Zendex

ZX-200A Single Board
Diskette Controller
ZX-200A Single Board
Diskette Controller

Pub.# 98-200A
Copyright (c) 1981 Zendex Corporation

6644 Sierra Lane
Dublin, CA  94566-9990
(415) 828-3000
TWX-910-389-4009

Specifications subject to change without notice. Intel, Intellec
and Multibus are trademarks of Intel Corporation.
WARRANTY

All products are warranted against defects in material and workmanship under normal and proper use and in their original unmodified condition. If found defective by Zendex Corp. within the terms of this warranty, Zendex Corp.'s sole obligation shall be to repair or replace at Zendex Corp.'s option the defective product. If Zendex Corp. determines that the product is not defective within the terms of this warranty, customer shall pay all costs of handling and return transportation. All replaced products become the property of Zendex Corp. As a condition of this warranty, customer must obtain a Zendex Corp. Return Material Authorization Number, and must return all products, transportation prepaid and insured, to Zendex Corp.'s Dublin, CA facility or other specified location.

Transportation charges for the return to customer shall be paid by Zendex Corp. within the contiguous United States only. These warranties outside the contiguous United States are limited to repair or replacement only and exclude all costs of shipping, customs clearance, and other related charges. Except for the express warranties stated above, Zendex Corp. disclaims all warranties on products, including all implied warranties of merchantability and fitness; and the stated express warranties are in lieu of all obligations or liabilities on the part of Zendex Corp. for damages, the use or performance with this product.

Warranty period is one (1) year from date of original shipment. Warranty registration card must be returned to Zendex for warranty to be in effect.

SERVICE POLICY

If a product should fail during the warranty period, it will be repaired for free. There will be a service charge for repair of a product after the warranty period. If a product exhibits misuse, negligence, or user misconnection, the failure will be treated as an out-of-warranty repair.

To return a product for in-warranty repair, first reverify that the unit is indeed at fault. Then, call the factory for Return Material Authorization (RMA) Number. The product should be carefully packaged and shipped prepaid using the provided RMA number on the outside of the package. Include a short statement of the malfunction, along with return address information, and the telephone number of a technical contact, in case the need arises.

For out of warranty repairs, a purchase order for repair charges must also be included.

Items should not be returned freight collect, as they will not be accepted. It is absolutely necessary to return products in the manner stated here, otherwise considerable delay will result in processing the return.

OUT OF WARRANTY REPAIRS

After the warranty has expired, or if no warranty registration is on file, any Zendex board product will be repaired or replaced (at Zendex's option) for a flat fee of $100, provided, in Zendex's opinion, the product has not been abused, misused, modified or damaged. Otherwise there will be a time and materials charge for returning it to original condition. This policy is subject to cancellation, modification, and change without notice.
# Table of Contents

## Chapter 1 - General Information

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Description</td>
<td>1</td>
</tr>
<tr>
<td>Equipment Supplied</td>
<td>2</td>
</tr>
<tr>
<td>Compatible Equipment</td>
<td>2</td>
</tr>
<tr>
<td>Specifications</td>
<td>3</td>
</tr>
</tbody>
</table>

## Chapter 2 - Preparation for Use

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Unpacking and Inspection</td>
<td>4</td>
</tr>
<tr>
<td>Installation Considerations</td>
<td>5</td>
</tr>
<tr>
<td>Jumper/Trace Cut Options</td>
<td>8</td>
</tr>
</tbody>
</table>

## Chapter 3 - Operating System

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

## Chapter 4 - Programming Information

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>16</td>
</tr>
<tr>
<td>Operational Modes</td>
<td>16</td>
</tr>
<tr>
<td>I/O Parameter Block</td>
<td>17</td>
</tr>
<tr>
<td>Disk Commands</td>
<td>25</td>
</tr>
<tr>
<td>Channel Commands</td>
<td>28</td>
</tr>
</tbody>
</table>

## Chapter 5 - Controller

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>33</td>
</tr>
<tr>
<td>Interrupts</td>
<td>33</td>
</tr>
<tr>
<td>I/O Base Address Selection</td>
<td>33</td>
</tr>
<tr>
<td>Drive Interface</td>
<td>33</td>
</tr>
<tr>
<td>Multibus Interface</td>
<td>36</td>
</tr>
<tr>
<td>Board Location Considerations</td>
<td>36</td>
</tr>
<tr>
<td>Controller Features</td>
<td>47</td>
</tr>
<tr>
<td>Controller Operation</td>
<td>44</td>
</tr>
</tbody>
</table>

## Appendix

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure A - ZX-85 Minimum Diskette System</td>
<td>46</td>
</tr>
<tr>
<td>Schematic</td>
<td></td>
</tr>
<tr>
<td>Comment Card</td>
<td></td>
</tr>
<tr>
<td>Bill of Materials</td>
<td></td>
</tr>
</tbody>
</table>
Figures

Figure 1  - The ZX-200A (photo)  
Figure 2  - Jumper Adaptions  
Figure 4.1 - IOPB Byte 1 - Channel Word  
Figure 4.2 - Summary of Byte One of the IOPB  
Figure 4.3 - IOPB Byte Two - Diskette Information  
Figure 4.5 - Port 7B 8B Read Result Byte  
Figure 5.1 - Ribbon Connections  
Figure 5.2 - Bus Access Timing  
Figure 5.3 - 8219 Set-up and Hold Timing  
Figure 5.4 - Implemented Serial Priority Network  
Figure A - ZX-85 Minimum Diskette System

Tables

Table 1 - ZX-200 Specifications  
Table 2 - Jumper Trace Cut Options  
Table 3.1 - Drive Number Assignments  
Table 3.2 - Drive Assignments for MDS-800 & MDS-230  
Table 4.1 - Op Code  
Table 4.2 - Unit Select, Bit 4-5  
Table 4.3 - Status Word  
Table 4.4 - Read Result Type  
Table 5.1 - Signals at Drive Interface J1  
Table 5.2 - Multibus Pin Assignments  
Table 5.3 - Multibus Signal Descriptions
Chapter 1 - General Information

1.1 Introduction

The ZX-200A is a single-board floppy disk controller which is able to interface from one to four eight-inch single density (FM) or double density (MMFM) disks to the multibus structure. The controller allows up to four single sided drives to be used, thus providing up to two megabytes of storage.

1.2 Description

The ZX-200A utilizes an 8085A microprocessor and 8257 DMA controller to perform all disk controller functions. Single or double density operation is under software control, and full emulation of standard Intel disk systems is possible. The ZX-200A can fully replace the Intel disk controller boards used in the MDS-800 and MDS 220/230 Development Systems, and can operate under ISIS-II software. The ZX-200A uses one Multibus card slot.

Figure 1 - ZX-200A
Zendex Corporation has been licensed by Micromation Corporation to build the ZX-200A using the Micromation MM-SBC-80F Multibus Floppy Disk Controller printed circuit artwork. Thus the layout, configuration, and features of the ZX-200A match the MM-SBC-80F. The Micromation manufactured board sold by Zendex was known as Model ZX-200.

1.3 Equipment Supplied

The following equipment is supplied with the ZX-200A:

- ZX-200A Hardware Reference Manual; with
- Schematic Diagram D200-01A; and
- Assembly Diagram

1.4 Compatible Equipment

CPU: The ZX-200A is compatible with any CPU, which is multibus compatible and is capable of multimaster operation, such as:
- ZX-85, 88, 86
- SBC-80/10B, 80/20, 80/24, 80/30
- SBC-86/12A
- ZX-80/05

Disk Drive: The ZX-200A is compatible with the following drives or their equivalents:
- Shugart Associates 800/801
- Memorex 550/552
- CDC 9404

Host Software: The ZX-200A is compatible with the following software:
- ISIS-II (Intel)
- CP/M for MDS
- Intel FORTRAN, BASIC
- UCSD PASCAL for CP/M
- Intel PLM, RMS-80 (Host)

Emulation: The ZX-200A emulates and can replace the following Intel disk controllers:
- Single Density - SBC-201, SBC-211, SBC-212, MDS-2DS, MDS-710
- Double Density - SBC-202, MDS-DDS, MDS-720
The ZX-200A when sold in combination with the Zendex ZX-730 Dual Drive Unit, is known as ZX-710/720 Mod 200A.

1.5 Specifications

### Table 1

| Operation Modes       | Single Density (FM) Ports 88H-8FH  
<table>
<thead>
<tr>
<th></th>
<th>Double Density (MMFM) Ports 78H-7FH</th>
</tr>
</thead>
</table>
| System Bus Interface  | Compatible with MULTIBUS specifications  
|                       | See Intel publication 9800083-02 |
| Floppy Disk Dive Interface | Accommodates Shugart 800 Series standard size disk drive (8 inch) |
| Power Requirement     | +5 volts at 2.75A (TYP) |
| Temperature           | 0 degrees to 40 degrees Centigrade |
| Humidity              | 0 to 90 percent RH non-condensing |
| Dimensions            | 12 inches long  
|                       | 6.75 inches wide  
|                       | 0.50 inches deep (one card slot) |
| Weight                | 14 ounces |
Chapter 2 - Preparation for Use

2.1 Introduction

This chapter provides information on preparing and installing the ZX-200A. Included are instructions on unpacking and inspection as well as information on installation procedure.

2.2 Unpacking and Inspection

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or waterstained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present, and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repair to a product damaged in shipment, contact Zendex, Inc. to obtain further instructions. A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be shipped.

Please fill out warranty card immediately and return to Zendex. This is your only way to receive regular ECO information and various updates that may become available.
2.3 Installation Considerations

The ZX-200A is intended to replace the two-board Intel Diskette Controller set (Channel and interface boards). It should be inserted into the card slot that would have held the Intel Diskette Interface Board. In order to ensure that the ZX-200A connects properly to the bus resolving multibus signals (BPRN/, BPRO/, BREQ/), it must occupy an odd-numbered card slot in the MDS-800 only. Series II and III can be in any slot.

The user should be aware of the fact that the Intellec MDS 220 has a single density drive mounted in the cabinet next to the CRT. This drive is controlled by the IO controller board, which is located not in the card cage, but at the back of the cabinet. The ZX-200A can still be plugged into the card cage, however, the original integrated single drive (ISD) will respond as :F4: under ISIS-II, rather than physical drive zero; and physical drive one will also respond as single density drive :F5:.

Before installing the ZX-200A, turn off all system power and remove the front panel (MDS 220/230), or the top panel (MDS 800). If the Intel channel and interface boards are installed, first remove the cable from the interface board, then remove both boards. Allow the ZX-200A to run both the single and double density systems. The Intel disk controller must be removed.

Before installing the ZX-200A, clean off the multibus and disk drive cable edge connector fingers with alcohol and for MDS-800 plug the ZX-200A controller into an odd-numbered slot of the card cage.
If the ZX-200A has been purchased in conjunction with a ZX-710/720 MOD 200 System, all cabling necessary to connect the ZX-200A to the ZX-730 disk drives is supplied. If the ZX-200A is purchased separately, a fifty-pin ribbon cable must be made. This must have a fifty-pin printed circuit connector in place at each end. This can connect the ZX-200A edge connector directly to four disk drives. The ZX-200A connector is pin for pin compatible with Shugart SA800. The cable supplied with the Intellec MDS is of no use and may be set aside.

Refer to Figure 2, for the jumper options required of a Shugart SA80IR for use with the ZX-200A.
Figure 2 - Jumper Adaptions for ZX-200A
2.4 Jumper/Trace Cut Options

Various jumper and trace cut options are offered to allow maximum flexibility when working with the ZX-200A. These are arranged as follows:

<table>
<thead>
<tr>
<th>Label</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1, W2</td>
<td>Drive Assignment Select</td>
</tr>
<tr>
<td></td>
<td>These two jumpers allow four possible choices with respect to the various drive name and density assignments under CP/M or ISIS. However, the present firmware does not make use of these jumper positions; therefore, the user may configure these jumpers as desired and allow the software to read their states and branch accordingly. Remember, the current firmware does not use these jumpers.</td>
</tr>
<tr>
<td></td>
<td>Factory Default</td>
</tr>
<tr>
<td></td>
<td>W1 in W2 in</td>
</tr>
<tr>
<td></td>
<td>Location</td>
</tr>
<tr>
<td></td>
<td>Near U12</td>
</tr>
<tr>
<td>W3</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td></td>
<td>W3 in</td>
</tr>
<tr>
<td></td>
<td>Near U43</td>
</tr>
<tr>
<td>(A-B-C)</td>
<td>Disk Drive Buffer Enable. This jumper allows the disk drive buffer to be either enabled permanently or to be enabled by the head load signal. Normally, the buffer should be enabled by the head load signal in order to qualify the disk control signals and avoid glitches on the drive select lines.</td>
</tr>
<tr>
<td></td>
<td>Head Load</td>
</tr>
<tr>
<td></td>
<td>Near U11</td>
</tr>
</tbody>
</table>
## Jumpers (continued)

<table>
<thead>
<tr>
<th>Label</th>
<th>Function</th>
<th>Factory Default</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV</td>
<td>Daisy Chain Priority Resolution. The ZX-200A can operate either with daisy chain or parallel resolution. To select daisy chain resolution, install a jumper plug from T to V. For parallel resolution, leave the jumper plug out.</td>
<td>Installed</td>
<td>Near U67</td>
</tr>
<tr>
<td>G, H, J, K, L, M, N, P, R, and S</td>
<td>System Interrupt. G and H plated through holes are the outputs from the interrupt generator circuitry. A jumper is installed at the factory from G and H, to N (INT2/). This is the standard configuration. G represents an interrupt associated with addresses 88H to 8FH, while H represents the same from 78H to 7FH.</td>
<td>G-H-N U71-U73</td>
<td>Between</td>
</tr>
</tbody>
</table>

### Trace Cut

<table>
<thead>
<tr>
<th>Label</th>
<th>Function</th>
<th>Connector</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-B, C-D</td>
<td>Disable Controller for 78H or 88H*. There are four plated through holes in the PCB next to the U43. Each pair of holes is connected with a trace on the component side when shipped from the factory. Cut the trace between the hole pair closest to U43 and the ZX-200A will not respond to addresses 78H to 7FH (usually, but not always for double density). Cut the other trace and the ZX-200A will not respond to addresses 88H to 8FH (always single density)</td>
<td>AB Connector CD Connector</td>
<td>Near U43 on component side</td>
</tr>
</tbody>
</table>
### Trace Cut (continued)

<table>
<thead>
<tr>
<th>Label</th>
<th>Function</th>
<th>Factory Default</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Enable</td>
<td><strong>Write Protect.</strong> There are two plated through holes between U12 and U13 which are connected by a trace on the solder side. If the trace is cut, the controller will not write to any of the disk drives connected to it.</td>
<td>Connected</td>
<td>Between U12, U13</td>
</tr>
<tr>
<td>G-H-J</td>
<td><strong>Advance Acknowledge</strong> There are three plated through holes between U69 and U70. The one closest to the 86-pin connector goes to the /AACK backplane signal and nowhere else. The other two holes are connected by a trace on the solder side and are connected to the /XACK backplane signal. Normally, the /XACK signal is the one that should be used, since the /AACK signal is being abandoned by various Multibus standards. However, should the user wish to use /AACK, cut the trace on the solder side and connect a jumper between the middle hole and the one closest to the 86-pin connector.</td>
<td>G-H connected H-J open</td>
<td>Between U69, U70</td>
</tr>
</tbody>
</table>

*Note:* In order to have the ZX-200A respond to addresses other than 78H to 7FH or 88H to 8FH, U43, the I/O MAP PROM, MAP23 at U43, must be changed by the user as desired. Zendex does not offer or support alternate PROMs. It should be kept in mind before changing PROMs that the factory PROM is compatible with CP/M and ISIS-II requirements.
Chapter 3 - Operating System

The ZX-200A will run under either the CP/M or ISIS-II operating systems. An important fact, which needs clarification at this point is how the disk drives are numbered according to recording density and operating system. Table 3-1 shows how the drive numbers are assigned.

