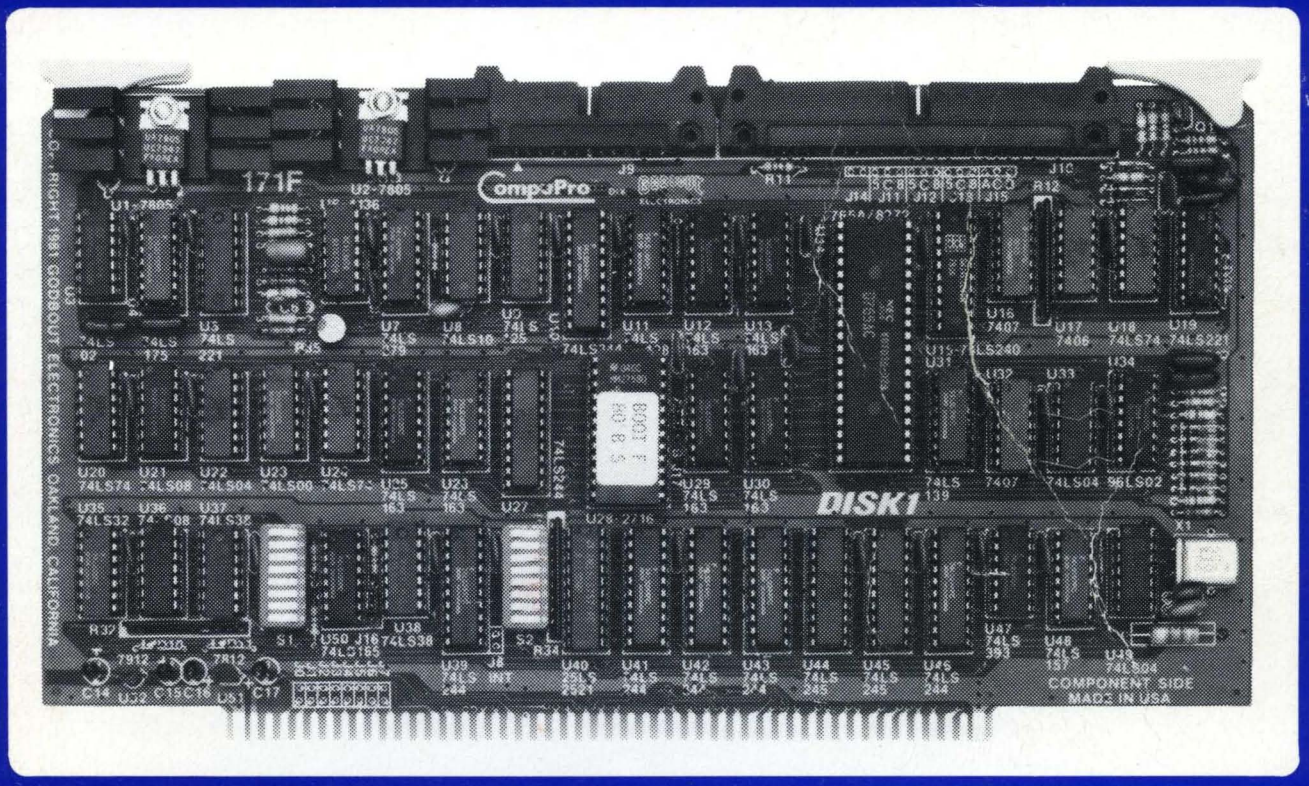
DISK 1™

TECHNICAL MANUAL



IEEE 696 / S-100

ARBITRATED 24 BIT DMA FLOPPY DISK CONTROLLER

171F mac

CompuPro™ division GODBOUT ELECTRONICS

---------- **DISCLAIMER** ----------

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

This document was proofread with the aid of SpellGuard™ from Inovative Software Applications, Menlo Park, CA.

TABLE OF CONTENTS

➤➤➤ WARNING ◄◄◄

Not all floppy disk controllers generate true IBM compatible 3740 and SYSTEM 34 formats. Therefore, we strongly recommend that you do not use the DISK 1 to copy data onto a diskette that has been formatted by another controller! The proper procedure is to format new diskettes using the DISK 1, and copy the contents of other diskettes onto the newly formatted diskettes. (SOFTWARE SECTION; Software User's Guide)

The technical reason for this problem arises from the fact that almost all versions of the 1791 type floppy controller insert a byte of 00s immediately following the header CRC bytes. This byte of 00s is not specified in either IBM standard, and may confuse the 8272/765 controller on the DISK 1.

If your controller generates a true IBM type format, or you are using diskettes formatted by IBM, you will not experience this problem.

ATTENTION PURCHASERS OF CP/M-80

If purchasing CP/M-80 with your DISK 1 controller, you will receive a diskette that contains your serialized version of CP/M-80 (for a 32K system) along with several support utilities and special BIOS files. This diskette should be used to create a working copy only, and not altered in any way. The following procedure should be strictly followed.

(1) Read the Software User's Guide in the SOFTWARE SECTION.

(2) Boot your system using the provided diskette.

(3) Run the FORMAT program and create a scratch diskette with a format identical to the format of your master diskette. (Your master will indicate its format on the label, i.e., 2D-256 = 256 byte format, 2D-1024 = 1024 byte format, etc.

(4) Run the COPY program and create a duplicate of your entire (system and data tracks) master diskette. (see NOTE (2) below)

(5) Remove your master diskette and store it in a safe place. Do not use this diskette again unless you damage your working copies. There will be a $35 charge for recreating your master diskette if you damage or alter it.

(6) Use your new working copy for all alterations that are made to your system.

NOTE (1): On your copy of CP/M you will receive two versions of the CompuPro BIOS. One version may be altered and reassembled using the CP/M supplied assembler (ASM), and another version must be reassembled and linked under the ACT assembler from SORCIM. The hex code from these files is identical, however, ASM will not assemble the files written for ACT.

NOTE (2): If you wish to change the density of your working disk or go from single sided to double sided, you must use SYSGEN to create the system, and PIP to transfer the data files. The COPY utility will only copy diskettes of identical formats.

ABOUT DISK 1

Congratulations on your decision to purchase the DISK 1 floppy disk controller. DISK 1 has been designed to be the highest performance floppy disk interface available that fully complies with with the IEEE 696/S-100 bus standard. Due to its provision for ready expansion and modification as the state of the computing art improves, the S-100 bus is the professional level choice for commercial, industrial, and scientific applications. We believe that this board, along with the rest of the S-100 portion of the CompuPro family, is one of the best boards available for that bus.

Features such as fully arbitrated DMA data transfer with 24 bits of addressing, 3rd generation LSI floppy disk controller, on-board phantom BOOT EPROM, I/O mapping for uninterrupted memory space, and a startup serial port make the DISK 1 another proud member of the CompuPro family.

TECHNICAL OVERVIEW

The DISK 1 was specifically designed to give the user one of the most powerful floppy disk interfaces available for the S-100 bus. Designed for full electrical and mechanical compatibility with the IEEE 696/S-100 bus standard, this board boasts several innovative features not found on currently available disk controllers. These features include 24 bit DMA data transfers with the ability to cross 64K boundaries, an advanced, 3rd generation floppy disk controller made by NEC or INTEL (765A or 8272), priority arbitration for the on-board DMA circuitry that will allow up to 16 temporary bus masters without conflict, an on-board BOOT EPROM with the capability of supporting eight different processors or BOOT routines, and a start-up serial port for ease of system initialization.

Other features standard to all CompuPro boards include thorough bypassing of all supply lines to suppress transients, on-board regulators, and low power Schottky TTL and MOS technology integrated circuits for reliable, cool operation. All this and sockets for all IC's go onto a double sided, solder masked printed circuit board with a complete component legend.

HOW TO GET YOUR DISK 1 UP AND RUNNING WITHOUT READING THE MANUAL

This section is for the user who is so anxious to see the DISK 1 running that he doesn't want to read the manual. This section will tell you how to set up the DISK 1 board as well as many other CompuPro boards so that it can run CP/M-80 or CP/M-86 in your system with either the on-board serial channel or any other supported serial channel.

We strongly recommend that you relax and read the manual. If, after reading and following the directions in this section, your system does not work, don't panic! Read the manual!

SYSTEM CONFIGURATIONS USING COMPUPRO HARDWARE

The following guide will help the user to configure CompuPro hardware for operation with CP/M type operating systems. This list covers boards in production at the date of printing, and if your particular CompuPro board is not listed, consult the individual manual. For the configuration of CompuPro hardware for the OASIS operating systems, refer to the configuration guide supplied by Phase One.

The following list will describe the proper switch settings and jumper options required by each CompuPro board when run with the DISK 1 controller.

