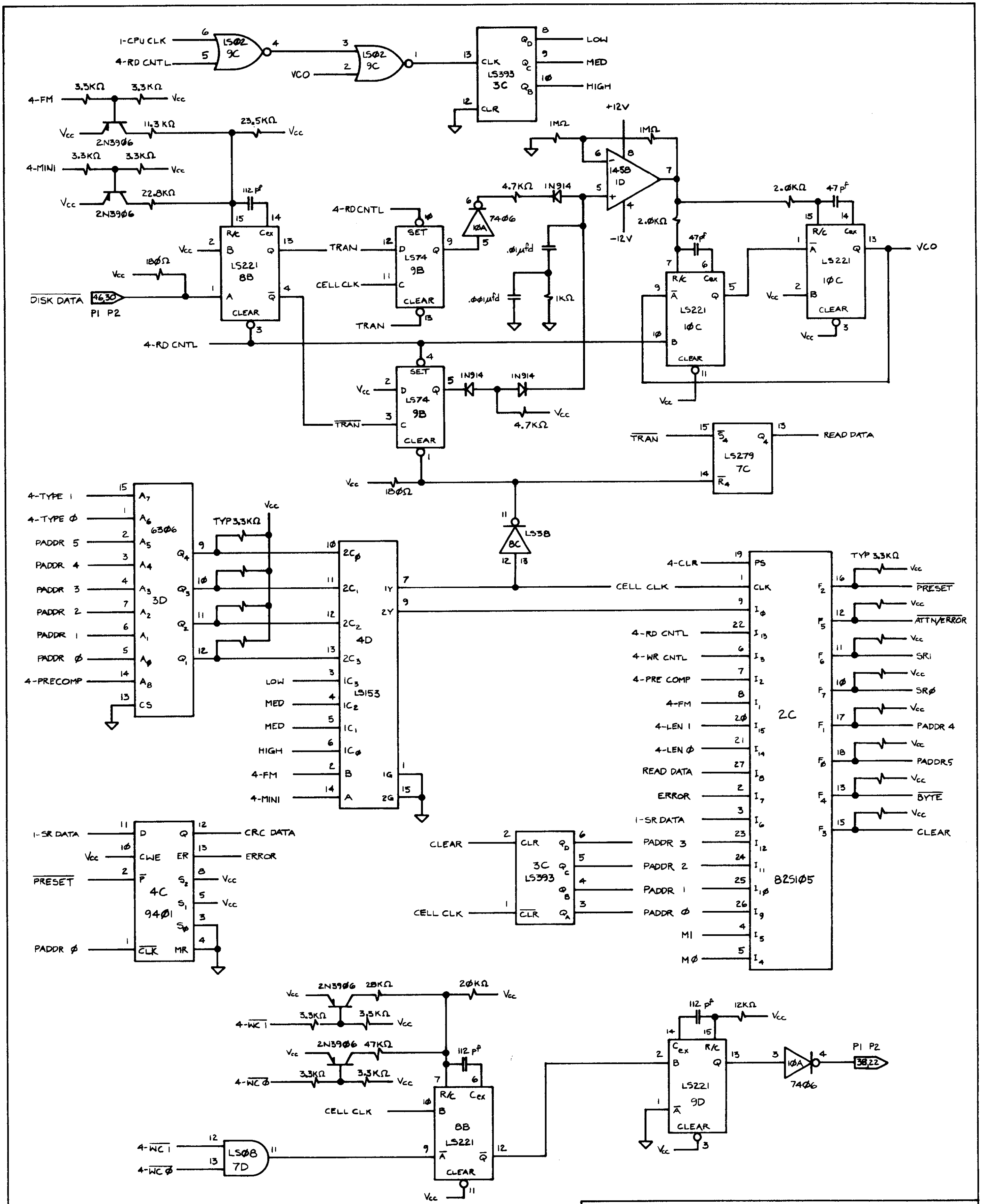


DISK JOCKEY / DMA

PAGE 4 of 5

DRIVE & CONTROL REGISTERS

© 1980 G. MORROW