<table>
<thead>
<tr>
<th>Physical Drive</th>
<th>ISIS Drive</th>
<th>CP/M Drive</th>
<th>Single Density</th>
<th>Double Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>:F0:</td>
<td>A:</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>:F1:</td>
<td>B:</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>:F2:</td>
<td>None</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>:F3:</td>
<td>None</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>:F4:</td>
<td>C:</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>:F5:</td>
<td>D:</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1 Drive Number Assignments

In order to bring up the operating system for a particular configuration, the double density system disk must be placed in Drive zero.

In a two drive Zendex or Intel double-density system, drive zero is on the right, drive one is on the left, and for the single density system, drive four is on the right, drive five is on the left. However, the situation becomes more complicated than
this, because a total of five drives could be utilized in an MDS-220 system, where the drive next to the CRT is controlled by the I/O controller in the MDS-220 chassis, and up to four external drives may be controlled by the ZX-200A.

The main thing to remember is that the logical number of a physical drive depends on the density of the diskette inserted in it at the time.

If a drive has a single-density diskette inserted in it, the only possible logical numbers for that drive are either :F4: (C:) or :F5: (D:). If it has a double-density inserted in it, the only possible numbers for that drive are: :F0:(A:), :F1:(B:), :F2:, :F3:. An attempt will be made to illustrate various drive number assignments for MDS-800, MDS-220 and MDS-230 Systems as a function of number of drives, density of diskette inserted. See the table on the following page.
Table 3.2
Drive Assignments for MDS-800, MDS-220

<table>
<thead>
<tr>
<th>Drive Assignment</th>
<th>Diskette</th>
<th>RH</th>
<th>LH</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS-800/ MDS-230</td>
<td></td>
<td>1</td>
<td>0</td>
<td>DD</td>
</tr>
<tr>
<td>MDS-800 Two-drive System</td>
<td></td>
<td>5</td>
<td>4</td>
<td>SD</td>
</tr>
<tr>
<td>MDS-800/ MDS-230</td>
<td></td>
<td>1</td>
<td>0</td>
<td>DD</td>
</tr>
<tr>
<td>MDS-800 Four-drive System</td>
<td></td>
<td>3</td>
<td>2</td>
<td>DD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>4</td>
<td>SD</td>
</tr>
<tr>
<td>MDS-220</td>
<td></td>
<td>1</td>
<td>0</td>
<td>DD</td>
</tr>
<tr>
<td>MDS-220 Five-drive System</td>
<td></td>
<td>3</td>
<td>2</td>
<td>DD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>4*</td>
<td>SD</td>
</tr>
<tr>
<td>MDS-220</td>
<td></td>
<td>1</td>
<td>0</td>
<td>DD</td>
</tr>
<tr>
<td>MDS-220 Three-drive System</td>
<td></td>
<td>5</td>
<td>4*</td>
<td>SD</td>
</tr>
</tbody>
</table>

*Single-density drive #4 always located in CRT chassis

The above illustration assumes that the ZX-200A is the only disk controller in the system aside from the IOC inside the MDS-Series II or III. The maximum number of drives in this case is five; four controlled by the ZX-200A and one controlled by the MDS-220 IOC.

**CAUTION:** If the CP/M is used with the MDS-220, and the MDS-220 is controlling two external drives, the MDS-220 drive is always single density and with ISIS is drive four. However, with CP/M this drive is invisible to the operating system and is not accessible.
Once the system configuration is well understood, all cabling is in place and the ZX-200A jumpers and trace cuts are understood and implemented, the operating system may be loaded. This is done according to the type of system being used, as follows:

**MDS-220/230 (All Series II or III)**

(1) Apply power to the MDS-220/230 and to the floppy disk drives. A prompt will appear on the CRT indicating that the system monitor has been entered.

(2) Insert the double density ISIS or CP/M system disk in Drive zero with the label facing up or left depending on horizontal or vertical drive mounting, and close the door.

(3) Press the system RESET button. A disk access will take place, the operating system gets loaded, and the sign-on message and prompt are displayed.

**MDS-800 and Zendex Models 835, 838**

(1) Turn the power on/off key to the on position. Apply power to the drives.

(2) Insert the double density ISIS or CP/M system disk in drive zero with the label facing up or left depending on horizontal or vertical drive mounting, and close the door.

(3) Depress the top of the BOOT push button on the MDS-800 panel. This enables the bootstrap PROM.
(4) Press the top of the RESET push button. A disk access will take place and the Interrupt two light on MDS-800 will be illuminated.

(5) After the Interrupt two light turns on, hit the space bar of the system terminal device. The Interrupt two light turns off. A Zendex system will sign on at this point.

(6) Press the bottom of the BOOT push button to disable the bootstrap PROM. The operating system is loaded from disk, and the sign-on message and prompt are displayed.
(1) After the reactor two filters are placed in the reactor chamber and the interturbine room. The filters are placed on the floor of the reactor chamber.

(2) After the reactor two filters are placed on the floor of the reactor chamber.
Chapter 4 - Programming Information

4.1 Introduction

The ZX-200A operates in an Intel Intellec MDS environment and responds to CPU commands issued over the multibus. The ZX-200A therefore, conforms to the software protocol of the Intel controller boards that it replaces.

4.2 Operational Modes

The ZX-200A operates in two modes:

(1) When it hasn't been selected to perform a disk related function, it is in the IDLE MODE. In this mode, it is constantly looping through a routine that checks the status of the disk drives. If a change in the status is noticed (a disk is removed or inserted, for instance), an interrupt is sent to the CPU to register the change.

(2) During program execution, The ZX-200A is selected

- to perform diskette reads and writes
- to be reset
- to stop prematurely a group of linked disk operations (in single density operation only)
- to render diskette drive status to the CPU
- to indicate the result of an operation to the CPU

Each of the above operations is initiated by a CPU input or output to a specific port. There are two base addresses, one for single density operation and one for double density. (Although
the ZX-200A reads or writes in both densities, Intel has separate floppy disk controllers for each density.)

The base address for single density operation is at 88H and the base address for double density operation is 78H.

4.3 I/O Parameter Block (IOPB)

The IOPB consists of ten (in single density operation) or seven (in double density operation) bytes of information which indicate the disk operation to be performed. The former has more bytes per IOPB because the original Intel single density controller permitted the linking of several IOPBs together. Several bits (see the description of the channel word) and bytes are present to accommodate this feature. The Intel double density controller does not include the linking feature.

The ten IOPB bytes are described on the following page. In the description, the first seven commands apply to both single and double density operation. The last three are used in single density only.
The IOPB is stored in main memory and thus is accessible by both the ZX-200A and the CPU.

The ten* IOPB bytes are:

- Byte 1: Channel Word
- Byte 2: Diskette Instruction
- Byte 3: Number of Records
- Byte 4: Track Address
- Byte 5: Sector Address
- Byte 6: Buffer Address (lower)
- Byte 7: Buffer Address (upper)
- * Byte 8: Block Number
- * Byte 9: Next IOPB Address (lower)
- * Byte 10: Next IOPB Address (upper)

The host CPU must write the IOPB to main memory. Once written, the host CPU instructs the ZX-200A as to the IOPB locations through I/O ports. These instructions are called channel commands and are explained later.

* Single density operations only.

**IOPB Byte 1 - Channel Word**

**Figure 4.1**

Channel Word

```
7 6 5 4 3 2 1 0 (LSB)
```

- Lock Override
- Random Format Sequence
- Interrupt Control
- Wait
- Branch on Wait
- Successor Bit
- Data Word Length
Bit 0, 1 - Wait, Branch on Wait

Single Density Mode (Port 88):

<table>
<thead>
<tr>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Immediately perform the current IOPB.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Idle for ten MS after which the Wait bit (bit 0) is examined. This loop is executed until the wait bit is reset.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Illegal</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>An unconditional jump to the 16-bit address pointed to by bytes nine and ten of the IOPB. The next IOPB to be performed must be resident at this address.</td>
</tr>
</tbody>
</table>

Double Density Mode (Port 78)

Bits zero and one are not used in the double density mode, since linked IOPBs are not supported in the double density mode. The ZX-200A Controller, therefore, will not wait and will execute only the correct IOPB.

Bit 2 - Successor

Single Density Mode (Port 88)

The successor bit (Bit 2) is reset if the current IOPB is the last (or only) one to be executed. Setting this bit indicates that a successor IOPB is to be executed; its address is in IOPB bytes nine and ten. The diskette controller will issue an interrupt when the operation is complete, bit two is reset, and bits four and five of this byte allow interrupt.

Double Density Mode (Port 78H)

Bit two is not used in the double density mode, since linked IOPBs are not supported.
Bit 3 - Data Word Length

Bit three must always be reset to a zero, to specify eight bit word length, since 16-bit word lengths are not allowed on the ZX-200A.

Bit 4, 5 - Interrupt Control

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>Bit 5</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Generates interrupt: (a) upon completion of an unchained diskette operation; (b) after the last operation in a chain of linked operations; or (c) upon detection of an error in any intermediate operation in a chain of linked operations.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Disable disk operation complete interrupt to CPU.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Generates disk operation complete interrupt to CPU after current operation even though it is not the last in a chain of linked IOPBs. This code is illegal in the double density mode.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Illegal Code.</td>
</tr>
</tbody>
</table>

Bit 6 - Random Format Sequence

A logical zero in this bit assigns sequential sector addresses when a disk is formatted. A logical one writes the sector addresses according to a pattern listed in a 52 byte memory buffer pointed to by bytes six and seven of the IOPB (see below).

Bit 7 - Lock Override

Single Density Mode (Port 88H)

When set (logical one), this bit prevents the "wait" bit from being set upon completion of the current operation specified in the IOPB. When reset (logical zero), this bit allows the ZX-200A to set the "wait" bit.
Double Density Mode (Port 78H)

This bit is never used in the double density mode, since the ZX-200A never sets the "wait" bit in double density.

The following figure summarizes byte one of the IOPB, the channel word.

Figure 4.2
Summary of Byte One of the IOPB

Single Density Mode (Port 88)

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Lock Override
Random Format Sequence
Interrupt Control
Data Word Length

Double Density Mode (Port 78)

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Random Format Sequence
Data Word Length
Interrupt Control

0 = Must be zero
X = Doesn't care

Figure 4.2
Byte Two - Diskette Instruction

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reserved
Unit Select
OP Code
Data Word Length
Table 4.1 Op Code

<table>
<thead>
<tr>
<th>Bits</th>
<th>Operation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 1 0</td>
<td>Operation</td>
</tr>
<tr>
<td>0 0 0</td>
<td>No operation</td>
</tr>
<tr>
<td>0 0 1</td>
<td>Seek: move the head to the track indicated in byte four of the IOPB.</td>
</tr>
<tr>
<td>0 1 0</td>
<td>Format track: write the address marks, gaps address fields and data fields on the track indicated in byte four of the IOPB. This type of format is determined by bit six of byte one as described below.</td>
</tr>
<tr>
<td>0 1 1</td>
<td>Recalibrate: move the head to track 00.</td>
</tr>
<tr>
<td>1 0 0</td>
<td>Read data: transfer N sectors (N indicated by byte three of the IOPB) from the disk to system RAM. The destination locations in memory start at the address pointed to by bytes six and seven of the IOPB. If the head is not already positioned over the track indicated by byte four of the IOPB, it is moved automatically. A CRC check is performed, which compares the two bytes of CRC written on the disk with the two generated from the data address mark and data field read.</td>
</tr>
<tr>
<td>1 0 1</td>
<td>Verify CRC: check the CRC of the indicated sector(s). The operation is similar to a READ; however, no data is transferred to memory.</td>
</tr>
<tr>
<td>1 1 0</td>
<td>Write Data: write N sectors (N specified in byte three of the IOPB) from the contents of memory that starts at the location indicated in IOPB bytes six and seven. As in read, the head is automatically moved to the desired track (from byte four of the IOPB) if it is not already there. Two bytes of CRC are generated and written to the disk. Note that multi-sector writes (N greater than 1) may not extend beyond a single track.</td>
</tr>
<tr>
<td>1 1 1</td>
<td>Write &quot;Deleted&quot; Data: write N sectors as described in the preceding operation except write a &quot;deleted data mark&quot; rather than the &quot;normal&quot; data mark.</td>
</tr>
</tbody>
</table>
Bit 3 - Data Word Length

Bit three must always be reset to a zero to specify eight-bit word length, since the ZX-200A will not handle 16-bit words.

Bits 4, 5 - Unit Select

<table>
<thead>
<tr>
<th>Single Density Mode (Port 88H)</th>
<th>Bits</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>Drive 0</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>Illegal</td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>Illegal</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>Drive 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Double Density Mode (Port 78H)</th>
<th>Bits</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>Drive 0</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>Drive 1</td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>Drive 2</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>Drive 3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2 - Unit Select, Bits 4 and 5

Bits 6, 7 - Reserved

These bits are not used and should be set to zero at all times.

IOPB Byte 3 - Number of Records

Byte 3 indicates the number of records (sectors) to be written/read. The number must be written in binary and may not exceed 26 in single density operation or 52 in double density operation. (Recall from the description of the Op Code above, that a read write operation may not extend beyond a single track.)
IOPB Byte 4 - Track Address
A binary code in byte 4 indicates the desired track number. Acceptable values are 00 to 4C hex (0 to 76).