DISK 1 - The standard switch settings for running CP/M are as follows:

"OFF"	S1	"ON"
	1	>
<	2	
<	3	
<	4	
<	5	
<	6	
<	7	
<	8	

"OFF"	S2	"ON"
*	1	*
*	2	*
<	3	
<	4	
	5	>
	6	>
	7	>
	8	>

J16 = Short B-C
J17 = Short A-C for CPU-Z or CPU 8085/88
J17 = Short B-C for CPU 8086/87

S2 POSITION 1	S2 POSITION 2	I/O BOARD SUPPORTED UNDER CP/M-80 AND CP/M-86
"ON"	"ON"	INTERFACER 1 or 2; Console at 00H, LPT List at 02H
"ON"	"OFF"	DISK 1 serial port as Console - CP/M-80 ONLY!!!
"OFF"	"ON"	SYSTEM SUPPORT 1 at 50H; Console 9600 baud, List as above
"OFF"	"OFF"	INTERFACER 3 or 4; 9600 baud, Console user 7, List user 6

These settings select DMA arbiter priority 15, port C0H-C3H, wait states enabled, and the BOOT routine as selected.

CPU-Z - The standard switch settings for running the CPU-Z with DISK 1 are:

"OFF"	S1	"ON"
<	1	
<	2	
<	3	
<	4	
<	5	
<	6	
<	7	
<	8	

"OFF"	S2	"ON"
<	1	
<	2	
<	3	
<	4	
<	5	
<	6	
<	7	
<	8	

"OFF"	S3	"ON"
<	1	
<	2	
<	3	
<	4	
	5	>
<	6	
<	7	
<	8	

This assumes that you are not planning to run at the slow CPU-Z speed and that you do not require wait states generated on the CPU.

CPU 8085/88 - The standard switch settings for running the CPU 8085/88 with DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"
	1	>	<	1		<	1	
	2	>	<	2			2	>
	3	>	<	3		<	3	
<	4		<	4		<	4	
<	5		<	5		<	5	
<	6		<	6		<	6	
*	7		<	7		<	7	
<	8		<	8		<	8	

Switch S4 should be set to the desired speed. Switch S1-7 should be "OFF" in systems having a front panel and "ON" in systems without a front panel. A wait state is inserted in all I/O cycles.

CPU 86/87 - The standard switch settings for running the CPU 86/87 with DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"	"OFF"	S4	"ON"	"OFF"	S5	"ON"
<	1		<	1		<	1		<	1		<	1	
<	2		<	2		<	2		<	2		<	2	
<	3		<	3			3	>		3	>	<	3	
<	4		<	4			4	>		4	>	<	4	
<	5		<	5		<	5			5	>	<	5	
<	6		<	6		<	6			6	>	<	6	
<	7		<	7			7	>		7	>	<	7	
<	8		<	8		<	8			8	>	<	8	
<	9		<	9						9	>	<	9	
<	10		<	10						10	>	<	10	

JUMPERS: Make sure there is a shorting plug installed at jumper location J8. J8 is located near the top left-hand corner of the board. There should be no shorting plugs installed at jumper locations J0 thru J7, which are located near the bottom left-hand corner of the board.

INTERFACER 1 - The standard switch settings for running the INTERFACER 1 as the console and list device I/O board with DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"
	1	>		1	>	<	1	
<	2			2	>		2	>
<	3			3	>		3	>
<	4			4	>		4	>
	5	>		5	>		5	>
<	6			6	>		6	>
<	7			7	>		7	>
<	8		<	8		<	8	

This configuration provides the Console device on channel "A" (ports 00H and 01H) at 9600 baud and the List device on channel "B" (ports 02H and 03H) at 9600 baud.

1. Jumpers J3 through J6 should be set in Slave mode.
2. We recommend that Jumpers J7 and J8 be set to match your terminal

and printer specifications and that Jumpers J9 and J10 have traces 2-15, 4-13, 6-11, and 8-9 cut.
3. Baud rates may be altered if desired by resetting S1.

INTERFACER 2 - The standard switch settings for running the serial section of the INTERFACER 2 as the console device I/O board with DISK 1 are as follows. The list device is assumed to be another INTERFACER 1/2 serial port at locations 02H and 03H.

"OFF"	S2	"ON"
	1	>
<	2	
<	3	
<	4	
<	5	
<	6	
<	7	
<	8	

"OFF"	S3	"ON"
	1	>
	2	>
	3	>
	4	>
	5	>
	6	>
	7	>
<	8	

"OFF"	S4	"ON"
	1	>
	2	>
<	3	
	4	>
	5	>
	6	>
	7	>
<	8	

This configuration provides the Console device on the serial channel (ports 00H and 01H) at 9600 baud and the List device on another INTERFACER 1 or 2 channel.

1. Jumpers J5 and J6 should be set in Slave mode.
2. We recommend that you set J9 to match your terminal specifications and that Jumper J10 have traces 2-15, 4-13, 6-11, and 8-9 cut.
3. Baud rate may be altered if desired by resetting S2.
4. The setting of S1 is dependent on your parallel requirements and does not affect the setting of the serial channel.
5. The address chosen for the parallel ports is the CompuPro standard of 08H through 0BH so that it does not overlap the serial port.

INTERFACER 3 - The standard switch settings for running the INTERFACER 3 as the console and list device I/O board with DISK 1 are:

"OFF"	S1	"ON"
<	1	
	2	>
	3	>
	4	>
<	5	
	6	>
	7	>
	8	>

This configuration provides the Console device on User 7 (left most) at 9600 baud and the List device on User 6 (2nd from left) at 9600 baud.

1. Jumpers J1 and J2 should be set in Slave mode.
2. We recommend that Jumpers J3 through J14 be removed.
3. Jumpers J15 and J16 may remain unwired.
4. Jumper J17 should be set for an appropriate number of wait states for your system speed.
5. USART assumes 8 data bits, 2 stop bits, no parity, asynch operation, RTS and DTR output in "spacing" (+12V) condition.

6. USART requires that CTS and DSR* be "spacing" (+12V). *(DSR is used to determine if printer is ready on User 6 only)

INTERFACER 4 - The standard switch settings for running the INTERFACER 4 as the console and list device I/O board with DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"
<	1			1	>	<	1	
<	2			2	>	<	2	
<	3		<	3		<	3	
<	4		<	4		<	4	
<	5			5	>		5	>
<	6			6	>		6	>
<	7			7	>		7	>
<	8		<	8			8	>
<	9			9	>			
<	10		<	10				

This configuration provides the Console device on User 7 (left most serial port-CONN3-A) at 9600 baud, the List device on User 6 (middle serial port-CONN3-B at 9600 baud or the CENTRONICS PARALLEL Channel-CONN 2 / SEE ITEM #3 BELOW), and the UL1 device on User 5 (right most serial channel).

1. Jumper Sockets JS1-JS3 should contain 8 position DIP SHUNTS (Slave mode for terminals or printers).
2. Jumper J6, J7, or J8 should be installed for 1, 2, or 3 wait states.
3. For a serial list device, use CONN3-B with jumper J26 having "A" shorted to "B", and "C" shorted to "D".
   For a parallel list device using the CENTRONICS PARALLEL Channel, use CONN2 with jumper J26 having "A" shorted to "C", and "B" shorted to "D"
   For an EPSON type printer, install J2-bottom, J3-top, and J4-bottom.
4. All other Jumpers may be removed.
5. USART assumes 8 data bits, 2 stop bits, no parity, asynch operation, RTS and DTR output in "spacing" (+12V) condition.
6. USART requires that CTS and DSR* be "spacing" (+12V). *(DSR is used to determine if printer is ready on User 6 only)

SYSTEM SUPPORT 1 - The standard switch settings for running the SYSTEM SUPPORT 1 as the console device I/O board with DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"
<	1			1	>	<	1	
<	2			2	>	<	2	
<	3			3	>	<	3	
	4	>		4	>	<	4	
<	5		<	5			5	>
<	6		<	6		<	6	
	7	>	<	7			7	>
<	8		<	8		<	8	

This configuration provides the Console device on the SYSTEM SUPPORT 1 serial channel at 9600 baud and the List device as an INTERFACER 1/2 serial channel at ports 02H and 03H.

1. USART assumes 8 data bits, 2 stop bits, no parity, RTS and DTR output in "spacing" (+12V) condition. USART requires that CTS be "spacing" (+12V).
2. J2 and J8 should be bussed straight across with a shunt.
3. For CP/M-86, either a 6116 RAM chip or a GO86 JUMP EPROM should be installed in U16. The above settings place it at 0FF000H for the 8086/88.

RAM 16 - The standard switch settings for operating a RAM 16 as the first 64K of system memory with the DISK 1, this setting places 64K in extended page 0 (base page), are:

"OFF"	S1	"ON"
	1	>
	2	>
	3	>
	4	>
	5	>
	6	>
	7	>
	8	>

RAM 17 - The standard switch settings for operating a RAM 17 as the first 64K of system memory with the DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"
*	1	*	<	1	
	2	>		2	>
	3	>		3	>
	4	>		4	>
	5	>		5	>
	6	>		6	>
	7	>		7	>
	8	>		8	>
	9	>		9	>
*	10	*		10	>

In systems with front panels, S1-1 should be "ON" and S1-10 should be "OFF". In systems without front panels, S1-1 should be "OFF" and S1-10 should be "ON". This setting places all 64K active and resident in extended page 0 (base page).