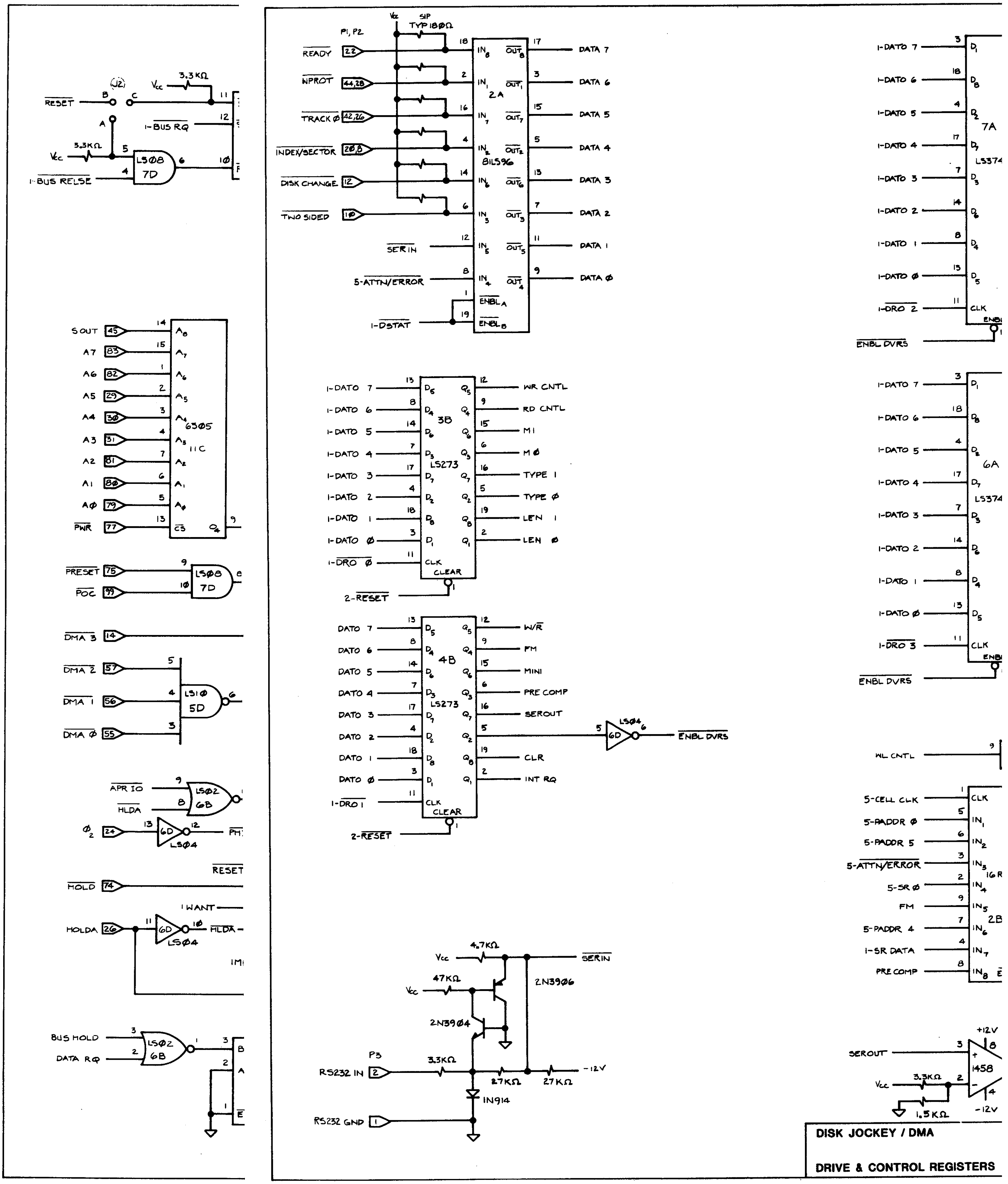


DISK JOCKEY / DMA

PAGE 5 of 5

DISK READ/WRITE LOGIC

© 1980 G. MORROW



DISK JOCKEY / DMA

DRIVE & CONTROL REGISTERS