IOPB Byte 5 - Sector Address
This byte specifies the first (or only) sector for operation. In single density mode specify 1-1A10 (2610) and in double density mode 1-3416 (5210). Bit five of this word must equal bit five of byte two in single density mode only.

IOPB Bytes 6, 7 - Buffer Address
These bytes specify the address of the disk buffer block for Read/Write/Format operations. Byte six is the least significant eight bits of the address while byte seven is the most significant portion.

Note: The next three IOPB Bytes (8, 9, 10) are used for single density applications only. They are used when a chain of IOPBs is to be executed. Note that this feature is not used frequently. (In fact, Intel dropped this feature from its double density controller.) If chaining is not used, these three bytes have no effect.

IOPB Byte 8 - Block Number (Single Density Only)
The specific number of the current IOPB is specified in this byte. Only six bits (5-0) are used. The block number allows the CPU to associate an I/O complete interrupt request from an intermediate link in a chain of IOPBs with the IOPB which actually caused the interrupt. The block number need only be initialized for linked IOPBs, since there can be no uncertainty when only a
single IOPB exists. This byte and bytes nine and ten are used for linked IOPBs.

**IOPB Byte 9 — Next IOPB (Lower Address)**

(Single Density Only)

The least significant byte of the 16-bit memory address of the next IOPB is entered in this byte.

**IOPB Byte 10 — Next IOPB (Upper Address)**

(Single Density Only)

The most significant byte of the 16-bit memory address of the next IOPB is entered in this byte. If the successor bit (two) of IOPB byte one is set (logical 1), the controller accesses the IOPB starting at this address upon completion of the current operation. If the successor bit is zero, it is assumed that the current IOPB is the last. The controller also looks at the "branch on wait" and "wait" bits in IOPB byte one. If both are set (logical 1), a jump to the IOPB at the address identified here is performed.

4.4 Disk Commands

The ZX-200A is capable of performing seven distinct operations: Recalibrate, Read, Write, Write Deleted Data, Record, Verify CRC, Seek, and Format. To begin any operation, the host CPU should output both bytes of the 16-bit memory address that point to byte one of the IOPB. The operation to be performed is specified in byte two of the IOPB. After the ZX-200A receives the upper byte of the IOPB address, it accesses the IOPB to interpret the operation to be performed and acquire the various parameters necessary to carry out the execution. The ZX-200A will set the
interrupt flip-flop after it has performed the operation or has halted operation due to errors.

The eight diskette operations are explained in more detail in the following paragraphs.

**Recalibrate (Opcode 3 of IOPB Byte 2)**

This operation will cause the selected unit's head to move over Track zero. Operation is mechanically verified by detectors in the drive itself.

**Seek (Opcode 1 of IOPB Byte 2)**

This operation will cause the selected drive to position its head over the specified track. Seek Track zero is tested for and, if issued, Recalibrate is executed instead.

**Read (Opcode 4 of IOPB Byte 2)**

This operation will return the specified number of data records to be written to the buffer, beginning at the given buffer address and continuing upward, starting with the track and sector given in the IOPB.

**Write Data (Opcode 6 of IOPB Byte 2)**

This operation is the same as Write Data except that each 128 byte data field is preceded with a deleted data address mark.

**Verify CRC (Opcode 5 of IOPB Byte 2)**

This operation will read the data records to verify the CRC check word. No data is transferred to the buffer.

**Format a Track (Opcode 2 of IOPB Byte 2)**

This operation is used to initialize a new disk or restore a "wiped-out" track. Prior track contents will be lost.

It should be noted here that a track can be "wiped-out" if the operator shuts off power to the diskette system while the
diskette is installed or the reset is hit while heads are loaded. Pop out diskettes BEFORE power-down or reset!

The order sector numbers that are assigned in the formatting of a track will depend on the state of the "Random Format" bit in byte one of the IOPB. If the random format bit is set, the pattern of sector addressing and initial sector data contents will be prescribed by the information in the buffer.

For Random Format the buffer contains the sector numbering, in order of assignment on the track, beginning with the first byte of the buffer and continuing through each odd numbered byte. The even numbered byte (one greater than sector address) will be the data to be initially written to all 128 bytes of the sector.

For example, if the buffer was constructed as:

<table>
<thead>
<tr>
<th>Byte</th>
<th>Contents (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>07</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>05</td>
</tr>
<tr>
<td>4</td>
<td>FF</td>
</tr>
</tbody>
</table>

the first physical sector of the track will be numbered seven with 20 as data in each of its 128 bytes. The second physical sector will be numbered five with all ones as initial data.

If the Random Format bit is reset, the order of sector numbering will be that of the physical sector and the initial data written to all sectors will be that of byte one of the buffer.

**No Operation (Opcode 0 of IOPB Byte 2)**

The No-Op instruction causes the ZX-200A to execute a read drive status and is intended to verify that the controller is functioning.
4.5 Channel Commands

Diskette status, result, and IOPB information are communicated over a set of I/O channels and are called, as a group, the Channel Command.

Once a proper IOPB has been constructed in main memory the controller must be informed of the IOPB address via Channel Commands. Upon completion, or interrupt, result data is available with error indications by way of Channel Commands.

When the Write IOPB Address Upper is executed the disk system will commence the operation specified in the IOPB. Therefore, the lower portion of address and the entire IOPB must have been properly constructed before this Channel Command is executed.

Out Port 7F 8F - Reset

This output channel command causes all logic in the ZX-200A to be reset to an initialized state. This command is intended to clear hang-ups.

Out Port 79 89 - Write IOPB Address Lower

This channel command outputs the low byte of the 16-bit address pointing to byte one in the IOPB.

Out Port 7A 8A - Write IOPB Address Upper

This channel command outputs the high byte of the IOPB's 16-bit address. This command also causes the ZX-200A to begin execution of the IOPB.
Out Port 8B - Stop

The diskette controller will stop operation AFTER completing the current IOPB instruction. It will not proceed to the next IOPB in a link. This channel command has no meaning in double density mode since linked operations are not performed.

In Port 78 88 - Status Input

This input channel command causes the ZX-200A to return the drive and controller ready status.

Table 4.3 - Status Word

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ready Status of Drive 0</td>
</tr>
<tr>
<td>1</td>
<td>Ready Status of Drive 1</td>
</tr>
<tr>
<td>2</td>
<td>Interrupt Pending</td>
</tr>
<tr>
<td>3</td>
<td>Controller Present</td>
</tr>
<tr>
<td>4*</td>
<td>Double Density Controller Present</td>
</tr>
<tr>
<td>5*</td>
<td>Ready Status of Drive 2</td>
</tr>
<tr>
<td>6*</td>
<td>Ready Status of Drive 3</td>
</tr>
<tr>
<td>7</td>
<td>Hard Disk Present</td>
</tr>
</tbody>
</table>

*Active in Port 78 only (double density mode)

Bits zero, one, five and six, allow the host to determine whether the target drive is ready (bit equals one) or not (bit equals zero).
In Port 79 89 - Read Result Type

This input channel command will return a two bit result type (bits 0 and 1) and, for 201 mode only, a block number in bits two through seven. The result type is decoded as:

<table>
<thead>
<tr>
<th>Bit</th>
<th>1</th>
<th>0</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>I/O complete. Result byte contains error bits.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>Single Density mode only. I/O complete, result byte contains linked error bits.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>Result byte has drive ready status.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>Reserved.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4 Read Result Type

The Result Type command must be issued to clear the system interrupt and diskette controller interrupt pending bits (which toggle together) in the status word.

In Port 7B 8B - Read Result Byte

The channel command causes the ZX-200A to return eight bits of operation results. The proper interpretation of the result byte depends upon the result type. For result types 00 and 01:

Figure 4.5

Port 7B-8B Read Result Byte for Result Types 00 and 01
If the host executes a 'read result byte' channel command, the diskette channel will return the result word on the system data bus. The bits are defined as follows:

**Not Ready:**

Bit seven indicates the selected unit was not ready or the selected unit changed to a not ready status during operation.

**Write Error:**

Bit six indicates that during a write operation a condition existed which precluded data integrity. An example of a condition causing this error is an attempt to write a 'not ready FDD.'

**Write Protect:**

Bit five indicates the selected drive contains a diskette, which is not write enabled. This condition is checked on format a track, write data (with data address marks) and write data (with deleted data address marks) operations.

**Data Overrun/Underrun Error:**

Bit four indicates that the ZX-200A controller was not able to service a byte transfer request from the drive before the next request occurred. The data byte is lost.

**Address Error:**

Bit three indicates that the disk address received from the CPU is invalid; that is:

- track address > 76\(_{10}\)
- sector address = 00
- sector address > 52\(_{10}\) (202 mode)
- sector address > 26\(_{10}\) + number of records > 52\(_{10}\) (202 mode)
- sector address + number of records > 26\(_{10}\) (201 mode)
Seek Error:

Bit two indicates that at the completion of a head movement sequence the head is not positioned over the expected track. This bit indicates the controller and/or FDD are malfunctioning, and a recalibrate diskette operation should be performed. Seek error can occur during any diskette operation.

CRC Error:

Bit one indicates the CRC characters generated during a read data or verify CRC operation were not the same as the two CRC characters appended to the data field when it was originally written on the diskette.

Deleted Record:

Bit zero indicates that a sector addressed during a read data or verify CRC operation was preceded by a deleted data address mark.

Two other error conditions are provided when more than one error bit is true. They are:

ID CRC Error:

If the address error and CRC error (bits 3 and 1, respectively) are true, this indicates the CRC characters generated during the reading of an ID field were not the same as the CRC characters appended to the field when it was written by a format track operation.

Data Mark Error or No Address Mark:

If the address error, seek error, and CRC error (bits 3, 2 and 1, respectively) are true, this indicates no address mark or a data mark error was encountered. This usually means the track has not been formatted.
Chapter 5 - Controller Operation

5.1 Introduction
This chapter is intended to give a brief description of the ZX-200A hardware operation and its external interfaces.

5.2 Interrupts
The ZX-200A will generate interrupt requests when allowed by certain channel commands (see Chapter 4). Any interrupt (INT0 through INT7) may be used, but most operating systems, especially ISIS, will require INT2/. This may be selected via jumper (see Chapter 2).

5.3 I/O Base Address Selection
The I/O base address is fixed by U43, a bipolar PROM.
For microcomputer development systems using ISIS the required address is 78H - 7FH for logical drives :F0:, :F1:, :F2: and :F3:, which are all double density.
The required address is 88H - 8FH for logical drives :F4: and :F5:, which are both single density, and independently add reusable.

5.4 Drive Interface
The ZX-200A has been designed to interface readily with the Shugart SA800. The 50-pin edge connector is pinned alike with the SA800, thus allowing simple ribbon connectors. More than one
drive may be used by simply paralleling it on a common connector and wire system as shown in the following figure.

Figure 5.1 Ribbon Connection

Line terminators must be installed for each signal line at the last drive unit to use it. In the figure most lines (except DS1 through DS3) will be terminated at drive one.

The table on the following page lists the signals available at the drive interface connector J1.
<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1/-DS4/J1-26</td>
<td>A low state selects the drive. When active DS1/-DS4/ allows drives 0 through 3 to accept the remaining drive input signals and to gate its output back to the ZX-200A.</td>
</tr>
<tr>
<td>WRITE ENABLE/</td>
<td>A low state will allow the data to be written on the diskette. When this signal is inactive the write electronics are disabled and the drive reads data from the diskette.</td>
</tr>
<tr>
<td>J1-40</td>
<td></td>
</tr>
<tr>
<td>STEP/J1-36</td>
<td>A signal pulsing low will cause the head to step one track, for each low-going, in the direction determined by DIR/.</td>
</tr>
<tr>
<td>DIR/J1-34</td>
<td>When this line is low the step signal will cause the head to move toward the track 76 (step in), and when high the head to step out toward the outermost track 00.</td>
</tr>
<tr>
<td>WR DAT/J1-38</td>
<td>This is the composite data/clock serial write signal. A high-to-low transition on this line indicates a bit to be written on the diskette.</td>
</tr>
<tr>
<td>RDY J1-22, 4,6,8</td>
<td>A low on this line indicates the selected drive is ready. This is a radial ready circuit.</td>
</tr>
<tr>
<td>WR PROT/J1-44</td>
<td>An active low on this line indicates a write protected diskette is installed in the selected drive.</td>
</tr>
<tr>
<td>TRKO/J1-42</td>
<td>An active low indicates the selected drive's head is positioned over track 00.</td>
</tr>
<tr>
<td>INDEX/J1-20</td>
<td>A low going pulse is passed on this line that is coincident with the index hole in the diskette.</td>
</tr>
<tr>
<td>READ DATA/J1-46</td>
<td>The composite data and clock signal generated during a diskette read operation. A high-to-low going transition indicates a clock or data one bit.</td>
</tr>
<tr>
<td>LOW CURRENT/J1-2</td>
<td>Signal to the drive to reduce write current through head. Active low for track 43.</td>
</tr>
<tr>
<td>FAULT RESET/J1-4</td>
<td>Reset signal to drive to clear fault indicator.</td>
</tr>
</tbody>
</table>
Table 5.1 Signals at Drive Interface J1 (continued)

<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAULT/ J1-6</td>
<td>Fault detected by drive. Signal is input to controller to cause fault routines to execute.</td>
</tr>
<tr>
<td>TWO SIDED/ J-10</td>
<td>Signal from FDD to indicate presence of a two sided media.</td>
</tr>
<tr>
<td>SIDE SELECT/ J1-14</td>
<td>Signal from FDC to FDD to select which side of a two sided media operation is to be performed.</td>
</tr>
</tbody>
</table>

Note: 1-49 odd are all signal grounds.

5.5 Multibus Interface

The ZX-200A communicates with the master CPU over the bus through the 86-pin connector Pl. Table 5.1 defines Multibus signals used and Figures 5.1 and 5.2 describe signal timings.

5.6 Board Location Considerations

The ZX-200A is a Multibus bus master and as such must be located in a backplane slot which provides for the BPRN/, BREQ/, BUSY, and BCLK signals. Other bus master cards in a system will consist of the CPU and DMA boards. Provision must be made for the various bus masters to communicate bus requests and grants through card arrangements and connections of the above bus signals. The most common priority resolution scheme is the serial type and is typically used in systems with less than four bus masters. A jumper should be used at wire wrap position T-V to provide BPRO/ output in serial priority schemes.