RAM 20 - The standard switch settings for operating two RAM 20 boards as the first 64K of system memory with the DISK 1 are:

"OFF"	S1	"ON"	"OFF"	S2	"ON"	"OFF"	S3	"ON"	"OFF"	S4	"ON"
	1	>	<	1			1	>	<	1	
	2	>	<	2			2	>	<	2	
	3	>	<	3			3	>	<	3	
	4	>	*	4	*		4	>	<	4	
	5	>	<	5			5	>	<	5	
	6	>		6	>		6	>	<	6	
	7	>	<	7			7	>	<	7	
	8	>	<	8			8	>	<	8	

To set the first RAM 20 (0-32K) should have S2-4 in the "OFF" position and the second RAM 20 should have S2-4 in the "ON" position. The two boards will form 64K (all active and resident) in extended page 0 (base page).

RAM 21

The standard switch settings for operating a RAM 21 as the first 128K of system memory with the DISK 1, this setting places 64K in extended page 0 (base page), and 64K in extended page 1, are as follows:

"OFF"	S1	"ON"
	1	>
	2	>
	3	>
	4	>
	5	>
	6	>
	7	>
	8	>

DRIVE INTERFACE - 8 INCH DRIVES

The interface to soft media 8" drives is standard except that the stepper motors must be enabled at all times (not tied to drive select or head load). This causes the steppers to be powered at all times (they will get warm), and allows stepping without the lamp on the front of the drive being "ON" (so be careful). In addition, the head load signal should not be tied to drive select since the 765A/8272 is always scanning the drives (this would result in a buzz). Standard 50 pin ribbon cable should be used to connect the drives to the controller, and the last drive in the line should be terminated as specified in the drive manual.

NOTE: Due to the steppers being enabled at all times, your disk power supply must be able to handle full load on the +24V line all the time and your drive box must have adequate cooling.

SHUGART 800/801: On a Shugart 800/801 drive, the shorting plugs should be installed and removed on all drives as shown on the following example:

INSTALLED (T2, A, B, C, Z, 800)
REMOVED (D, DC, X, Y, HL, DS)

Drive select (DSx) should be installed appropriately and the terminators T1, and T3-T6 should be installed on the last drive of the cable.

SHUGART 850/851: On a Shugart 850/851 drive, the shorting plugs should be installed and removed as shown below, and the DIP shunt should be altered as required. NOTE: SHUGART changes the drive P.C. board on the 850/851 regularly, so this information is for the MLC 12 series and may be dated.

INSTALLED (C,2S,S2,IW,RS,DL,IT,AF,M,850)
REMOVED (FS,TS,Y,DS,HLL,HI,D,DC,NF)

SHUNT: CUT HL AND X, ALL OTHERS INTACT. TERMINATOR RESISTORS SHOULD BE INSTALLED ON THE LAST DRIVE OF THE CABLE.

QUME DATA TRACK 8: On Qume Data Track 8 drives, the shorting plugs should be installed and removed as shown below, and the DIP shunt should be altered as required.

INSTALLED (C,2S,DL,DSx)
REMOVED (T40,GND,DS,D,DC,Y,HA)

SHUNT: CUT HL AND X, ALL OTHERS INTACT. TERMINATOR RESISTORS SHOULD BE INSTALLED ON THE LAST DRIVE OF THE CABLE.

SIEMENS MODEL FDD 100-8: On Siemens Model D drives, the shorting plugs should be installed and removed as shown below.

INSTALLED (SS,SE,E,D,RR,0,2,F,RI,L,U,H)
REMOVED (HS,8,16,32,1,TE,A,V,B,J,K,M,G)

PC BOARD MODIFICATION FOR RUNNING TWO OR MORE DRIVES: This modification involves the raw read data on pin 46 and the step inhibit signal on IC 6C (7438).

1. Remove the P.C. board and cut the trace leading to IC 6C pin 9.
2. Connect IC 6C pin 9 to IC 6C pin 12 and re-install the P.C. board.

TANDON MODEL 848 On a TANDON 848 drive, the drive will run as shipped except for the alteration of the shunt and the installation of the drive select plug. The following connections should be observed.

INSTALLED (DC,2S,S2,C,RR,RI,WP)
REMOVED (Y,DS,DL,HLL,D,NP,RM,S1)

SHUNT: CUT HL AND X (5 AND 2 OF U3), ALL OTHERS INTACT. TERMINATOR RESISTORS SHOULD BE INSTALLED ON THE LAST DRIVE OF THE CABLE.

MITSUBISHI MODEL M2894-63 On a MITSUBISHI M2894 drive, the shorting plugs should be installed and removed as shown below, and the DIP shunt (PJ1-8) should be altered as required.

INSTALLED (E,Z,2S,I,R,S2,IW,C,WP)
REMOVED (A,F,Y,DC,IT,S3)

SHUNT: OPEN PJ4 AND PJ5, ALL OTHERS CLOSED. TERMINATOR RESISTORS SHOULD BE INSTALLED ON THE LAST DRIVE OF THE CABLE.

5.25 INCH DRIVES

Connection to 5.25" minifloppies requires that the DISK 1 board be modified as per the instructions entitled "MODIFICATIONS FOR MINIFLOPPIES" prior to the

following drive connections. Standard 34 pin ribbon cable should be used. However, the female transition connector must be offset to the extreme right when seated in connector J10. The serial port may be used as the motor "ON-OFF" control bit if switch S1-3 is placed in the "ON" position.

No modifications need to be made to minifloppy drives except to select the proper drive select line on the programming shunt and leave it intact. If your minifloppy drive does not drive the READY* line, a jumper must be installed between "C" and "B" of J15. If it does drive the ready line, a jumper must be installed between "C" and "A" of J15. Jumpers J11-J13 should be set for 5.25" operation and J14 should be installed.

TRYING TO BOOT THE CONTROLLER

Before inserting your diskette, you should see the following things:

(1) Your drive activity light should be glowing very dimly to indicate that the floppy controller is scanning the drives.

(2) If the controller is trying to boot, you will see the activity light of drive #1 flashing on and off approximately once a second (dependent on the CPU speed).

If both of these are present, insert your write protected diskette and listen. If using an I/O port other than the on-board serial port, you should hear several (approx. four) clicking sounds from the drive and see the sign-on message. If using the on-board serial port you should hear one click, and after striking the "U" on the terminal, you should hear the four clicks and see the sign-on message.

TROUBLESHOOTING

If you are having problems getting your DISK 1 up and running and you have read the previous sections of this manual completely, read on. The following section may help you solve your problem.

PROBLEM: ONE OR MORE OF YOUR DRIVE ACTIVITY LIGHTS STAYS "ON" BRIGHT ALL THE TIME.

Solution: This almost always indicates that the drive cable is backwards at either the drive box connector or at the drive. This may be verified by removing the 50 pin cable from either the board or the box. If the lamp goes off, the cable is reversed. NOTE: When using our controller with drives from Morrow Designs, the cable must be reversed due to their non-standard connector pinouts.

PROBLEM: ACTIVITY LAMPS DO NOT GLOW DIMLY OR FLASH BRIGHT

Solution: This generally indicates that the drive is jumpered wrong or there is a controller fault. Make sure that the drives are jumpered correctly and that the activity lamp is activated by drive select and not head load. Make sure that the drive has all of the different DC voltages that it requires. If the

lamp still does not light, there could be a problem in either the controller or an open in the 50 pin drive cable.

PROBLEM: ACTIVITY LAMP GLOWS DIMLY BUT DOES NOT FLASH

Solution: This typically indicates either that the CPU is not executing the code in the BOOT EPROM due to the memory not being phantomed OFF, or that the host CPU does not have its jump-on-reset circuit turned off. The lamp will flash even if there is nothing in the system but the CPU and the DISK 1 board (no RAM!). Try removing everything but the CPU and the DISK 1 and verifying that the lamp flashes. If it does not, either of these boards could be at fault. Review your switch settings.

PROBLEM: DRIVE LOADS HEAD THEN UNLOADS AND REPEATS

Solution: If the board clicks and then pauses, and then repeats itself, this is typically an indication that the controller is unable to read the data from the drive. This could be from the data on the diskette being bad, the phase lock loop being out of adjustment, the DMA cycle being inhibited by the CPU, or a hardware problem on either the drive, the cable, or the DISK 1 board. Try to isolate the problem by substitution if possible, otherwise check switch settings and drive hardware for proper configuration. CAUTION: Controller calibration should only be performed by an authorized dealer, OEM or the factory.

PROBLEM: DRIVE LOADS HEAD ONCE THEN STOPS

Solution: A single loading of the head and then nothing generally indicates that the controller is reading the first several sectors OK but either the data is transferred into memory improperly or the system memory is either bad or misaddressed. Improper transfer into memory generally occurs only with dynamic RAM boards that reley on specific CPU timing. If you have dynamic memory, make sure that is can handle DMA and that it generates its own refresh timing. Otherwise make sure that the RAM is addressed properly and in the proper extended page (page 0). Remember that for CP/M-80 you only need 32K and for CP/M-86 you need 64K. If you can run a RAM test, verify that that your memory is OK.