Figures 5.1 and 5.2 on the following page show bus access timing and 8219 set-up and hold timing.
Figure 5.2  Bus Access Timing

BCLK
(P1-13)

BCR
(8219-25)

BREQ
(P1-18)

BPRO/
(P1-16)

BUSY/
(P1-17)

ADEN
(8219-19)

Figure 5.3  8219 Set-up and Hold Timing

XCY
(8219-5)

MRDC/ or MWTC/
(P1-19 OR P1-20)

XACL/
(P1-23)

XCY
(8219-4)
# Table 5.2 Mutibus Pin Assignments

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Signal Ground</td>
<td>2</td>
<td>GND</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>3</td>
<td>+5</td>
<td>+5 VDC</td>
<td>4</td>
<td>+5</td>
<td>+5 VDC</td>
</tr>
<tr>
<td>5</td>
<td>+5</td>
<td>+5 VDC</td>
<td>6</td>
<td>+5</td>
<td>+5 VDC</td>
</tr>
<tr>
<td>7</td>
<td>+12</td>
<td>+12 VDC</td>
<td>8</td>
<td>+12</td>
<td>+12 VDC</td>
</tr>
<tr>
<td>9</td>
<td>-5</td>
<td>-5 VDC</td>
<td>10</td>
<td>-5</td>
<td>-5 VDC</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
<td>Signal Ground</td>
<td>12</td>
<td>GND</td>
<td>Signal Ground</td>
</tr>
</tbody>
</table>

## Power Supplies

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>BCLK/</td>
<td>Bus Clock</td>
<td>14</td>
<td>INIT/</td>
<td>Initialize</td>
</tr>
<tr>
<td>15</td>
<td>BPRN/</td>
<td>Bus Priority In</td>
<td>16</td>
<td>BREQ/</td>
<td>Bus Request</td>
</tr>
<tr>
<td>17</td>
<td>BUSY/</td>
<td>Bus Busy</td>
<td>18</td>
<td>MWTC/</td>
<td>Memory Write Command</td>
</tr>
<tr>
<td>19</td>
<td>MRDC/</td>
<td>Memory Read Command</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>IORC/</td>
<td>I/O Read Command</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>XACK/</td>
<td>Transfer Acknowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>AACK/</td>
<td>Advance Acknowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Bus Controls

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>BREN/</td>
<td>Byte High Enable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>CBQ/</td>
<td>Common Bus Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>CCLK/</td>
<td>Constant Clock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>INTA/</td>
<td>Interrupt Acknowledge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>INT6/</td>
<td>Interrupt Requests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>INT4/</td>
<td>Interrupt Requests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>INT2/</td>
<td>Interrupt Requests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>INT0/</td>
<td>Interrupt Requests</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Address

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>ADRE/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>ADRC/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>ADRA/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>ADRB/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>ADR8/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>ADR2/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>ADR3/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>ADR4/</td>
<td>Address Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Data Lines

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>59</td>
<td>DATE/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>DATC/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>DATA/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>DATB/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>DAT6/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>DAT4/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>DAT2/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>DAT0/</td>
<td>Data Lines</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Power Supplies

<table>
<thead>
<tr>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
<th>PIN</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>GND</td>
<td>Signal Ground</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>-10</td>
<td>-10 VDC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>-12</td>
<td>-12 VDC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>+5</td>
<td>+5 VDC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>GND</td>
<td>Signal Ground</td>
<td>85</td>
<td>GND</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>84</td>
<td>+5</td>
<td>+5 VDC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

38
<table>
<thead>
<tr>
<th>Signal Mnemonic</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AACK/</td>
<td>Advanced Acknowledge. This signal is an advanced indication that the requested Read or Write operation will be completed in a specified time (tAACK).</td>
</tr>
<tr>
<td>ADR0/-ADRF/</td>
<td>Address. These 16 lines are used to transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.</td>
</tr>
<tr>
<td>ADR10/-ADR13</td>
<td>Address Extension. These four lines are appended to the address to allow accessing of megabyte memories.</td>
</tr>
<tr>
<td>BCLK/</td>
<td>Bus Clock. The high-to-low transition of BCLK/ is used to synchronize bus contention resolution circuits. BCLK/ is asynchronous to the CPU clock. It has a minimum period of 100 nanoseconds and a 35% duty cycle. BCLK/ may be slowed, stopped, or single stepped for troubleshooting.</td>
</tr>
<tr>
<td>BHEN/</td>
<td>Byte High Enable. Indicates use of data lines DAT8-F.</td>
</tr>
<tr>
<td>BPRN/</td>
<td>Bus Priority In. Indicates to a particular master module that no higher priority module is requesting use of the bus. BPRN/ is synchronized with BCLK/. This signal is not bussed on the backplane.</td>
</tr>
<tr>
<td>BPRO/</td>
<td>Bus Priority Out. Used with serial (daisy chain) bus priority resolution schemes. BPRO/ is passed to the BPRN/ input of the master module with the next lower busy priority. BPRO/ is synchronized with BCLK/. This signal is not bussed on the backplane.</td>
</tr>
<tr>
<td>CBRQ/</td>
<td>Common Bus Request. Indicates a master module, not currently in control, requests use of the bus.</td>
</tr>
<tr>
<td>BUSY/</td>
<td>Bus Busy. This signal is driven by the bus master currently in control to indicate that the bus is in use. BUSY/prevents all other master modules from gaining control of the bus. BUSY/ is synchronized with BCLK/.</td>
</tr>
<tr>
<td>CCLK/</td>
<td>Constant Clock. Provides a clock signal of constant frequency for unspecified general use by modules on the bus. CCLK/ has a minimum period of 100 nanoseconds and a 35% to 65% duty cycle.</td>
</tr>
<tr>
<td>Signal Mnemonic</td>
<td>Functional Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>DAT0/-DATF/</td>
<td>Data. These 16 bi-directional lines transmit or receive information to or from a memory location or I/O port. DATF/ is the most significant bit. In eight bit systems only lines DAT0/-DAT7/ are used, in which case DAT7/ is the most significant bit.</td>
</tr>
<tr>
<td>INH1/</td>
<td>Inhibit RAM. Prevents RAM devices from responding to the memory address on the address lines. INH1/ effectively allows ROM devices to override RAM devices when ROM and RAM are assigned the same address space. INH1/ may also be used to allow memory mapped I/O devices to override RAM.</td>
</tr>
<tr>
<td>INH2/</td>
<td>Inhibit ROM. Prevents ROM devices from responding to the memory address on the address lines. INH2/ effectively allows start-up software such as ROM based bootstrap programs to override another ROM device when the two ROMs are assigned the same address space. INH2/ may also be used to allow memory mapped I/O devices to override ROM.</td>
</tr>
<tr>
<td>INIT/</td>
<td>Initialization. Resets entire system to a known internal state. INIT/ may be driven by one bus master or by an external source such as a front panel reset switch.</td>
</tr>
<tr>
<td>INTO/-INT7/</td>
<td>Interrupt. Each of these eight lines causes the master processor to generate INTA as defined below. INTO/ has the highest priority; INT7/ has the lowest priority as assigned by an interrupt priority resolution network.</td>
</tr>
<tr>
<td>IORC/</td>
<td>I/O Read Command. Indicates that the address of an input port has been placed on the address lines and that the data is to read from the data lines. IORC/ is asynchronous with BCLK/.</td>
</tr>
<tr>
<td>MRDC/</td>
<td>Memory Read Command. Indicates the address of a memory location has been placed on the address line and that a data word (eight or 16 bits) is to be read from the data lines. MRDC/ is asynchronous with BCLK/.</td>
</tr>
<tr>
<td>MWTC/</td>
<td>Memory Write Command. Indicates that the address of a memory location has been placed on the address lines and that a data word (eight or 16 bits) has been placed on the data lines. MWTC/ specifies that the data word is to be written into the addressed memory location. MWTC/ is asynchronous with BCLK/.</td>
</tr>
</tbody>
</table>
Table 5.3 - Multibus Signal Descriptions (continued)

<table>
<thead>
<tr>
<th>Signal Mnemonic</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XACK/</td>
<td>Transfer Acknowledge. The required response of memory location or I/O port to indicate that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the data lines. XACK/ is asynchronous with BCLK/.</td>
</tr>
</tbody>
</table>

Figure 5.3 shows an implemented serial priority network. In this type of arrangement the bus master with the highest priority can be identified by its pin 15 (BPRN/) being grounded.

![Figure 5.3 Implemented Serial Priority Network](image)

In the above figure, the master in slot J1 will drop BPRO/ low to pass priority grant on to J2 if the master in J1 is NOT requesting the bus. BUSY/ will be high if no master is currently using the bus. If the master J2 doesn't need the bus, it will drop its BPRO/ low to cause J3 BPRN/ to go low and thus grant a J3 request. If J2 requests the bus it will raise its BPRO/ high and wait until its BPRN/ goes low, BUSY/ goes high, and BCLK/ does a negative transition. J2 will then latch BUSY/ low and that will disallow even higher priority master requests until J2 drops its use of the bus.
Higher speed and bus contention resolution schemes involving four or more masters use a parallel priority network. BREQ/ signal is output to an SN74148 (typical) priority encoder. Then
an SN74S138 (typical) decoder outputs a master's BPRN/ for input grant.

5.7 Controller Features

The Zendex ZX-200A Multibus floppy disk controller emulates and enhances the Intel iSBC 202 Double Density and iSBC 201 Single Density floppy disk controllers. A single CONTROLLER card can replace both sets of boards (each Intel controller consists of two boards, the channel and interface cards) in an Intel Intellec MDS system and interfaces as many as four floppy disk drives. These drives can be single or double sided with single or double density format diskettes. The operator does not need to enter any commands to indicate the recording density of the diskette to be read from or written to.

From the system's perspective, the CONTROLLER appears as two separate controllers, one for single density diskettes and one for double density diskettes. Typically, each of the controllers is accessed via a different port address. Usually, these addresses fall in the following ranges:

Single density: 88 - 8F hex
Double density: 78 - 7F hex

The resident CONTROLLER firmware decodes the address to determine the recording density of the diskette. However, there are circumstances where the address range for the single density controller is 78 - 7FH. The firmware has been written to accommodate this configuration as well. (See Note on following page.)
Note: the first revision of the CONTROLLER firmware does not support single density operation from port 78H.

Data to be read from or written to the disk is fully buffered. The CONTROLLER contains 1K of static RAM temporarily storing the data for transfer. This means two things to the system designer: (1) the chance of a data overrun or underrun is completely removed, and (2) the controller cannot get control of the system bus in time for a byte of data read from the disk to be transferred to system RAM (data is transferred on a byte by byte basis). Consequently, that byte disappears as the next one is read from the disk and moved to RAM (assuming the controller gets control of the bus in time to transfer this byte). Data underrun is similar, but occurs during a disk write operation. With the CONTROLLER, the entire sector (128 bytes) is written to the onboard RAM independently of the system bus. Data is transferred to system RAM with a DMA controller. If this gets interrupted by a higher priority component, the DMA controller keeps a record of the last location transferred, so that when it has control of the bus again, it can begin where it left off.

The rate of data exchange is 250K bits per second in single density and 500K bits per second in double density. Single density recording uses the standard IBM 3740 format. This format uses the frequency modulation (FM) recording technique and specifies 26 sectors of 128 bytes of data per track. In double density, MMFM (modified-modified frequency modulation) recording is used with 52 sectors of 128 bytes per track.
5.8 Controller Operation

The ZX-200A CONTROLLER firmware responds to ten I/O addresses output by the CPU. These addresses typically appear between 78 - 7FH for double density operation, 88 - 8FH for single density operation. In some circumstances 78 - 7H is also used for single density. The CONTROLLER firmware can interpret these addresses as well. (Note: these ports can be changed by inserting a different I/O address decode PROM u43). Of the eight available ports in each range, only five are used. The CPU selects one of seven disk-related operations by outputting to the port associated with that operation.

The CONTROLLER operates as a system within the system. It contains a 8085A processor, which executes the requested disk operation via a memory-map. For instance, a read sector command from the CPU requires a number of processes be performed in addition to the read operation (for example, move head to track, synchronize with the disk, read data, read and compare CRC, etc.) Each of these functions is enabled by an output to the port within the CONTROLLER dedicated to it.

When the command requests a disk read or write, two DMA operations follow. First, the seven or ten bytes of the I/O parameter block (IOPB) are moved to a 1K RAM buffer on the CONTROLLER RAM buffer. A resident 8257 DMA controller performs this task. Once it has control of the system bus from the CPU, the whole content of the data buffer is transferred to system RAM for a disk read or vice versa for a disk write. The requisite bus signals are generated to access memory. If the controller is
interrupted by a higher priority component in the system, the 8257 "remembers" the last byte transferred. When it has regained control of the bus, the next byte is transferred.

The CONTROLLER operates in two modes:

(1) when it hasn't been selected to perform a disk related function, it is in the IDLE MODE. In this mode, it is constantly looping through a routine, which checks the status of the disk drives. If a change in the status is noticed (a disk is removed or inserted, for instance), an interrupt is sent to the CPU to register the change.

(2) During program execution, the CONTROLLER is selected:
- to perform diskette reads and writes
- to be reset
- to stop prematurely a group of linked disk operations (in single density operation only)
- to render diskette drive status to the CPU
- to indicate the result of an operation to the CPU

Each of the above operations is initiated by a CPU output to a specific port. There are two base addresses, one for single density operation and one for double density. (Although the ZX-200A CONTROLLER reads or writes in both densities, Intel has separate floppy disk controllers for each density.)