PROBLEM: IMSAI FRONT PANEL "LOCKS UP" WHEN DISK 1 IS INSTALLED

Solution: The new IEEE 696/S-100 bus standard calls for lines 20 and 70 to be grounded on the bus. When the IMSAI front panel was designed, many memory boards had the capability to be "protected" from accidental writes to the board. To accommodate these boards, the IMSAI front panel grounded line 70 but pulled up line 20 with a resistor that also enabled all the front panel functions. When DISK 1 is inserted, line 20 is grounded and the front panel is disabled. The solution is to cut line 20 on the front panel since this line is now defined as ground. ADDITIONAL NOTE: To access system RAM from the front panel after reset or power-up without running the DISK 1, the BOOT INHIBIT switch (S1-4) must be set "ON" to disable the BOOT EPROM.

DISK INTERFACE PORT MAP

The DISK 1 interface uses a block of four port addresses for communication between it and the host processor. DISK 1 occupies no memory space of the host processor and performs all data transfers via DMA. The address of the first port is switch settable to any address which is a multiple of four. The ports will be referred to as relative ports 0 - 3.

RELATIVE PORT — FUNCTION

0 . . . FDC main status register (read only)

1 . . . FDC data register

2 . . . STATUS register (when read)
D7 indicates FDC INT output is asserted.
No other bits are significant.

DMA address register (when written)
The DMA address register is actually a push- down stack of three one byte registers. A three byte, twenty-four bit DMA address should be loaded most significant byte first.

3 . . . SERIAL PORT
When read, bit D7 will contain the current status of the serial input line.
When written, bit D7 should contain the new state for the serial output line. The state will be latched until changed.
When written with a "0" in bit D0, the BOOT EPROM will be disabled. A system reset is required to re-enable the BOOT EPROM.

(1) Bit D7 = 1 indicates a SPACING condition or binary 0. (A start bit is a space)

(2) Bit D7 = 0 indicates a MARKING condition or a binary 1.

(3) On RESET, the output will be cleared to the MARKING state.

If the SERIAL port is not required for RS232 communications, a switch will allow the output data latch to serve as MOTOR-ON for the 5.25 inch floppy disks.

COMPUPRO 8 INCH DRIVE INTERFACE ADDRESS

The current implementation of all software packages written for the DISK 1 controller and 8 inch drives including the CompuPro BIOS for CP/M-80 and CP/M-86, as well as single and multi-user OASIS, requires that the base port address be set to C0H.

ACTUAL PORT	FUNCTION
C0	FDC main status register (read only)
C1	FDC data register
C2	STATUS register (when read)
	DMA address register (when written)
C3	SERIAL PORT

COMPUPRO 5.25 INCH DRIVE INTERFACE ADDRESS

The recommended base port address for the DISK 1 controller and 5.25 inch drives is CCH.

ACTUAL PORT	FUNCTION
CC	FDC main status register (read only)
CD	FDC data register
CE	STATUS register (when read)
	DMA address register (when written)
CF	SERIAL PORT

PORT ADDRESSING

DIP switch S2, positions 3 thru 8 are used to select the base address of the four port block in a binary fashion as shown below:

SWITCH POSITION	ADDRESS BIT	
3	A7	
4	A6	
5	A5	"ON" = "0"
6	A4	"OFF" = "1"
7	A3	
8	A2	

EXAMPLE: To address this board at addresses C0H thru C3H for the CompuPro CP/M BIOS, positions 3 and 4 would be "OFF" and positions 5 thru 8 would be "ON".
EXAMPLE: To address this board at addresses CCH thru CFH, positions 3, 4, 7 and 8 would be "OFF" and positions 5 and 6 would be "ON".

SERIAL PORT

The serial port on the DISK 1 was designed for initial system startup only and should not be used as the console device for longer than required to patch the BIOS. Since this port is controlled entirely by software, its baud rate limitations are dependent on the host processor's clock rate. With the current implementation of the BIOS, the speed of the terminal is calculated when an upper case "U" is struck on the keyboard, and the sign-on message is then displayed. The terminal may be set to any baud rate, but the following maximum baud rates should be observed for best reliability:

CPU CLOCK SPEED	MAX. BAUD RATE
2MHz	600 baud
4MHz	1200 baud
6MHz	2400 baud

If your processor does not run at any of the above speeds, don't worry- these are only suggested maximum rates. Connector J9 accepts a standard INTERFACER cable with ground on pin 7, transmit data on pin 3, and receive data on pin 2. No RS-232 handshaking lines are provided.

INTERRUPTS

The DISK 1 is capable of running in either a polled mode or an interrupt driven mode that is particularly suited for multi-user environments. The STATUS port (relative port 2) allows the user to run in the polled mode by sampling the interrupt output of the floppy disk controller on data bit 7. To run in an interrupt driven mode, the interrupt output of the floppy disk controller is driven onto one of the vectored interrupt lines (VI0* thru VI7*) or the INT* line of the S-100 bus. This is accomplished by installing a shorting plug or a #30 wrap wire across the posts at jumper locations J0 thru J7 or J8. Jumpers J0 thru J7 correspond directly to VI0* thru VI7*, and J8 is for INT*. It is recommended that the highest priority vectored interrupt be used to insure that it is not accidentally masked off.

BOOT EPROM

The BOOT EPROM contains the software routines required to load the initial sectors of the disk operating system into memory for system startup. The exact contents of this routine is covered in the SOURCE LISTINGS section under ROM Boot.

Upon power-up, the BOOT EPROM will appear as 256 bytes of memory at the host CPU's reset address. If the CPU does not reset to a location on a 256 byte boundary, the BOOT EPROM will align itself on the nearest 256 byte page. For example, an 8085 or a Z-80 will reset to 0000H, which is on a 256 byte boundary. Therefore, in this case the BOOT EPROM will appear from 0000H to 00FFH. An 8088 or an 8086 resets to 0FFFF0H, which is not page aligned, therefore, the BOOT EPROM will appear from 0FFF00H to 0FFFFFH.

The DISK 1 requires that a minimum of 256 bytes of system RAM at the same location as the BOOT EPROM be capable of responding to PHANTOM* by disabling itself. This RAM may be of any amount greater than 256 bytes since the BOOT routine requires no memory for proper operation.

BOOT EPROM ROUTINE ADDRESSING

The BOOT EPROM is capable of holding two sets of four switch selectable BOOT routines of up to 256 bytes each. Positions 1 and 2 of switch S2 select one of the four routines in a binary fashion and jumper J17 selects either the low or high half of the EPROM as shown below:

J17 POSITION	SWITCH S2 POSITION 1	2	EPROM STARTING ADDRESS	BOOT ROUTINE #
B	ON	ON	000H	0
B	ON	OFF	100H	1
B	OFF	ON	200H	2
B	OFF	OFF	300H	3
A	ON	ON	400H	4
A	ON	OFF	500H	5
A	OFF	ON	600H	6
A	OFF	OFF	700H	7

NOTE: In some cases, the DISK 1 may be shipped set for routines 4-7 rather than 0-3. In these cases, please leave J17 as shipped and treat the routines as 0-3 only.

BOOT ROUTINE SHORT DESCRIPTION

As shipped, the BOOT EPROM contains routines for loading several sectors of track 0 into memory and passing on the value associated with the particular BOOT routine. The value passed on allows the proper console I/O routine to be selected as described in a following section. The specific details for passing this value is contained in the description of the CP/M-80 BIOS. The BIOS for CP/M-80 and CP/M-86 (for running under CPU 8085/88) as implemented interprets this value as shown below. Software designed CPU 68K will contain information describing which routine to use.

#4 This BOOT routine specifies a standard INTERFACER 1 or 2 serial port at locations 00H and 01H for the console device and the LPT list device at locations 02H and 03H. (For CP/M-80 and CP/M-86)

#5 This BOOT routine specifies the DISK 1 on-board serial port as the console device and the list device the same as routine #1. (For CP/M-80 only! CP/M-86 does not support the on-board serial port.)

#6 This BOOT routine specifies the serial channel on the SYSTEM SUPPORT 1 board addressed at 50H and for 9600 baud as the console device and the list device the same as in routine #1. (For CP/M-80 and CP/M-86)

#7 This BOOT routine specifies two serial channels on an INTERFACER 3 or 4 addressed at 10H and set for 9600 baud, with user 7 as the console and 6 as the list device. (For CP/M-80 and CP/M-86)

#0 This routine is identical to #4 but for CPU 8086/87 only.
#1 This routine is not used by CPU 8086/87.
#2 This routine is identical to #6 but for CPU 8086/87 only.
#3 This routine is identical to #7 but for CPU 8086/87 only.

WAIT STATE ENABLE

The DISK 1 is capable of inserting wait states into the BOOT EPROM read as well as the I/O and DMA read or write cycles when fast processors are being used. The wait states become necessary when the access time of the BOOT EPROM and the floppy disk controller are longer than the fetch time of the host processor. The wait states are enabled by placing position 1 of switch S1 in the "ON" position. When enabled, the EPROM will have 5 wait states inserted, and the I/O and DMA cycles can have either 2, 3 or 4 wait states. With J16 in position "A", 2 wait states will be inserted, with J16 in position "B", 3 wait states will be inserted, and with J16 removed, 4 wait states will be inserted.

BOOT ENABLE/INHIBIT

The BOOT EPROM may be disabled by putting position 4 of S1 in the "ON" position, and enabled by placing it in the "OFF" position. A possible reason for disabling the BOOT routine would be if two or more DISK 1 controllers were

placed in the system at one time or the DISK 2 hard disk controller was the BOOTING device. In this case, more than one controller trying to boot would cause a system conflict.

Remember, the BOOT hardware on the DISK 1 board requires that a 256 byte page of memory respond to PHANTOM* at the host processor's reset address. If the memory residing at this address does not respond to PHANTOM*, a bus drive conflict will occur and possible damage could result.

ARBITER AND PRIORITY SELECTION

The DISK 1 controller allows multiple DMA devices to be active on the S-100 bus at one time. As long as a DMA board (temporary bus master) conforms to the IEEE 696 specifications concerning DMA arbitration and prioritization, up to 16 different bus masters may gain use of the bus in order of their assigned priority. Remember, there should never be more than one temporary bus master at the same priority level.

The priority of the DISK 1 board is selected in a binary fashion on positions 4 5 thru 8 on DIP switch S1 as shown below:

SWITCH POSITION (S1)	PRIORITY LEVEL	VALUE	
5	PRIORITY 3	8	
6	PRIORITY 2	4	"ON" = NO VALUE
7	PRIORITY 1	2	"OFF" = VALUE
8	PRIORITY 0	1	

EXAMPLE:
1. For the highest priority (15), positions 5 thru 8 would be "OFF" (8+4+2+1=15).
2. For priority 9, positions 5 and 8 would be "OFF" and positions 6 and 7 would be "ON" (8+1=9).
3. For the lowest priority (0), positions 5 thru 8 should be "ON".

MOTOR CONTROL ENABLE

A switch has been provided to allow the output bit of the software serial startup port to act as a "MOTOR-ON" bit for minifloppies. If switch S1 position 3 is "OFF", the serial port may be used, and the minifloppy motors will be on at all times. If position 3 is "ON", the "MOTOR ON" line is active and the motors may be turned "ON" by outputting a "0" to the control bit. The motors may be turned "OFF" by outputting a "1" to the control bit.

MODIFICATIONS FOR MINIFLOPPY DRIVES

If you have purchased a DISK 1 controller for 8 inch drives and wish to convert it to 5.25 inch operation, this modification can be performed at the factory for a nominal charge. Otherwise, the following section may serve as a guide for technically competent users to alter this board.

Several modifications need to be made to the DISK 1 board configured as an 8 inch controller prior using it with minifloppy drives. These changes will be outlined below:

(1) The components indicated in the parts list under ALTERNATE PARTS LIST FOR MINIFLOPPY DRIVES must be substituted for the parts that are currently on the board unless they have been substituted at the factory.

(2) The three traces on the solder side of the board pointed to by the three arrows (behind J11, J12, and J13) must be cut with a sharp knife. Three jumpers must then be installed to connect the pads labeled "5" and "C". In addition install J14.

(3) If your minifloppy does not drive the READY* line, install a jumper on J15 between "A" and "B". If your minifloppy does drive the READY* line, install a jumper between "A" and "C".

At this point, you should be ready to connect your minifloppy drive to the controller and verify its operation. Some alteration of the setting of R35 may be necessary for reliable double density operation. Remember that your 34 pin cable should be offset to the extreme right in connector J10 for proper operation.

DISK 1 CONNECTOR PINOUT WITH 8 AND 5.25 INCH DRIVES

DISK 1 J10 PIN	8 INCH DRIVE SIGNAL	DISK 1 J10 PIN	5 INCH DRIVE CABLE PIN	5 INCH DRIVE SIGNAL
2	LOW CURRENT	2	NC	
4	FAULT RESET	4	NC	
6	FAULT	6	NC	
8	NC	8	NC	
10	TWO SIDED	10	NC	
12	NC	12	NC	
14	SIDE SELECT	14	NC	
16	NC	16	NC	
18	HEAD LOAD	18	2	NC
20	INDEX (8")	20	4	NC
22	READY	22	6	DRIVE SELECT 4
24	INDEX (5")	24	8	INDEX
26	DRIVE SELECT 1	26	10	DRIVE SELECT 1
28	DRIVE SELECT 2	28	12	DRIVE SELECT 2
30	DRIVE SELECT 3	30	14	DRIVE SELECT 3
32	DS4/MOTOR ON	32	16	MOTOR ON
34	DIRECTION SELECT	34	18	DIRECTION SELECT
36	STEP	36	20	STEP
38	WRITE DATA	38	22	WRITE DATA
40	WRITE GATE	40	24	WRITE GATE
42	TRACK 00	42	26	TRACK 00
44	WRITE PROTECT	44	28	WRITE PROTECT
46	READ DATA	46	30	READ DATA
48	NC	48	32	SIDE SELECT
50	NC	50	34	READY

PINS 1-49 ODD ARE ALL GROUND RETURNS.

THEORY OF OPERATION

The DISK 1 board can be broken down into five subsections that will be discussed in detail in the following pages. These five subsections correspond to the five pages of the schematic, and include: (1) The Bus Interface and Boot Circuitry, (2) The DMA Counters and Address Drivers with the Serial Port, (3) The DMA Sequencing Logic and Priority Arbiter, (4) The Disk Read/Write Circuitry and Data Separator, (5) and The Floppy Disk Controller and Interface Circuitry. While reading this section it is suggested that the reader refer to the schematic and the data sheet on the controller chip.

SECTION (1): THE BUS INTERFACE AND BOOT CIRCUITRY

This section includes the logic for the S-100 bus interface to the floppy controller, the boot circuitry, the strobe generators, the wait state circuitry and the the data bus interface. Octal bus driver U41 buffers the address lines A0 thru A7 for on-board use by the strobe decoder (U11), the boot EPROM (U28), and the address decoder (U40). Octal comparator U40 uses DIP switch S2, address lines A2 thru A7, and sOUT and sINP* to decode a valid four port board select signal BDSEL*. BDSEL*, A0 and A1, OUTPUT* (which is sWO* buffered by U10A), and BUS STB (generated from pDBIN and pWR* by U23), are decoded by the 3-8 line decoder U11 to generate the strobes for the floppy disk controller, the DMA registers, and the serial port. Three of these strobes are multiplexed by U10B,U9 and U16 with BC*, WE, and pDBIN to generate the strobes for the floppy disk controller (RD* and WR*) and the DMA clock (DMA CLK) depending on whether a DMA bus cycle (BC) is occurring or not. During a DMA cycle, WE and pDBIN generate the control strobes for the controller chip, and BC* clocks the DMA counters. During non-DMA cycles, U11 generates these strobes.

The boot circuitry consisting of two SR latches (U7A,C) and U24a, is initialized by INIT* (buffered pRESET* from U10A). When U7A is enabled by switch S1-4, INIT* generates BOOT, which together with sMEMR generates PROM ENA* to enable the boot EPROM whenever a memory read occurs. The boot software is contained in EPROM U28. One of eight routines within the EPROM is selected by positions 1 and 2 of switch S2 and J17. The data from the EPROM enters the internal data bus (D0-D7) and is buffered onto the S-100 bus by U45. Since PHANTOM* is generated by U38 whenever BOOT is asserted and a DMA bus cycle (BC*) is not occurring, and since system memory boards should be set to become disabled when PHANTOM* is asserted, the host CPU will read the BOOT EPROM during a boot sequence. When the boot is through, U7A is reset by Q of U24a with D0=0 and SER OUT STB* and BOOT is released.

Octal bi-directional bus drivers U44 and U45 buffer data to and from the controller board with steering from U8 and U21. When either a DMA bus cycle (BC*) or an OUTPUT* (command write to the FDC) occurs, U44 is enabled, and the DMA transfer line (XFER) determines the direction of the data. When XFER is high, data is transferred out of the board and onto the DO lines. When XFER is low, data is transferred from the DO bus onto the board (command write). When either a BDSEL*, PROM ENA* or a BC* occurs, the WEN line is asserted, and if BUS STB and OUTPUT* are also asserted, U45 is enabled. When XFER* is high, the internal data is output onto the DI lines (PROM read, FDC read, or DMA write). When XFER* is low, the data goes from the DI lines to the FDC for a disk write operation.