**Note:** The base port for the Intel single density controller is address 88H; the base port for the double density controller is at 78H.
Appendix
Figure A - ZX-85 Minimum Diskette System

ZX-85 Minimum Diskette System
### ZX-200A MAP 23 LISTING

<table>
<thead>
<tr>
<th>Addr</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0010</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0020</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0030</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0040</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0050</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0060</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0070</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0080</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0090</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00A0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00B0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00C0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00D0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00E0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>00F0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Addr</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0110</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0120</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0130</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0140</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0150</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0160</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0170</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0180</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>0190</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01A0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01B0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01C0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01D0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01E0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>01F0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
### ZX-200A MAP PROM LISTING

#### MAP 20

<table>
<thead>
<tr>
<th>Addr</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
</tr>
<tr>
<td>0010</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
</tr>
<tr>
<td>0020</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0030</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0040</td>
<td>DF</td>
<td>DF</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DD</td>
<td>DD</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DB</td>
<td>DA</td>
</tr>
<tr>
<td>0050</td>
<td>DA</td>
<td>DA</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
</tr>
<tr>
<td>0060</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0070</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0080</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
</tr>
<tr>
<td>0090</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
<td>EE</td>
</tr>
<tr>
<td>00A0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00B0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00C0</td>
<td>DF</td>
<td>DF</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DE</td>
<td>DD</td>
<td>DD</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DB</td>
<td>DA</td>
</tr>
<tr>
<td>00D0</td>
<td>DA</td>
<td>DA</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
<td>DD</td>
</tr>
<tr>
<td>00E0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00F0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
</tbody>
</table>

#### MAP 21

<table>
<thead>
<tr>
<th>Addr</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>61</td>
<td>61</td>
<td>DD</td>
<td>61</td>
<td>61</td>
<td>DD</td>
</tr>
<tr>
<td>0010</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>0020</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0030</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0040</td>
<td>DD</td>
<td>5D</td>
<td>DF</td>
<td>5B</td>
<td>5D</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>0050</td>
<td>DB</td>
<td>5F</td>
<td>DD</td>
<td>5D</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>0060</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0070</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>0080</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>0090</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>00A0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00B0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00C0</td>
<td>DD</td>
<td>5D</td>
<td>DF</td>
<td>5B</td>
<td>5D</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>00D0</td>
<td>DB</td>
<td>5F</td>
<td>DD</td>
<td>5D</td>
<td>DD</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>59</td>
<td>5D</td>
<td>5D</td>
<td>5D</td>
</tr>
<tr>
<td>00E0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>00F0</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
<td>FF</td>
</tr>
</tbody>
</table>
TITLE 'ZX-200A FLOPPY CONTROLLER V1.2'
MACLIB I8085

REVISION: 10/02/81 BY S.R.
VERSION 1.2
LAST MODIFIED AUG 6, 1981
RICK MAIN, PGMR

ZENDEX CORPORATION
6680 SIERRA LANE
DUBLIN, CA 94566
(415)829-1284
TWX 910-389-4009

THIS PROGRAM IS ORGANIZED TO RUN IN THE ZENDEX
ZX-200A FLOPPY DISK CONTROLLER. THE FIRMWARE IS
CONTAINED IN A SINGLE 2716 EPROM.

PORT EQUATES

0066 = RDSTAT EQU 66H; READ FDD I/F STATUS
0066 = WRCONT EQU 66H; WRITE FDD I/F CONTROL
0067 = RDRDY EQU 67H; READ INT & FDD READY STATUS
0067 = WRCNT1 EQU 67H; WRITE INT CLR & WR ENABLE, HEAD LOAD
0080 = CLRINT EQU 80H; CLEAR RST INTERRUPT FLIP-FLOP
2000 = DMADDR EQU 2000H; DMA BASE ADDRESS
2001 = DMATC EQU 2001H; DMA TERMINAL COUNT ADDRESS
2008 = DMAMOD EQU 2008H; DMA MODE SET REGISTER
6000 = MKRCRC EQU 6000H; READ/WRITE MARK/CRC BYTE
6100 = MARK EQU 6100H; READ/WRITE MARK
6200 = DISKIO EQU 6200H; IO DISK DATA
6300 = WRCRC EQU 6300H; WRITE CRC
6300 = SYNCPL EQU 6300H; READ SYNC PLO
6401 = PORT8B EQU 6401H; PORT 8B TO HOST
6402 = PORT79 EQU 6402H; PORT 79 TO HOST
6403 = PORT7B EQU 6403H; PORT 7B TO HOST
6400 = PORT89 EQU 6400H; PORT 89 TO HOST
6500 = WRCRC EQU 6500H; WRITE CLOCK
6500 = RDPRT1 EQU 6500H; READ 4X4 FILE PORT
07FF = ORG 7FFH; SET VERSION STAMP AT LAST ADDRESS
07FF 12 = DB 12H; FOR VERSION 1.2
0000 = ORG 0; BEGIN AT RESET POINT

0000 DB80 IN CLRINT; CLEAR INTERRUPTS TO HOST
0002 AF XRA A
0003 310042 LXI SP,4200H ; INITIALIZE STACK POINTER
0006 210040 LXI H,CHANWD ; CLEAR BUFFER, FLAGS

RAMCLR:
0009 77 MOV M,A
000A 2C INR L
000B C20900 JNZ RAMCLR ; LOOP TILL BUFFER CLEARED
000E DB67 IN RRDY ; READ INT. & FDD READY PORT
0010 E60F ANI 0FH
0012 321440 STA RDYBIT ; READY STATUS

DSKCLR:
0015 3EFF MVI A,OFFH
0017 320D40 STA UNIT
001A AF XRA A ; CLEAR HOST INTERRUPTS AND UNLOAD HEAD
001B D367 OUT WRCNT1
001D C35400 JMP L54

; TRAP INTERRUPT
INDEX:
; TRAP INTERRUPT SOURCE FROM FDD INDEX MARK

0024 24H
0024 F5 PUSH PSW
0025 C34A00 JMP L4A

MA78:
; MEMORY ADDRESS UPPER WRITTEN BY HOST FOR SD

002C 2CH
002C 2A0165 LHLDB 6501H ; READ IOPB ADDRESS FROM 4X4
002F DB80 IN CLRINT ; RESET RST F-F
0031 C3C000 JMP LCO

; RESTART 6.5

MA88:
; MEMORY ADDRESS UPPER WRITTEN BY HOST FOR DD

0034 2A0164 LHLDB PORT8B ; READ IOPB ADDRESS FROM 4X4
0037 DB80 IN CLRINT ; RESET RST F-F
0039 C39C00 JMP L9B

L3C:
STOP:
; STOP OPERATION INTERRUPT FOR SD DRIVES ON 88

003C 3CH
003C F5 PUSH PSW
003D 3A0040 LDA CHANWD ; RESET LINK BIT IN IOPB CHANNEL WORD
0040 E6FB ANI NOT 4
0042 320040 STA CHANWD
0045 3E10 MVI A,10H ; RESET RST 7.5 FF
0047+30 DB 30H
0048 F1 POP PSW
0049 C9 RET

L4A:
004A 3A1340 LDA INDXCT ; DECREMENT INDEX COUNT FLAG BY ONE FOR EACH
004D 3D DCR A ; REVOLUTION OF DISK, RST SOURCE INDEX MARK.
004E 321340 STA INDXCT

L54:
0051+20 DB 20H
0052 F1 POP PSW
0053 C9 RET
0054 3E08  MVI A,8 ; OPEN UP ALL RST X.5 INTERRUPTS
0056 FB    EI
0057+30  SIM 30H
0058 DB67  IN HRRDNY ; READ INT. & FDD READY PORT
005A 47   MOV B,A
005B E6CO  ANI OCOH ; ISOLATE INT PND BITS
005D C26A00 JNZ L6A ; JUMP IF INT PND
0060 78   MOV A,B
0061 E60F  ANI OFH ; ISOLATE DRIVE READY BITS
0063 211440 LXI H,RDYBIT ; READY STATUS
0066 BE   CMP M ; SEE IF OLD READY=NEW READY
0067 C47400 CNZ RDYINT ; JUMP TO DO UPDATE/INTERRUPT IF NOT

L6A:
006A 3A1340 LDA INDXCT ; INDEX COUNT
006D B7   ORA A ; REV COUNT EXPIRED?
006E C25400 JNZ L54 ; NO, LOOP
0071 C31500 JMP DSKCLR ; YES, DESELECT DISK UNIT

; INTERRUPT HOST ON READY CHANGE
; RDYINT:
0074 4F    MOV C,A ; FORMAT UNIT READY BITS
0075 0F    RRC
0076 0F    RRC
0077 47    MOV B,A
0078 79    MOV A,C
0079 07    RLC
007A 07    RLC
007B B0   ORA B
007C F60F  ORI OFH
007E 2F    CMA
007F 320164 STA PORT8B ; WRPORT 8B, SD RESULT BYTE
0082 320364 STA PORT7B ; WRPORT 7B, DD RESULT BYTE
0085 3E02  MVI A,2 ; RESULT TYPE = 2
0087 320064 STA PORT89 ; WRPORT 89 RESULT TYPE SD
008A 320264 STA PORT79 ; WRPORT 79 RESULT TYPE DD
008D 3A0D40 LDA UNIT ; TEST FOR INITIALIZATION CODE
0090 3C    INR A ; WAS IT FF?
0091 CA9600 JZ L95 ; RESET WR2D IF INIT
0094 3E08  MVI A,8 ; SET WR2D

L95:
0096 F6C0  ORI OCOH
0098 D367  OUT WRCNT1
009A 71    MOV M,C
009B C9    RET

L9B:
009C D1    POP D ; DD PROCEDURE
009D CD3901 CALL L13A ; DISCARD RST CALL RETURN
00A0 320364 STA PORT7B ; STORE IN 7B, RESULT BYTE
00A3 3E00  MVI A,0 ; STORE TYPE 0 RESULT
00A5 320264 STA PORT79 ; PORT 79H(HOST)
00A8 3E04  MVI A,04H ; TURN OFF WRITE ENABLE
00AA D367  OUT WRCNT1
00AC 3E08  MVI A,8 ; DELAY DESELECT 8 DISK REVS
00AE 321340 STA INDXCT ; INDEX COUNT
00B1 3A0040 LDA CHANWD ; ISOLATE INTERRUPT CONTROL
00B2 E610 ANI 10H
00B3 C25400 JNZ L54 ; RETURN TO MAINLINE IF DISABLED
00B4 3E44 MVI A,44H
00B5 D367 OUT WRCNT1 ; OTHERWISE SET INT 78
00B6 C35400 JMP L54
00B7 D1 POP D ; SD PROCEDURE
00B8 E610 ANI 10H
00B9 C25400 JNZ L54 RETURN TO MAINLINE IF DISABLED
00BB 3A0040 LDA CHANWD
00BC 47 MOV B,A
00BD E610 ANI 4 ; ISOLATE SUCCESSOR BIT
00BE D367 OUT WRCNT1 CLEAR WRITE ENABLE
00BF 3E04 MVI A,4
00C0 78 MOV A,B
00C1 F3 DI ; DISABLE ALL BUT INDEX TRAP
00C2 CD6101 CALL L162 ; DO SD DISK OPERATION
00C3 F5 PUSH PSW ; SAVE RESULT CODE ON STACK
00C4 320164 STA PORTBB ; WRITE RESULT BYTE TO HOST PORT 8BH
00C5 3A0040 LDA CHANWD
00C6 47 MOV B,A
00C7 D367 OUT WRCNT1 OTHERWISE SET INT 78
00C8 F1 POP D ; DISCARD RST CALL RETURN
00C9 F5 PUSH PSW
00CA 320164 STA PORT8B ; WRITE RESULT BYTE TO HOST PORT 8BH
00CB 3A0040 LDA CHANWD
00CC 47 MOV B,A
00CD E610 ANI 4 ; ISOLATE SUCCESSOR BIT
00CE 320164 STA PORTBB ; WRITE RESULT BYTE TO HOST PORT 8BH
00CF 320164 STA PORT8B ; WRITE RESULT BYTE TO HOST PORT 8BH
00D0 3A0040 LDA CHANWD
00D1 F3 DI ; DISABLE ALL BUT INDEX TRAP
00D2 CD6101 CALL L162 ; DO SD DISK OPERATION
00D3 F5 PUSH PSW ; SAVE RESULT CODE ON STACK
00D4 320164 STA PORTBB ; WRITE RESULT BYTE TO HOST PORT 8BH
00D5 3A0040 LDA CHANWD
00D6 47 MOV B,A
00D7 3E04 MVI A,4 ; ISOLATE SUCCESSOR BIT
00D8 E610 ANI 4 ; ISOLATE SUCCESSOR BIT
00D9 3E04 MVI A,4 ; ISOLATE SUCCESSOR BIT
00DA 2F CMA
00DB 320064 STA PORT89 ; WRITE FOR HOST READ PORT 89H
00DC F1 POP PSW ; POP ERROR CODE TO ACC
00DD B7 ORA A ; ERRORS=NOT 0
00DE C43000 CNZ STOP ; CALL LINK CANCEL IF ERROR
00DF 3E0D MVI A,0DH ; MASK ALL BUT RST 6.5(SD CMD)
00E0 FB EI
00E1 2A0840 LHLD 400BH ; SET WAIT BIT IN HOST IOPB CHANWD
00E2 321340 STA INDXCT ; INDEX COUNT
00E3 CB5A00 LDA CHANWD SAVE CHANWD ON STACK
00E4 F5 PUSH PSW
00E5 07 RLC TEST LOCK OVERRIDE BIT
00E6+30 30H DB 30H ; FINAL RESULT TYPE
00E7 3E04 MVI A,4 ; FINAL RESULT TYPE
00E8 D367 OUT WRCNT1 ; CLEAR WRITE ENABLE
00E9 3E08 MVI A,8 ; CLEAR WRITE ENABLE
00EB 321340 STA INDXCT ; INDEX COUNT
00ED 400BH ; SET WAIT BIT IN HOST IOPB CHANWD
00EE FB EI
00F0 78 MOV A,B
00F1 F5 PUSH PSW
00F2 07 RLC TEST LOCK OVERRIDE BIT
00F3 DA0701 JC L108 ; JUMP IF OVERRIDE SET
00F4 2A0840 LHLD 400BH ; SET WAIT BIT IN HOST IOPB CHANWD
00F5 321340 STA INDXCT ; INDEX COUNT
00F6 2643 PUSH H
00F7 0F MVI H,43H ; INDEX 2114 BUFFER
00F8 0F RRC
00F9 F601 ORI 1 ; SET WAIT BIT
00FA 77 MOV M,A ; PUT BACK IN 2114 BUFFER
00FB E1 POP H
00FC 010040 LXI B,CHANWD
00FD CDF901 CALL DMALOD ; USE DMAC TO WRITE HOST
00FE F1 POP PSW ; POP CHANWD
00FF 47 MOV B,A
00E0 E620 ANI 20H ; ISOLATE INTERRUPT CONTROL
00E1 C21D01 JNZ L11E ; JUMP IF INTERRUPT NEEDED BY HOST
00E2 2A0840 LHLD NXIOPB ; NEXT IOPB ADDRESS
00E3 78 MOV A,B
L11E:
011D 3E84 MVI A,84H ; SET INTERRUPT 88
011F D367 OUT WRCNT1