A wait state is left pending in shift register U50 at all times, however it is gated onto the bus only when S1-1 is closed and WEN is high. After BUS STB is asserted, U50 is allowed to shift its data once on each rising edge of Φ* during a bus strobe (BUS STB*- pDBIN or pWR*). The first low to inputs d-h of U50 will terminate the wait state after it has been shifted to Qh. Five cycles are required to terminate an EPROM wait state and 2, 3 or 4 cycles to terminate either an I/O or DMA wait state. The wait states are enabled only when switch S1-1 is closed (ON), and WEN is asserted (EPROM read, I/O operation, or DMA cycle).

SECTION (2) DMA COUNTERS AND ADDRESS DRIVERS

The six DMA counters (U12,13,25,26,29, and 30) form a parallel loading 24 bit counter for address generation during the DMA transfer cycle. When XFER is not asserted (low), and DMA CLK rises (from DMA STB*, non-DMA cycle), the counters are loaded from the internal data bus (U13, U30) or from the previous stage of counter (U12, U29 or U25, U26). When XFER is asserted (a DMA cycle), DMA CLK is generated by BC*, and the counters are incremented for the next byte transferred.

The address buffers (U39, 42, and 43) drive the 24 bits of information from the DMA counters out onto the bus whenever BC* is asserted (during a DMA transfer). Since a full 24 bits of counter are present in this circuit, DMA transfers become independent of the 64K boundaries normally present in 8 bit processors. This allows 16 bit processors with up to 24 bits of direct addressing to be fully supported.

The software controlled serial channel is supported through two sections of op-amp U6, "D" type latch U24B, and one section of Tri-state driver U9. Data bit 7 is latched by U24B on the rising edge of SER OUT STB*, and the Q and Q* outputs drive U6 to convert the levels to the ±12V for RS-232 operation. The Q output also drives the MOTOR ENA line for motor control of minifloppy drives, and INIT* clears the latch on power-up. The RS-232 input levels are shifted to a TTL compatible level with the other section of U6 and this TTL level is gated onto the data bus by buffer U9 whenever a SER IN STB* occurs.

SECTION (3) THE DMA ARBITER AND SEQUENCER LOGIC

A DMA cycle is initiated when the floppy disk controller asserts the data request (DRQ) line, the pHOLD* line is not asserted, and the hold acknowledge line (HLDA) is not asserted. After a delay of up to 1.6 uSEC through U22, U7B, R36 and C19, U20A is preset and causes the assert priority line (APRIO) to be asserted along with the pHOLD* line. The priority is asserted onto the DMA0* - DMA3* lines as generated and checked by the logic of U35, 36, 37, and switch S1 positions 5 thru 8. The highest priority line DMA3* is asserted first depending on the switch setting, and if the DMA3* line agrees with the asserted priority (no one with higher priority is asserting the line), then the next lowest priority line (DMA2*) is asserted and checked as in the first case. This process repeats until either a priority mismatch occurs and the sequence is held or the IMHI line is asserted indicating that this board is the highest priority. U20A and the APRIO sequence is held while the processor acknowledges the hold by asserting HLDA line and the DMA cycle occurs. The sequence is terminated at the end of the DMA cycle with BC* and STB INH (U23) or when a reset occurs (INIT* -

U21). As soon as HLDA and IMHI are asserted (U21), the transfer state is entered (XFER or XFER*) in U20B on the next rising edge of Φ*. This state causes the S-100 control strobes to be driven onto the bus in an IDLE state condition with all addresses, status, DO buffers, and control strobes disabled by buffer U27 and U38.

The transfer state is controlled by the sequencing logic of quad "D" latch U4, NOR gates of U3, and U21. Each new state is initiated on the meeting of several conditions and the rising edge of Φ. When XFER* is asserted, and STB INH is not, the bus cycle (BC, BC*) is initiated, the S-100 status lines (U46) and the DMA address lines (U39, 42, and 43) are driven onto the bus. In addition, BC* is fed back to the FDC as an acknowledge signal (DACK), allowing the FDC to enter the data transfer mode. In the next state after BC is asserted, the STB ENA and STB ENA* lines are asserted. This causes the strobes to be switched from the IDLE state to either a read or write state depending on the WE* signal from the FDC. The strobes pDBIN* and pWR* are generated by U31A, pSYNC is generated from BC* and STB INH in U3, and pSTVAL* is generated from pSYNC and Φ* in U23. The READY state is constantly being generated by pRDY* and is updated every cycle of Φ. When either READY is asserted or STB INH is asserted (U3), and STB ENA* is asserted, the strobe inhibit state (STB INH or STB INH*) is entered. This causes the strobes to be returned to the IDLE state as long as a wait state is not requested. If a wait state has been requested, the strobes will not be changed until the READY state occurs. STB INH also causes the APRIO line to be reset, and on the next rising edge of Φ, the bus cycle (BC) will be terminated causing the status and address lines to be removed from the bus. As soon as pHOLD* is released and HLDA stops being asserted, the next edge of Φ* will cause the termination of the transfer state (XFER) and the removal of the S-100 strobes and the IDLE state on the bus.

SECTION (4) DISK READ/WRITE CIRCUITRY AND DATA SEPARATOR

The master clock for the FDC is generated by a crystal oscillator consisting of X1, L1, and U49. This clock is fed to the FDC and is divided by U47A to yield the WRITE CLK through U48 at the proper frequency determined by the signal MFM from the FDC. WRITE CLK is converted to a 250 nSEC. pulse by U34B and is fed to the FDC. The two write precompensation signals PS0 and PS1 and the low current signal LC alter the value of the timing resistor of U34A by diode current gating, and allow a varying delay in the disk write data. WRITE DATA from the FDC is fed to U34A, and when enabled by WE and time varied by the precompensation logic, and variable length COMP WRITE* signal is generated. This pulse will have typical output length of 1.1, 1.3, and 1.5 uSEC. for an alteration from NORMAL,(200 nSEC), EARLY, and LATE precompensation. The COMP WRITE* signal is fed to U19B for conversion to a 250 nSEC. pulse signal (WRITE PULSE*) for feeding to the disk.

A phase lock loop is formed by a charge pump consisting of U18, U17, diodes D1-3, and resistors R13 and R18, an error amplifier and integrator consisting U6, C5-6, and R6-8 and R35, and a voltage controlled oscillator (VCO) consisting of U5, R5, R9, and C7-8. The VCO is adjusted by R35 to oscillate at approximately 2.0 MHz when enabled by the signal VCO ENA from the FDC. The output VCO is divided by U47B, and the divided outputs are selected by multiplexor U48 for the proper frequency depending on the current mode (MFM) to generate the signals WINDOW and CELL CLK. Read data (READ DATA*) from the disk is fed to U19A for conversion to a 500 nSEC. or 1000 nSEC. pulse depending on

the state of the FM line (500 nSEC. in MFM, 1000 nSEC. in FM). DATA and DATA* along with CELL CLK and CELL CLK* are fed to the charge pump which determines the relative position of the CELL CLOCK with respect to the DATA and generates a voltage at pin 5. This voltage represents the error in the positioning of the DATA signal with respect to the edge of CELL CLK. This error voltage is integrated and fed to the error amplifier which adjusts the frequency of the voltage controlled oscillator to bring the DATA pulse back in line with the CELL CLK and correspondingly adjust the WINDOW signal (which defines the data cell for the FDC). The DATA* and CELL CLK* signals set and clear U7D, and generate a constant 250 nSEC. read pulse for the FDC (RD DATA).

SECTION (5) THE FLOPPY DISK CONTROLLER AND INTERFACE CIRCUITRY

The floppy disk controller chip (U14) provides for the bridging of the previous four sections to the disk interface circuitry and the disk drives. The FDC is reset by INIT* inverted by U33, and is connected to the internal data bus through D0-D7. The FDC interface to the internal bus is controlled by A0, WR*, RD*, and XFER, which leave the FDC enabled at all times except during a transfer state, and steer the data with WR*, and RD*. DRQ initiates the DMA cycle, and BC* initiates the FDC data transfer. CLK and WCK are pulled up by R19 and R24 to achieve a greater voltage swing for the internal clocking logic of the FDC. WINDOW describes the data cell time of the RD DATA pulse from the data separator, and VCO ENA enables the voltage controlled oscillator and MFM describes whether the FDC is expecting single or double density data. WRITE DATA feeds the write precompensation circuitry with PS0 and PS1 describing the amount of precomp needed for the particular data pattern. The WE and WE* (inverted by U33) lines control the enabling of the write circuitry and are used to determine the direction of the data transfers to and from the FDC. The INT output from the FDC is the general interrupt signal that is fed to buffer U17 for driving the vectored interrupt lines (VI0-VI7) and pINT if jumpered. INT is also buffered by Tri-state buffer U9, which gates the interrupt status onto D7 when a STATUS STB* occurs.