L122:
0121 3A0040 LDA CHANWD ; ISOLATE SUCCESSOR BIT
0124 B604 ANI 4
0126 CA5400 JZ L54 ; RETURN TO IDLE LOOP IF NOT LINKED
0129 3E0A MVI A,0AH ; SET RST 6.5 MASK
012B+30 DB 30H
012D DB67 IN RDRDY ; READ INT. & FDD READY PORT
012E B680 ANI 80H ; ISOLATE INT 88 BIT
0130 C22101 JNZ L122 ; LOOP TILL FALSE, HOST MUST ACK
0133 2A0840 LHLDA NXIOPB ; NEXT IOPB ADDRESS
0136 C3C100 JMP LC1

L13A:
0139 CDE301 CALL L1D7 ; FETCH IOPB FROM HOST TO BUFFER
013C 3A0140 LDA DKINST ; LOAD UNIT NO. SELECTED BY HOST
013F OF RRC
0140 OF RRC
0141 OF RRC
0142 OF RRC
0143 B603 ANI 3 ; ISOLATE UNIT NUMBER
0145 4F MOV C,A ; SAVE IN C
0146 1E04 MVI E,4 ; SEND DD MODE TO WRCNT1 PORT
0148 3E0B MVI A,OBH ; SPECIFY DD DATA ADDRESS MARK
014A 321640 STA DAMARK ; DATA ADDRESS MARK STORE
014D 3E0F MVI A,0FFH ; SET DD FLAG MODE
014F C38401 JMP L185 ; GO AROUND L153

L153:
0152 2A0840 LHLDA NXIOPB ; NEXT IOPB ADDRESS
0155 0F RRC ; TEST BRANCH BIT
0156 DA6101 JC L162 ; IF SET, BRANCH TO LINKED IOPB
0159 060A MVI B,0AH ; OTHERWISE IDLE 10 MS
015B CDBF02 CALL DELAY
015E 2A0B40 LHLDA 400BH ; RESTORE ORIGINAL IOPB ADDRESS

L162:
0161 220B40 SHLD 400BH ; SAVE AS ADDRESS OF IOPB TO EXECUTE
0164 CDE301 CALL L1D7 ; MOVE HOST IOPB TO OUR BUFFER
0167 3A0040 LDA CHANWD ; FETCH CHANNEL WORD
016A 0F RRC
016B DA5201 JC L153 ; JUMP IF WAIT BIT SET
016E 3A0140 LDA DKINST ; FETCH OPCODE
0171 OF RRC ; POSITION & ISOLATE UNIT SELECT BITS
0172 OF RRC
0173 OF RRC
0174 B606 ANI 6 ; INTERMEDIATE TO C
0176 4F MOV C,A ; COMBINE
0177 OF RRC
0178 A9 XRA C
0179 B603 ANI 3 ; MASK 00=0, 11=1, 01&10 ILLEGAL
MOV C, A ; SEND SD MODE CONTROL TO WRCONT1 PORT
MVI E, 05H ; SPECIFY SD DATA ADDRESS MARK, FOR USE LATER
MVI A, 0FBH ; DATA ADDRESS MARK
STA DAMARK ; DATA ADDRESS MARK
XRA A ; ACC=0, FLAG SD OPERATION

STA DDSDFL ; STORE SD/DD FLAG PASSED IN ACC.
LXI H, UNITAB ; DO INDEX TO UNIT SELECT TABLE
DAD B ; MSD=0
DAD B ; DOUBLE INDEX, TWO ENTRIES PER ITEM
MOV D, M ; MOVE SELECT CODE TO D
IN RRDY ; READ INT. & FDD READY PORT
ANA D ; TEST FOR READY ERROR
MVI A, 80H ; ANTICIPATE BY WRITING ERROR CODE IN ACC.
RNZ ; RETURN IF READY ERROR

; OTHERWISE FETCH NEXT ITEM IN UNIT TABLE
MOV A, C
OUT WRCONT ; WRITE FDD CONTROL PORT TO SELECT THAT DRIVE
STA 400AH ; ALSO SAVE SELECT CODE FOR LATER USE
LDA UNIT ; FETCH LAST UNIT CODE
CMP C ; SAME?

MOV A, C ; SAVE NEW CODE
OUT WRCONT1 ; SET DD OR SD MODE AND HEAD LOAD
LXI H, 401AH
DAD B ; CALL DELAY FOR HEAD TO SETTLE IF NEW UNIT

LXI H, CTAB ; BASE INDEX TO COMMAND TABLE
LDA DKINST ; FETCH OPCODE
RCL ; INDEX BY TWO'S
ANI 0EH ; ISOLATE OPCODE BITS
ORA A ; ADD TO CTAB BASE ADDRESS

XCHG ; JUMP TO COMMAND PROCEDURE

CTAB:

DW L20C ; NOP
DW SEEK ; SEEK
DW FORMAT ; FORMAT
DW RECAL ; RECALIBRATE
DW READ ; READ
DW VERIFY CRC
DW WRITE ; WRITE
UNITAB:

01DB 01 DB 1 ; UNIT 1
01DC BF DB NOT 40H
01DD 02 DB 2 ; UNIT 2
01DE DF DB NOT 20H
01DF 04 DB 4 ; UNIT 3
01E0 EF DB NOT 10H
01E1 08 DB 8 ; UNIT 4
01E2 F7 DB NOT 8

L1D7:

01E3 010980 LXI B,8009H ; SELECT HOST READ, 9 BYTES
01E6 CDF901 CALL DMALOD ; INVOKE DMA TRANSFER
01E9 2643 MVI H,43H ; ADDRESS ON-BOARD BUFFER
01EB 110040 LXI D,CHANWD ; POINT TO IOPB CHANNEL WORD
01EE 080A MVI C,10 ; MOVE 10 BYTES FROM DMA BUFFER TO IOPB ARRAY

L1E6:

01F0 7E MOV A,M ; READ BUFFER
01F1 12 STAX D ; STORE AT IOPB
01F2 13 INX D ; ADJUST POINTERS
01F3 2C INR L
01F4 0D DCR C ; DECREMENT COUNTER
01F5 C2F001 JNZ L1E6
01F8 C9 RET

DMALOD:

01F9 97 SUB A
01FA D365 OUT 65H ; CLR EXTENDED ADDRESS
01FC E5 PUSH H ; SAVE HL
01FD EB XCHG
01FE 210020 LXI H,DMADDR ; SEND TO 8257 DMA REG PORT
0201 73 MOV M,E
0202 72 MOV M,D
0203 23 INX H ; POINT TO TC PORT
0204 71 MOV M,C ; WRITE TC AND CYCLE CODE
0205 70 MOV M,B
0206 3E41 MVI A,41H ; WRITE MODE WORD, TC STOP BIT & CHAN. 0 ENABLE
0208 320820 STA DMAMOD
020B E1 POP H ; RESTORE DMA ADDRESS
020C C9 RET

L20C:

020D CD7C02 CALL SEEK ; DO SEEK
0210 B7 ORA A ; TEST ERROR
0211 C0 RNZ ; RETURN IF ERROR
0212 3A0240 LDA NUMRCD ; LOAD # RECORDS
0215 E607 ANI 7
0217 07 RLC
0218 4F MOV C,A
0219 0600 MVI E,0
021B 212402 LXI H,L223
021E 09 DAD B
021F 5E MOV E,M
0220 23 INX H
0221 56 MOV D,M
0222 EB XCHG
0223 E9 PCHL ; EXECUTE PROCEDURE
L223:  DW L233
  DW L23A
  DW L240
  DW L246
  DW L256
  DW L25B
  DW L262
  DW L267
L233:  LXI D,DISKIO ; RD DISK
  JMP L236 ; LOOP
L23A:  LXI D,6100H ; RDMRKA
  JMP L236 ; GO INTO LOOP
L240:  LXI D,DISKIO ; RD DISK
  JMP L249
L246:  LXI D,MARK ; RDMRKA
  PUSH D
  CALL WRENBL ; ENABLE WRITE CONTROLS
  POP D
  RZ ; RETURN IF WRITE PROTECTED
  LDA SCTADR ; LOAD SECTOR ADDRESS
L252:  STAX D ; WMK(CRC)
  JMP L252
L256:  MVI C,0
  JMP L25D
L25B:  MVI C,OFFH ; DMA 256 BYTES
L25D:  MVI B,80H ; READ FROM HOST RAM
  JMP L26B
L262:  MVI C,0 ; NO DATA TRANSFER
  JMP L269
L267:  MVI C,OFFH ; 256 BYTES
L269:  MVI B,40H ; WRITE TO HOST RAM
L26B:  PUSH B
  MVI C,OFFH ; 256 BYTES
  LHLB BUFFER ; LOAD BUFFER ADDRESS
  CALL DMALOD ; MOVE DATA
  POP B ; RESTORE BC
  XRA A ; ACC=0
  ORA C
  JNZ L26B ; JUMP IF NOT A VERIFY CYCLE
  RET
SEEK A TRACK - THE BYTE IN TRKADR IS USED TO DETERMINE THE TARGET TRACK TO STEP TO. IF ZERO RECALIBRATE IS USED INSTEAD, ENTERED DIRECTLY AS A HOST COMMAND OR AS A PRELUDE TO ANOTHER OPERATION FOR IMPLIED SEEK.

SEEK:

027C 3A0340 LDA TRKADR ; SEEK OPERATION. FETCH TARGET TRACK ADDRESS
027F B7 ORA A ; ZERO?
0280 CAD502 JZ RECAL ; DO RECAL IF ZERO
0283 57 MOV D,A ; TRK ADR TO D
0284 F84D CPI 4DH ; TEST FOR ILLEGAL TRACK
0286 3E08 MVI A,8 ; ANTICIPATE WITH ERROR CODE
0288 D0 RNC ; RETURN IF ERROR
0289 7A MOV A,D ; TRK ADR TO ACC.
028A FE2B CPI 2BH ; TRK>43?
028C DA9602 JC S1 ; JUMP IF NOT
028F 210A40 LXI H,400AH
0292 7E MOV A,M
0293 E6FB ANI 0FBH ; SET LO CURR BIT
0295 77 MOV M,A

S1:

0296 CDCA02 CALL TRKREG ; SET HL = TRK REG

S2:

0299 7A MOV A,D ; TRK ADR TO ACC.
029A 96 SUB M ; COMPARE TARGET TRACK TO DESTINATION TRACK
029B C8 RZ ; RETURN WHEN EQUAL
029C CD2A02 CALL STEP ; DO A STEP OF HEAD
029F C39902 JMP S2 ; LOOP

STEP:

02A2 3A0A40 LDA 400AH ; LOAD UNIT SELECT
02A5 DAAC02 JC L2AB ; JUMP TO DO STEP OUT
02A8 E6FD ANI OFDH ; ELSE STEP IN
02AA 34 INR M ; INCREMENT CURRENT TRACK
02AB 34 INR M

L2AB:

02AC 35 DCR M ; DECREMENT CURRENT TRACK
02AD D366 OUT WRCNT ; WRITE FDD CONTROL PORT ; STEP=1
02AF 3D DCR A
02B0 00 NOP
02B1 D366 OUT WRCNT ; WRITE FDD CONTROL PORT, STEP=0
02B3 00 NOP
02B4 F601 ORI 1
02B6 D366 OUT WRCNT ; WRITE FDD CNT PORT, STEP=1
02BB D366 OUT WRCNT ; WRITE FDD CONTROL PORT, DIR=1

TENMS:

02BD 0608 MVI B,8 ; 10 MS TIMING FOR STEP RATE

DELAY:

02BF 0ED6 MVI C,0D6H ; DELAY LOOP

L2B9:

02C1 0D DCR C
02C2 C2C102 JNZ L2B9
02C5 05 DCR B
02C6 C2BF02 JNZ DELAY
02C9 C9 RET
TRKREG:

02CA 210E40
02CD 3A0D40
02D0 4F
02D1 0600
02D3 09
02D4 C9

LXI H, TRKRGO ; TRACK REGISTER, UNIT 0, INDEX BASE ADDR.
LDA UNIT ; FETCH REQUESTED UNIT NUMBER
MOV C, A ; ADD TO INDEX
MVI B, 0
DAD B ; INDEX # ADDED TO BASE= TARGET TRK CURRENT REG
RET

RECAL:

CALL TRKREG ; RECALIBRATE COMMAND
L2DO:

MVI M, 0 ; ZERO TRK REG
IN RDSTAT ; READ FDD STATUS PORT
ANI 40H ; ISOLATE TRK 0 SIGNAL FROM FDD
RZ ; RETURN WHEN TRK=0
STC ; SET FLAG FOR STEP
CALL STEP ; DO A STEP
JMP L2DO

WRENBL:

LDA DDSDFL
CMA
ANI 1 ; ISOLATE SD/DD BIT
ORI 6 ; SET WRENA & HEAD LOAD BITS
OUT WRCNT1 ; WRITE CONTROL PORT
IN RDSTAT ; READ FDD STATUS PORT
ANI 20H ; ISOLATE WRITE PROTECT LINE
MVI A, 20H
RET ; RETURN WITH FLAGS SET

READ:

XRA A ; READ, VERIFY CRC COMMAND
JMP L300

WritDL:

LXI H, 4016H ; FIX FLAG, WRITE DELETED COMMAND
MOV A, M
ANI NOT 7
WRITE:

0302 CDB602 CALL WRENBL ; WRITE COMMAND, CALL WR ENABLE
0305 C8 RZ ; RETURN ERROR IF WRITE PROTECTED
0306 3EFF MVI A, OFFH ; SET FLAG

L300:
0308 321540 STA RWFLG ; READ WRITE FLAG ; STORE FLAG
030B CD7C02 CALL SEEK ; FIX CNTL, DO SEEK
030E B7 ORA A
030F C0 RNZ ; RETURN IF SEEK ERROR
0310 210440 LXI H, SCTADR ; POINT TO SECTOR ADDRESS
0313 0636 MVI B, 36H ; DD EOT SECTOR +1
0315 3A1240 LDA DDSDFL ; TEST FOR DD
0318 B7 ORA A
0319 7E MOV A, M ; FETCH SECTOR ADDRESS
031A C22203 JNZ L31A ; JUMP IF D
031D E61F ANI 1FH ; ISOLATE LEGAL SD SECTOR ADDRESSES
031F 77 MOV M, A ; RESTORE CORRECTED ADDR
0320 061C MVI B, 1CH ; MAX SD ADDR +1

L31A:
0322 3C INR A ; STORE NEXT SECTOR ADDRESS
0323 321940 STA FLAGD
0326 3D DCR A ; TEST FOR ZERO
0327 3B08 MVI A, 8 ; ANTICIPATE ADDRESS ERROR
0329 C8 RZ ; RETURN IF SECTOR ADDR=0, ILLEGAL
032A 110240 LXI D, NUMRCD ; FETCH # RECORDS
032C 1A LDAX D ; ADD ADDRESS TO # OF RECORDS
032F B8 CMP B ; TOO MUCH?
0330 3B08 MVI A, 8 ; ANTICIPATE ERROR
0332 D0 RNC ; RETURN ERROR IF TOO MUCH
0333 1A LDAX D ; LOAD AGAIN # RECORDS
0334 B7 ORA A
0335 1F RAR
0336 321840 STA FLAGC
0339 CE00 ACI 0
033B 12 STAX D

L334:
033C AF XRA A
033D 321740 STA FLAGB
0340 2A0540 LHLD BUFFER ; LOAD BUFFER ADDR
0343 017F80 LXI B, 807FH ; DMA CYCLE, TC
0346 3A1540 LDA RWFLG ; READ WRITE FLAG, FLAG =0 FOR READ, NZ FOR WRITE
0349 B7 ORA A
034A C4F901 CNZ DMALOD ; DMA LOAD UP
034D 3E04 MVI A, 4 ; COUNT FOUR DISK REVS
034F 321340 STA INDXCT ; INDEX COUNT

L34A:
0352 210065 LXI H, 6500H
0355 110062 LXI D, DISKIO
0358 3A1240 LDA DDSDFL
035B B7 ORA A
035C CAA704 JZ L49B

L357:
035F 3600 MVI M, 0 ; WRCLK
0361 DB66 IN RDSTAT ; READ FDD STATUS PORT
L35B:  
LDA INDEXCT ; INDEX COUNT
ORA A
JM L45F

L362:  
INR B
JZ L35B
LDA SYNCPCL ; SYNCHPLO
INR A
JNZ L362
MVI C,06

L36F:  
LDA D ; DISKRD
INR A
JNZ L362
DCR C
JNZ L36F

L383:  
LDAX D ; READ TRACK I.D.
LXI H,TRKADR ; COMPARE TRK
CMP M
JNZ L467
LDAX D ; DISKRD, DISCARD SIDE I.D.
INX H ; COMPARE SECTOR ADDR
LDAX D ; DISK READ, SECTOR I.D.
CMP M
JNZ L34A
LDAX D ; READ SECTOR LENGTH CODE
LDAX D ; READ CRC BYTES
LDAX D

L393:  
IN RDSTAT ; READ FDD STATUS PORT
RAL ; CRCSTAT?
JC L481 ; JUMP CRC ERROR
LDAX D ; DISKRD
INR M ; INX SECTOR ADR
INR M
LDAX D ; DISKRD
LDA DDSDFL ; TEST SD/DD MODE
ORA A
JZ L4D3 ; JUMP IF SD
LDAX D ; DISKRD
LDA RWFLG ; READ WRITE FLAG TEST
ORA A
JNZ L3FB ; JUMP IF WRITE
MVI B,12H

L3B1:  
LDAX D ; DISKRD
CP/M MACRO ASSEM 2.0  #013  ZX-200A FLOPPY CONTROLLER V1.2

```
03BA 05  DCR  B
03BB C2B903  JNZ  L3B1
03BE 1A  LDAX  D
03BF 210065  LXI  H,6500H
03C2 3600  MVI  M,0 ; MAKE SYNCHMARK=0

03C4 00  NOP

L3BC:
03C5 3A00663
03C8 3C
03CA C2C403
03CC 3670
03CE 1A
03CF 3A0061
03D2 FE0B
03D4 C29004

L3BC:
03D7 2A0540  LHLB  BUFFER ; LOAD BUFFER ADDRESS
03D9 2643  MVI  H,43H

L3BD:
03DC 1A
03DD 77
03DE 0E7F

L3C:
03E0 2C  INR  L ; ADJUST 2114 POINTER
03E1 1A  LDAX  D ; READ DISK BYTE
03E2 77  MOV  M,A ; STORE IN 2114
03E3 0D  DCR  C
03E4 C2E003  JNZ  L3D9
03E5 1A  LDAX  D ; READ CRC BYTES
03E6 1A  LDAX  D
03E9 1A  LDAX  D ; GAP 3 READ
03ED DB66  IN  RDSTAT ; READ FDD STATUS PORT
03EC 17  RAL ; CRCSTAT?
03ED 3E02  MVI  A,2 ; ANTICIPATE CRC ERROR
03EF D8  RC ; RETURN IF ERROR
03F0 2A0540  LHLB  BUFFER ; LOAD BUFFER ADDR
03F3 017F40  LXI  B,407FH ; READ CYCLE, TC, FOR READ COMMAND TO DMAC
03F6 3A0140  LDA  DKINST ; FETCH OPCODE
03F9 0601  ANI  1 ; SEE IF DATA TRANSFER TYPE
03FB CCF901  CZ  DMALOD ; LOAD DMAC CHIP
03FE C33B04  JMP  L432

L3FB:
0401 1A  LDAX  D ; READ GAP 2 BYTE
0402 0E09  MVI  C,9

L3FE:
0404 1A  LDAX  D ; READ GAP 2
0405 0D  DCR  C ; DECREMENT LOOP
0406 C20404  JNZ  L3FE ; LOOP TEN TIMES
0409 12  STAX  D ; SPLICE IN SAME BYTE READ
040A 12
```
L409:

0414 010061 LXI B,MARK ; POINT TO WRMK

; WRITE DISK FROM 2114 BUFFER

; L411:

L421:

043A 12 STAX D ; SPLICE GAP 3

043B 210640 LXI H,BUFFER+1 ; BUFFER MSD

043E 34 INR M ; ADVANCE BUFFER ADDRESS BY ONE BLOCK

043F 210240 LXI H,NUMRCD ; DCR # RECORDS

0442 35 DCR M

0443 C23604 JNZ L334

0446 111840 LXI D,FLAGC

0449 1A LDAX D

044A B7 ORA A

044B C8 RZ

044C 77 MOV M,A

044D AF XRA A ; A=0

044E 12 STAX D

044F 3A1940 LDA FLAGD

0452 210440 LXI H,SCTADR

0455 56 MOV D,M ; FETCH SECTOR ADDR

0456 77 MOV M,A ; REPLACE IT

0457 92 SUB D

0458 1F RAR

0459 57 MOV D,A

045A 1280 MVI E,128

045C 2A0540 LHLX BUFFER ; INX BUFFER ADDR ONE SECTOR COUNT
L45F:
0466 3A1740   LDA   FLAGB
0469 B7       ORA   A
046A C0       RNZ
046B 3E0E     MVI   A,0EH
046D C9       RET

L467:
046E 1A       LDAX  D
046F 1A       LDAX  D
0470 1A       LDAX  D
0471 1A       LDAX  D
0472 1A       LDAX  D
0473 1A       LDAX  D     ; SIXTH TIME

L46D:
0474 DB66     IN     RDSTAT  ; RCSTAT?
0476 17       RAL
0477 DA8804   JC     L481   ; JUMP CRC ERROR
047A 3E04     MVI   A,04
047C 321740   STA   FLAGB
047F CDD502   CALL  RECAL
0482 CD7C02   CALL  SEEK
0485 C35203   JMP   L34A

L481:
0488 3E0A     MVI   A,0AH
048A 321740   STA   FLAGB
048D C35203   JMP   L34A

L489:
0490 FE08     CPI   08H
0492 C39704   JMP   L4A2

L48E:
0495 FEF8     CPI   OF8H

L4A2:
0497 3E0F     MVI   A,0FH

L492:
0499 321740   STA   FLAGB
049C 3E01     MVI   A,1
049E C8       RZ
049F 210440   LXI   H,4004H
04A2 35       DCR   M
04A3 35       DCR   M
04A4 C35203   JMP   L34A

L49B:
04A7 36FF     MVI   M,0FFH

L49D:
04A9 3A1340   LDA   INDXCT  ; INDEX COUNT
04AC B7       ORA   A
04AD FA6604   JM    L45F

L4A4:
04B0 04       INR   B     ; INX LOOP COUNT
04B1 CAA904   JZ    L49D
04B4 3A0063   LDA   SYNCPL  ; SYNCPL
04B7 B7       ORA   A
04B8 C2B004   JNZ   L4A4   ; LOOP TILL 00 READ
L4AF:
04BB 1A LDA X ; READ GAP TILL 00 READ
04BC B7 ORA A
04BD C2B004 JNZ L4A4
04CO 1A LDA X
04C1 B7 ORA A
04C2 C2B004 JNZ L4A4

L4B9:
04C5 36C7 MVI M,OC7H ; WRITE C7 SYNCH MARK

L4BB:
04C7 3A0060 LDA MRKCR ; READ DATA MARK
04CA B7 ORA A
04CB CAC704 JZ L4BB ; LOOP TILL NON-ZERO
04CE FEFE CPI OFEH ; SEE IF CORRECT DATA MARK
04D0 CA8B03 JZ L383 ; JUMP IF GOOD
04D3 36FF MVI M,OFFH
04D5 3A0063 LDA SYNCPL
04D8 B7 ORA A
04D9 CAC504 JZ L4B9
04DC C3A704 JMP L49B

L4D3:
04DF 3A1540 LDA RWFLG ; READ WRITE FLAG
04E2 B7 ORA A
04E3 C20405 JNZ L4F8
04E6 1A LDA X
04E7 060A MVI B,0AH ; WAS 7, READ IN THREE EXTRA TO IGNORE
 ; THANKS TO LARRY BOBERG.

L4DD:
04E9 1A LDA X
04EA 05 DCR B
04EB C2E904 JNZ L4DD
04EE 1A LDA X
04EF 210065 LXI H,6500H
04F2 36FF MVI M,OFFH ; WRCLK
04F4 3A0063 LDA SYNCPL
04F7 36C7 MVI M,OC7H
04F9 3A0060 LDA MRKCR
04FC FEFB CPI OFBH
04FE C29504 JNZ L48E
0501 C3D703 JMP L3CF

L4F8:
0504 1A LDA X
0505 1A LDA X
0506 3EFF MVI A,OFFH
0508 12 STAX D
0509 12 STAX D
050A 12 STAX D
050B AF XRA A
050C 12 STAX D
050D 12 STAX D
050E 12 STAX D
050F 12 STAX D
0510 010060 LXI B,MRKCR
0513 C31704 JMP L411
FORMAT A TRACK - THIS COMMAND WILL SEEK THE TRACK, READ IN USER BUFFER, AND EITHER WRITE SEQUENTIAL OR RANDOM NUMBERED SECTORS FILLED WITH USER DATA FILLER BYTE. FOR SEQUENTIAL SECTORS ONLY THE FIRST BYTE OF USER BUFFER SPECIFIES WHAT ALL SECTORS ON THE TRACK WILL HAVE THEIR DATA FIELDS FILLED WITH. FORMAT WRITES ALL MARKS, GAPS, AND FIELDS ACCORDING TO FM FORMAT OR MMFM FORMAT. MMFM/FM TYPE IS SIGNalled BY FLAG "DDSDFL".

FORMAT:

0516 CD7C02 CALL SEEK ; FORMAT COMMAND, DO SEEK
0519 B7 ORA A ; RETURN IF ERRORS
051A C0 RNZ ; RETURN IF ERROR
051B CDE602 CALL WRENBL ; CALL WRITE ENABLE
051E C8 RZ
051F 016880 LXI B,8068H ; DMA CYCLE, TC BYTES
0522 2A0540 LHLD BUFFER ; READ FORMAT TABLE FROM HOST AT BUFFER
0525 CDF901 CALL DMALOD ARM DMA
0528 2643 MVI H,43H ; POINT TO PAGE IN 2114
052A 3A0040 LDA CHANWD FETCH CHANNEL WORD
052D B640 ANI 40H ; RANDOM FORMAT?
052F C24105 JNZ L535 ; JUMP IF RANDOM
0532 E5 PUSH H ; ELSE DO SEQUENTIAL SECTORS
0533 7E MOV A,M ; FILL ALL SECTORS WITH THIS DATA
0534 010134 LXI B,3401H LOOP COUNT =34H, SECTOR # 1

L52B:

0537 71 MOV M,C ; WRITE SECTOR NUMBER
0538 2C INR L
0539 77 MOV M,A ; WRITE DATA BYTE
053A 2C INR L
053B 0C INR C
053C 05 DCR B
053D C23705 JNZ L52B
0540 E1 POP H

; RANDOM FORMAT. AT THIS POINT THE TABLE OF SECTOR ORDER AND FILLER BYTE IS COMPLETE.

L535:

0541 011340 LXI B,INDXCT ; INDEX COUNT
0544 110062 LXI D,DISKIO ; POINTER FOR STAX D TO WRITE DISK
0547 3E01 MVI A,1 ; INDEX 1 DISK REV.
0549 02 STAX B ; STORE IN FLAG
054A 3A1240 LDA DDSDFL ; TEST DD OR SD FORMAT
054D B7 ORA A
054E CABB05 JZ L5B1 ; JUMP IF SD
0551 0A L545: LDAX B ; FETCH INDEX COUNTER
0552 B7 ORA A
0553 C25105 JNZ L545 ; LOOP TILL INDEX MARK
0556 AF XRA A
0557 0E3F MVI C,3FH

L54D:

DDGAP1:

0559 12 STAX D ; WRITE GAP 1, 64 ZERO'S
055A OD DCR C
055B C25905 JNZ L54D
055E 12 STAX D ; ONE MORE
055F 0634 MVI B,34H ; DD SECTOR COUNT

L555:
0561 DB66 IN RDSTAT
0563 3EFF MVI A,OFFH
0565 0EO9 MVI C,9

L55B:
0567 12 STAX D ; WRITE OFFH TEN TIMES
0568 0D DCR C
0569 C6705 JNZ L55B ; LOOP
056C 12 STAX D ; ONE MORE
056D 3E8E MVI A,0EH ; WRITE I.D. ADDRESS MARK FOR MMFM
056F 32061 STA MARK ; WRMK
0572 3A0340 LDA TRKADR ; LOAD TRACK ADDR
0575 12 STAX D ; WRDISK
0576 AF XRA A ; ACC=0
0577 12 STAX D ; WRDISK
0578 7E MOV A,M ; FETCH SECTOR I.D.
0579 12 STAX D ; WRDISK
057A AF XRA A ; ACC=0
057B 2C INR L
057C 12 STAX D ; WRDISK
057D 320063 STA WRCRC ; WRCRC
0580 320063 STA WRCRC ; WRCRC
0583 0E11 MVI C,11H

DDGAP2:
L57A:
0585 12 STAX D ; WRITE 11 ZERO'S
0586 0D DCR C
0587 C2505 JNZ L57A
058A 12 STAX D ; ONE MORE
058B 3EFF MVI A,OFFH ; WRITE NINE OFFH's
058D 0E09 MVI C,9