The Drive interface signals are decoded and buffered as described below. The HD line is inverted by U33 and buffered by U32 to drive the SIDE SELECT* line to determine which head of a double sided drive to use. HDL is inverted and used to drive the HEAD LOAD* line of the drive. The RDY signal is controlled by the inverted (U33) READY* line and the IDX signal is controlled by the inverted (U33) INDEX* line from the drive. The unit select lines US0 and US1 are fed to decoder U31B to decode the one of four drives possible with this controller, and these four signals are buffered by four sections of U32 to drive the DR SEL0* through DR SEL3* lines of the drives. The four signals WP/TS, FLT/TR0, FR/STP, and LCT/DIR each have two functions which are determined by the RW/SK line (read-write/seek). During a read or write operation, the drive signals WRITE PROT*, and FAULT* are inverted and fed to the FDC lines WP/ and FLT/ and the FDC lines FR/ and LCT/ are inverted (U15) and buffered (U16) to drive the FAULT RESET* and LOW CURRENT* drive lines. During a seek operation, the drive signals TWO SIDED*, and TRACK 0* are inverted and fed to the FDC lines /TS and /TR0 and the FDC lines /STP and /DIR are inverted (U15) and buffered (U16) to drive the STEP* and DIRECTION* drive lines. These changes are controlled by the RW/SK line which determines which half of U15 will be enabled. The WRITE PULSE* line is buffered (U16) to drive the WRITE DATA* line and the READ DATA* line is terminated and fed directly to U19A. The MOTOR ENA signal is run through disable switch S1-3 and inverted to feed the ME line which becomes

MOTOR ON* on the minifloppy drives. All lines coming from the floppy drives are terminated by 150 ohm resistors, and jumpers J11-15 are used to select either 5.25" or 8" drives.

Regulators U1 and U2 provide the regulated +5 volts for the board, and U51 and U50 provide the regulated ±12 volts for the op-amp. Diodes D10 and D11 insure that U50 and U51 do not shut each other down on power-up due to the op-amp not being ground referenced.

TEST ROUTINES FOR THE DISK 1 CONTROLLER

The following program contains primitive routines for (1) Loading the DMA address counters, (2) Specifying the drive characteristics to the controller, (3) Recalibrating the drive (seeking track 00), (4) Seeking a particular track on the drive, (5) Reading sectors from the drive in either single density or 1024 byte double density, and (6) Writing sectors to the drive in either single density or 1024 byte double density. The user should be familiar with the data sheet for the floppy disk controller since these routines are designed to give the user a greater understanding of the operation of the 8272/765A. These routines are by no means the basis for a disk operating system, and are designed for experimentation only.

This program should be treated as several callable subroutines using the jump table at the beginning for reference.

NOTE: This program was written to assemble under an ACT assembler and may require slight modification to assemble under CP/M-80 ASM.

```
        TITLE 'TEST ROUTINES FOR COMPUPRO DISK CONTROLLER.'

;DTEST - Test Routines for CompuPro Disk Controller.
;                  +----------------------+
;
;                        DISK TEST
;
;                  +----------------------+
;                          COMPUPRO
;                       Oakland Airport
;                  Oakland, California  94611
;
;             Copyright 1981, CompuPro Corporation.
;
;   This product is a program product of CompuPro and is supplied
;   for use with the CompuPro IEEE 696 Floppy Disk Controller.
;
;       Version number:  1.1B  Version date:  April 13, 1981

BEGIN       EQU     0100h

;           Assembly Constants
FDPORT      EQU     0C0H            ;Base port address for Controller
FDCS        EQU     FDPORT          ;Status register
FDCD        EQU     FDPORT+1        ;Data register
```

```
FDMA        EQU     FDPORT+2        ;Dma address (when write)
INTS        EQU     FDPORT+2        ;Status Register (when read)
SER         EQU     FDPORT+3        ;Serial port

;           Controller function definitions
;           Specify (00) command
F.RTK       =       02              ;Read track
F.SPEC      =       03              ;Specify
F.DSTS      =       04              ;Drive status
F.RDAT      =       06              ;Read sector FM
F.DRDT      =       46H             ;Read sector MFM
F.WRAT      =       05              ;Write sector FM
F.WRDT      =       45H             ;Write sector MFM
F.RECA      =       07              ;recalibrate
F.RSTS      =       08              ;Read status
F.SEEK      =       0Fh             ;Seek

SRT         =       16-8            ;= Shugart 800s (8 ms)
;                   16-3            ;= Shugart 850s (3 ms)
;                   16-3            ;= Remex (3 ms)
HUT:        =       240/16          ;Head unload = 240 ms
HLT:        =       (35+1)/2        ;Head load = 35 ms
ND:         =       00              ;Set DMA mode

;           JUMP TABLE FOR ROUTINES
;           These routines are callable subroutines
;           Some require parameters passed in the "A" and "C" registers

            ORG     BEGIN
START:      JMP     DDMA            ;load DMA address
            JMP     DSPEC           ;specify drive stat
            JMP     RCAL            ;recalibrate (track 00)
            JMP     DSEEK           ;seek a track
            JMP     READS           ;read sector (FM)
            JMP     DREADS          ;read sector  (MFM)
            JMP     WRS             ;write sector (FM)
            JMP     DWRS            ;write sector (MFM)

;           DMA address load routine using 16 bit value in
;           HL register for the 24 bit DMA value
DDMA:       MVI     A,0             ;extended address
            OUT     FDMA            ;output
            MOV     A,H             ;high byte
            OUT     FDMA            ;output
            MOV     A,L             ;low byte
            OUT     FDMA            ;output
            RET

;           Drive Specify Command
DSPEC:      MVI     B,LSPEC         ;3 byte command
            LXI     D,SPEC          ;point to command bytes
SPEC1:      IN      FDCS            ;get status
            ANI     0C0H
            CPI     80H
            JNZ     SPEC1           ;if no master ready bit
```

```
            LDAX    D               ;load command byte
            OUT     FDCD            ;to controller
            INX     D               ;next byte
            DCR     B               ;dec. counter
            JNZ     SPEC1           ;if more bytes
            RET

;           Recalibrate drive (seek track 0)
RCAL:       MVI     B,LRECAL        ;2 byte command
            LXI     D,RECAL         ;point to command bytes
RCAL1:      IN      FDCS            ;get status
            ANI     0C0H
            CPI     80H
            JNZ     RCAL1           ;if no master ready bit
            LDAX    D               ;load command byte
            OUT     FDCD            ;to controller
            INX     D
            DCR     B
            JNZ     RCAL1           ;if more bytes
            RET

;           Seek a Track with cylinder number in "A"
DSEEK:      MVI     B,LSEEK         ;3 byte command
            LXI     D,SEEK          ;point to command bytes
            STA     CYLD            ;store cylinder #
SEEK1:      IN      FDCS            ;check status
            ANI     0C0H
            CPI     80H
            JNZ     SEEK1           ;if not ready
            LDAX    D               ;load command byte
            OUT     FDCD            ;to controller
            INX     D
            DCR     B
            JNZ     SEEK1           ;if more bytes
            RET

;           FM Sector read command with sector in "A"
;           and cylinder in "C"
READS:      MVI     B,LREAD         ;9 byte command
            LXI     D,READ          ;point to command bytes
            STA     RSEC            ;store sector number
            MOV     A,C
            STA     RSCYL           ;store cylinder number
READ1:      IN      FDCS            ;check status
            OR      A
            JP      READ1           ;if no master ready bit
            LDAX    D               ;load command byte
            OUT     FDCD            ;to controller
            INX     D
            DCR     B
            JNZ     READ1           ;if more bytes

READ2:      IN      INTS            ;check interrupt status
            ORA     A               ;for read complete
            JP      READ2           ;If not complete
```

```
READ3:      IN      FDCS            ;in status
            ORA     A
            JP      READ3           ;if not ready
            IN      FDCD            ;read result byte ST0
            SUI     40h             ;strip
            MOV     L,A             ;save
READ4:      IN      FDCS            ;in status
            ORA     A
            JP      READ4           ;if not ready
            IN      FDCD            ;read result byte ST1
            SUI     80h             ;strip
            MOV     H,A             ;save

            MVI     B,7-2           ;5 more bytes
READ5:      IN      FDCS            ;in status
            OR      A
            JP      READ5           ;if not ready
            IN      FDCD            ;read result byte
            DEC     B
            JNZ     READ5           ;wait until all done
            MOV     A,L             ;check results
            ORA     H
            RZ                      ;return no error
            STC                     ;set carry
            RET                     ;return from error

;           MFM Sector read command with sector in "A"
;           and cylinder in "C"
DREADS:     MVI     B,DLREAD        ;9 byte command
            LXI     D,DREAD         ;point to command bytes
            STA     DRSEC           ;store sector number
            MOV     A,C
            STA     DCYL            ;store cylinder number
DREAD1:     IN      FDCS            ;check status
            OR      A
            JP      DREAD1          ;if no master ready bit
            LDAX    D               ;load command byte
            OUT     FDCD            ;to controller
            INX     D
            DCR     B
            JNZ     DREAD1          ;if more bytes
            JMP     READ2

;           FM Sector write command with sector in "A"
;           cylinder number in "C"
WRS:        MVI     B,LWR           ;9 byte command
            LXI     D,WR            ;point to command bytes
            STA     WSEC            ;store sector number
            MOV     A,C
            STA     WSCYL           ;store cylinder number
WR1:        IN      FDCS            ;check status
            OR      A
            JP      WR1             ;if no master ready bit
            LDAX    D               ;load command byte
```

```
                OUT     FDCD            ;to controller
                INX     D
                DCR     B
                JNZ     WR1             ;if more bytes
                JMP     READ1

;               MFM Sector write command with sector in "A"
;               and cylinder in "C"
DWRS:           MVI     B,DLWR          ;9 byte command
                LXI     D,DWR           ;point to command bytes
                STA     DWSEC           ;store sector number
                MOV     A,C
                STA     DWRCYL          ;store cylinder number
DWR1:           IN      FDCS            ;check status
                OR      A
                JP      DWR1            ;if no master ready bit
                LDAX    D               ;load command byte
                OUT     FDCD            ;to controller
                INX     D
                DCR     B
                JNZ     DWR1            ;if more bytes
                JMP     READ2

;               Function data for controller
SPEC            DB      F.SPEC          ;specify command
                VFD     4@SRT,4@HUT
                VFD     7@HLT,1@ND
LSPEC           =       *-SPEC

RECAL           DB      F.RECA,0        ;recalibrate command
LRECAL          =       *-RECAL

SEEK            DB      F.SEEK          ;seek command
                DB      0
CYLD            DB      0
LSEEK           =       *-SEEK

READ:           DB      F.RDAT          ;read command (FM)
                DB      0               ;hds,ds1,ds0
RSCYL           DB      0               ;C = cylinder info
                DB      0               ;Head
RSEC:           DB      1               ;Record (first sector)
                DB      0               ;N 128 BYTE SECTOR
                DB      25              ;EOT (last sectors)
                DB      7               ;GPL
                DB      128             ;DTL
LREAD           =       *-READ

DREAD:          DB      F.DRDT          ;read command (MFM)
                DB      0               ;hds,ds1,ds0
DCYL            DB      2               ;C = cylinder info
                DB      0               ;Head
DRSEC:          DB      1               ;Record (first sector)
                DB      3               ;N 1024 BYTE SECTOR
                DB      7               ;EOT (last sectors)
```

```
                DB      35H             ;GPL
                DB      0               ;DTL
DLREAD          =       *-DREAD

WR:             DB      F.WRAT          ;write command (FM)
                DB      0               ;hds,dsl,ds0
WSCYL           DB      0               ;C = cylinder info
                DB      0               ;Head
WSEC:           DB      1               ;Record (first sector)
                DB      0               ;N 128 BYTE SECTOR
                DB      25              ;EOT (last sectors)
                DB      7               ;GPL
                DB      128             ;DTL
LWR             =       *-WR

DWR             DB      F.WRDT          ;write command (MFM)
                DB      0               ;hds,dsl,ds0
DWRCYL          DB      2               ;C = cylinder info
                DB      0               ;Head
DWSEC:          DB      1               ;Record (first sector)
                DB      3               ;N 1024 BYTE SECTOR
                DB      7               ;EOT (last sectors)
                DB      35H             ;GPL
                DB      0               ;DTL
DLWR            =       *-DWR
                END
```