L584:
058F 12 STAX D ; WRDISK
0590 0D DCR C
0591 C2805 JNZ L584
0594 12 STAX D ; ONE MORE
0595 3E8B MVI A,0BH ; WRITE DATA MARK
0597 320061 STA MARK ; SEND TO DISK
0599 7E MOV A,M ; LOAD DATA BYTE FOR FILL
059B 0E7F MVI C,7FH ; ONE SECTOR COUNT-1

L593:
059D 12 STAX D ; FILL DATA FIELD
059E 0D DCR C
059F C29D05 JNZ L593 ; ONE MORE
05A2 12 STAX D
05A3 AF XRA A
05A4 320063 STA WRCRC ; WRITE TWO DATA FIELD CRC BYTES
05A7 2C INR L
05A8 320063 STA WRCRC ; WRCRC
05AB 0E11 MVI C,11H

DDGAP3:
L5A3:
05AD 12 STAX D
05AE 0D DCR C
05AF C2AD05  JNZ  L5A3  ; WRITE 17 ZERO'S
05B2 12     STAX  D  ; ONE MORE
05B3 05     DCR  B
05B4 C26105  JNZ  L555  ; DO NEXT RECORD
35B7 12     STAX  D  ; WRITE A ZERO
05B8 C31E06  JMP  L616  ; DO GAP 4 AND RETURN
           ; WAIT FOR INDEX MARK
           ; L5B1:
05BB 0A     LDAX  B  ; TEST INDEX FLAG
05BC B7     ORA  A
05BD C2BB05  JNZ  L5B1  ; LOOP TILL INDEX
05C0 3EFF   MVI  A,OFFH
05C2 0E48   MVI  C,48H
           ; SINGLE DENSITY FORMAT RECORD
           ; L5BA:
05C4 12     STAX  D  ; WRITE 48H OFFH'S
05C5 0D     DCR  C
05C6 C2C405  JNZ  L5BA
05C9 12     STAX  D  ; ONE MORE
05CA 061A   MVI  B,26  ; 26 SECTORS LOOP COUNT
           ; L5C2:
05CC AF     XRA  A
05CD 12     STAX  D  ; WRITE 6 ZERO'S
05CE 12     STAX  D
05CF 12     STAX  D
05D0 12     STAX  D
05D1 12     STAX  D
05D2 12     STAX  D
05D3 3EFE   MVI  A,0FEH  ; WRITE I.D. RECORD MARK
05D5 320060  STA  MRKCRC
05D8 3A0340  LDA  TRKADR  ; FETCH TRACK ADDRESS
05DB 12     STAX  D  ; WRITE TRACK I.D.
05DC AF     XRA  A
05DD 12     STAX  D  ; WRITE SIDE 0 I.D.
05DE 7E     MOV  A,M  ; FETCH SECTOR ADDRESS
05DF 12     STAX  D  ; WRITE SECTOR I.D.
05E0 AF     XRA  A
05E1 2C     INR  L
05E2 12     STAX  D  ; WRITE SECTOR LENGTH CODE=0
05E3 3EFF   MVI  A,OFFH
05E5 320063  STA  WRCRC  ; WRITE TWO CRC BYTES
05E8 320063  STA  WRCRC  ; WRCRC
        ; WRITE SD GAP 2
        ; 05EB 0E0A  MVI  C,0AH
        L5E4:
05ED 12     STAX  D  ; WRITE FF
05EE 0D     DCR  C
05EF C2ED05  JNZ  L5E4
05F2 12     STAX  D  ; ONE MORE FF
XRA A  ; WRITE ZERO'S
STAX D
STAX D
STAX D
STAX D
STAX D
STAX D
STAX D

; WRITE DATA FIELD

MVI A,0FBH  ; WRITE DATA ADDRESS MARK
STA MRKCRC  ; WRMRKCRC
MOV A,M  ; FETCH FILLER DATA
MVI C,7FH

L5FA:
STAX D  ; FILL ANOTHER 127 BYTES WITH SAME DATA

DCR C
JNZ L5FA

MVI A,0FH  ; WRITE TWO CRC BYTES
STA WRCRC
INR L
STA WRCRC
MVI C,1AH  ; WRITE GAP 3

L60B:
STAX D  ; FILL WITH FF
DCR C
JNZ L60B

STAX D  ; ONE MORE FF
DCR B
JNZ L61B

GAP4:
L61B:
XCHG

LXI D,INDXCT  ; INDEX COUNT
MOV B,A

MOV M,B  ; WRITE GAP 4 TILL INDEX MARK
LDAX D  ; FETCH INDEX COUNT
ORA A
JP L61B
XRA A
RET  ; RETURN TO MAINLINE

;******************************************************************************

; RAM BUFFER

4000
ORG 4000H

IOPB:
4000
CHANWD: DS 1  ; IOPB CHANNEL WORD
4001
DKINST: DS 1  ; DISK INSTRUCTION
4002
NUMRCD: DS 1  ; NUMBER OF RECORDS
4003
TRKADR: DS 1  ; TRACK ADDRESS
4004          SCTADR: DS 1 ; SECTOR ADDRESS
4005          BUFFER: DS 2 ; BUFFER ADDRESS
4007          BLOCKN: DS 1 ; BLOCK NUMBER, SD LINKED ONLY
4008          NXIOPB: DS 2 ; NEXT IOPB ADDRESS, LINKED
400A          UNITSL: DS 1 ; ADDR=400AH
400D          ORG 400DH

400D          UNIT: DS 1
400E          TRKRG0: DS 4 ; UNIT TRACK REGISTER ARRAY
4012          DDSDFL: DS 1 ; SINGLE/DATA DENSITY FLAG
4013          INDXCT: DS 1 ; INDEX COUNTER
4014          RDYBIT: DS 1 ; READY BIT STATUS FROM FDD
4015          RWFLG: DS 1 ; READ/WRITE FLAG
4016          DAMARK: DS 1 ; DATA ADDRESS MARK
4017          FLAGB: DS 1
4018          FLAGC: DS 1
4019          FLAGD: DS 1

401A          END  ZX202
<table>
<thead>
<tr>
<th>Address</th>
<th>Block</th>
<th>Buffer</th>
<th>Channel</th>
<th>CLrint</th>
<th>CTAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>0265</td>
<td>04000</td>
<td>00080</td>
<td>01CB</td>
<td>01CB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>MARK</th>
<th>MRCRC</th>
<th>NUMRCD</th>
<th>NXRPTB</th>
<th>RPT79</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>06000</td>
<td>04002</td>
<td>04008</td>
<td>6402</td>
<td>6402</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Port7B</th>
<th>Port89</th>
<th>PORT8B</th>
<th>RAMCLR</th>
<th>RDPRT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>06400</td>
<td>06401</td>
<td>0009</td>
<td>6500</td>
<td>6500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>RDRDY</th>
<th>RDSTAT</th>
<th>RDYBIT</th>
<th>RDYINT</th>
<th>RSYST1</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>00660</td>
<td>04014</td>
<td>0074</td>
<td>4004</td>
<td>4004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>RECAL</th>
<th>RWFLG</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>04015</td>
<td>04016</td>
<td>03057</td>
<td>03058</td>
<td>03059</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>SDGAP1</th>
<th>SDGAP3</th>
<th>SEEK</th>
<th>STEP</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>0613</td>
<td>027C</td>
<td>02A2</td>
<td>003C</td>
<td>003C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>SYNCPL</th>
<th>TRKADR</th>
<th>TRKREG</th>
<th>TRKREG</th>
<th>TRKREG</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>06300</td>
<td>000E</td>
<td>400E</td>
<td>400E</td>
<td>400E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>UNIT</th>
<th>UNITAB</th>
<th>WRCRC</th>
<th>WRCRC</th>
<th>WRCRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>01600</td>
<td>0400D</td>
<td>0400A</td>
<td>03000</td>
<td>03000</td>
<td>03000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>ZX202</th>
<th>01000</th>
<th>02000</th>
<th>03000</th>
<th>04000</th>
</tr>
</thead>
</table>
## Bill of Materials

<table>
<thead>
<tr>
<th>Item #</th>
<th>Qty.</th>
<th>Part#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R8, 9</td>
<td>2</td>
<td>10KOHM</td>
<td>Resistor</td>
</tr>
<tr>
<td>R2</td>
<td>1</td>
<td>1500OHM</td>
<td>Resistor</td>
</tr>
<tr>
<td>R5, 7</td>
<td>2</td>
<td>1KOHM</td>
<td>Resistor</td>
</tr>
<tr>
<td>R11, 12</td>
<td>2</td>
<td>2.2 KOHM</td>
<td>Resistor</td>
</tr>
<tr>
<td>U47, 48</td>
<td>2</td>
<td>2114-3</td>
<td>I. C.</td>
</tr>
<tr>
<td>U63</td>
<td>1</td>
<td>2716-ZX200</td>
<td>I.C. EPROM Version 1.1 for ZX-200</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>3300HM</td>
<td>Resistor</td>
</tr>
<tr>
<td>R113, 14</td>
<td>2</td>
<td>4308R-101-471</td>
<td>Bourns 8P SIP</td>
</tr>
<tr>
<td>R3, 4, 10</td>
<td>3</td>
<td>4700HM</td>
<td>Resistor</td>
</tr>
<tr>
<td>XU43</td>
<td>1</td>
<td>516-AG11D</td>
<td>Dip Socket</td>
</tr>
<tr>
<td>XU20-22</td>
<td>3</td>
<td>520-AG11D</td>
<td>Dip Socket</td>
</tr>
<tr>
<td>XU63</td>
<td>1</td>
<td>524-AG11D</td>
<td>Dip Socket</td>
</tr>
<tr>
<td>W3</td>
<td>1</td>
<td>530153-1</td>
<td>Jumper</td>
</tr>
<tr>
<td>XU64, 65</td>
<td>2</td>
<td>540-AG11D</td>
<td>Dip Socket</td>
</tr>
<tr>
<td>U11, 36, 41, 45, 52, 82, 84</td>
<td>7</td>
<td>74LS00</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U51</td>
<td>1</td>
<td>74LS02</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U9, 18, 42</td>
<td>3</td>
<td>74LS04</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U12</td>
<td>1</td>
<td>74LS08</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U24, 35, 61</td>
<td>3</td>
<td>74LS10</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U6, 59</td>
<td>2</td>
<td>74LS113</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U70</td>
<td>1</td>
<td>74LS12</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U53, 54, 55</td>
<td>3</td>
<td>74LS13</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U25</td>
<td>1</td>
<td>74LS155</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U39</td>
<td>1</td>
<td>74LS157</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U40</td>
<td>1</td>
<td>74LS163</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U15, 16</td>
<td>2</td>
<td>74LS165</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U4, 17, 31, 38, 83</td>
<td>5</td>
<td>74LS174</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U30</td>
<td>1</td>
<td>74LS244</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U33, 34</td>
<td>2</td>
<td>74LS266</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U32</td>
<td>1</td>
<td>74LS273</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U27</td>
<td>1</td>
<td>74LS279</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U23</td>
<td>1</td>
<td>74LS302</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U10, 69</td>
<td>2</td>
<td>74LS367</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U49</td>
<td>1</td>
<td>74LS373</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U14</td>
<td>1</td>
<td>74LS374</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U8, 13</td>
<td>2</td>
<td>74LS38</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U76-81</td>
<td>6</td>
<td>74LS670</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U5, 19, 50</td>
<td>3</td>
<td>74LS74</td>
<td>I.C., LS TTL</td>
</tr>
<tr>
<td>U68</td>
<td>1</td>
<td>74S00</td>
<td>I.C. TTL SCHOTTKY</td>
</tr>
<tr>
<td>U1, 60</td>
<td>2</td>
<td>74S04</td>
<td>I.C.</td>
</tr>
<tr>
<td>U57</td>
<td>1</td>
<td>74S10</td>
<td>I.C. TTL SCHOTTKY</td>
</tr>
<tr>
<td>U2, 3</td>
<td>2</td>
<td>74S163</td>
<td>I.C. TTL SCHOTTKY</td>
</tr>
<tr>
<td>U20, 21</td>
<td>2</td>
<td>74S471</td>
<td>TI TBP18S22 Bipolar PROM</td>
</tr>
<tr>
<td>U58, 71</td>
<td>2</td>
<td>74S74</td>
<td>I.C.</td>
</tr>
<tr>
<td>Item #</td>
<td>Qty</td>
<td>Part #</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>----------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>U65</td>
<td>1</td>
<td>8085A-2</td>
<td>I.C., 8-Bit CPU</td>
</tr>
<tr>
<td>U64</td>
<td>1</td>
<td>8257-5</td>
<td>DMA CHIP</td>
</tr>
<tr>
<td>XW3</td>
<td>3</td>
<td>B7022-A</td>
<td>AMP Wire Wrap Post</td>
</tr>
<tr>
<td>U43</td>
<td>1</td>
<td>93446PC</td>
<td>I.C., BIPOLAR PROM</td>
</tr>
<tr>
<td>U66</td>
<td>1</td>
<td>9401</td>
<td>IC CRC Generator &amp; Checker</td>
</tr>
<tr>
<td>U46, 62, 74, 75</td>
<td>4</td>
<td>AM2946</td>
<td>I.C.</td>
</tr>
<tr>
<td>U29</td>
<td>1</td>
<td>AM2956</td>
<td>I.C.</td>
</tr>
<tr>
<td>U72, 73</td>
<td>2</td>
<td>AM2957</td>
<td>I.C.</td>
</tr>
<tr>
<td>C1</td>
<td>1</td>
<td>C40C100K</td>
<td>Centralab Cap</td>
</tr>
<tr>
<td>C2-8, &amp; 10-12</td>
<td>41</td>
<td>C41C104K</td>
<td>Centralab Cap</td>
</tr>
<tr>
<td>U7</td>
<td>1</td>
<td>CD4040BE</td>
<td>I.C. CMOS</td>
</tr>
<tr>
<td>Y1</td>
<td>1</td>
<td>CY22A</td>
<td>Crystek 20 MHz Crystal</td>
</tr>
<tr>
<td>Y2</td>
<td>1</td>
<td>CY6C</td>
<td>Xtal 6.144 MHz</td>
</tr>
<tr>
<td>C9</td>
<td>1</td>
<td>TE-1211</td>
<td>Sprague Cap</td>
</tr>
<tr>
<td>--</td>
<td>1</td>
<td>PCZX-200A</td>
<td>PC Board, ZX-200A</td>
</tr>
</tbody>
</table>