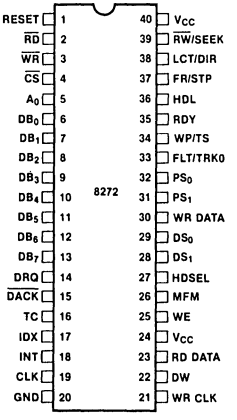
PRELIMINARY

8272
SINGLE/DOUBLE DENSITY FLOPPY DISK CONTROLLER

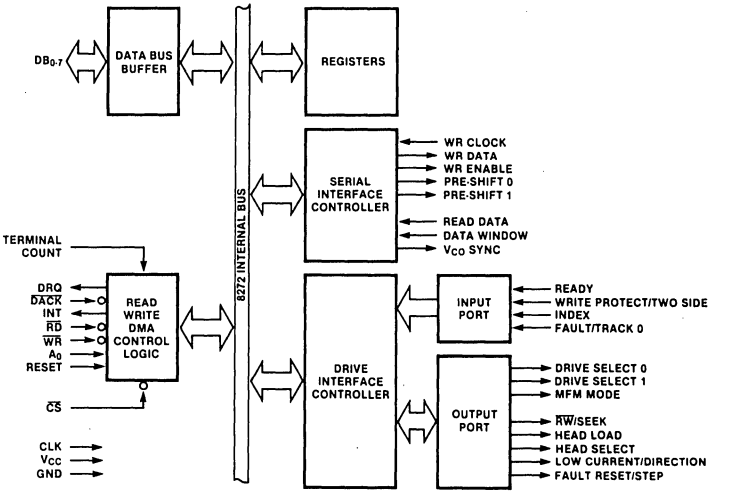
- **IBM Compatible in Both Single and Double Density Recording Formats**
- **Programmable Data Record Lengths: 128, 256, 512, or 1024 Bytes/Sector**
- **Multi-Sector and Multi-Track Transfer Capability**
- **Drive Up to 4 Floppy Disks**
- **Data Scan Capability — Will Scan a Single Sector or an Entire Cylinder's Worth of Data Fields, Comparing on a Byte by Byte Basis, Data in the Processor's Memory with Data Read from the Diskette**
- **Data Transfers in DMA or Non-DMA Mode**
- **Parallel Seek Operations on Up to Four Drives**
- **Compatible with Most Microprocessors Including 8080A, 8085A, 8086 and 8088**
- **Single-Phase 8 MHz Clock**
- **Single +5 Volt Power Supply**
- **Available in 40-Pin Plastic Dual-in-Line Package**

The 8272 is an LSI Floppy Disk Controller (FDC) Chip, which contains the circuitry and control functions for interfacing a processor to 4 Floppy Disk Drives. It is capable of supporting either IBM 3740 single density format (FM), or IBM System 34 Double Density format (MFM) including double sided recording. The 8272 provides control signals which simplify the design of an external phase locked loop, and write precompensation circuitry. The FDC simplifies and handles most of the burdens associated with implementing a Floppy Disk Drive Interface.

PIN CONFIGURATION

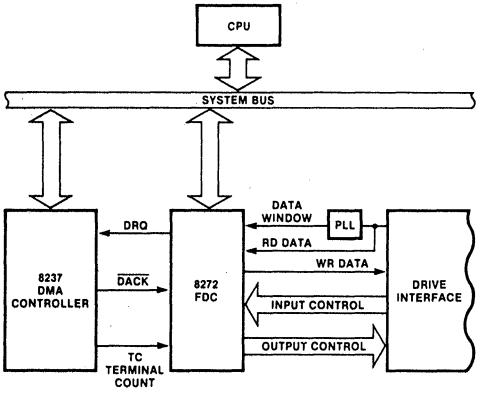


8272 INTERNAL BLOCK DIAGRAM



JANUARY 1980

8272 SYSTEM BLOCK DIAGRAM



DESCRIPTION

Hand-shaking signals are provided in the 8272 which make DMA operation easy to incorporate with the aid of an external DMA Controller chip, such as the 8237. The FDC will operate in either DMA or Non-DMA mode. In the Non-DMA mode, the FDC generates interrupts to the processor for every transfer of a data byte between the CPU and the 8272. In the DMA mode, the processor need only load a command into the FDC and all data transfers occur under control of the 8272 and DMA controller.

There are 15 separate commands which the 8272 will execute. Each of these commands require multiple 8-bit bytes to fully specify the operation which the processor wishes the FDC to perform. The following commands are available.

- Read Data
- Read ID
- Read Deleted Data
- Read a Track
- Scan Equal
- Scan High or Equal
- Scan Low or Equal
- Specify
- Write Data
- Format a Track
- Write Deleted Data
- Seek
- Recalibrate (Restore to Track 0)
- Sense Interrupt Status
- Sense Drive Status

FEATURES

Address mark detection circuitry is internal to the FDC which simplifies the phase locked loop and read electronics. The track stepping rate, head load time, and head unload time may be programmed by the user. The 8272 offers many additional features such as multiple sector transfers in both read and write modes with a single command, and full IBM compatibility in both single (FM) and double density (MFM) modes.

8272 REGISTERS — CPU INTERFACE

The 8272 contains two registers which may be accessed by the main system processor; a Status Register and a Data Register. The 8-bit Main Status Register contains the status information of the FDC, and may be accessed at any time. The 8-bit Data Register (actually consists of several registers in a stack with only one register presented to the data bus at a time), stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to program or obtain the results after execution of a command. The Status Register may only be read and is used to facilitate the transfer of data between the processor and 8272.

The relationship between the Status/Data registers and the signals $\overline{RD}$, $\overline{WR}$, and A_0 is shown below.

A_0	$\overline{RD}$	$\overline{WR}$	FUNCTION
0	0	1	Read Main Status Register
0	1	0	Illegal
0	0	0	Illegal
1	0	0	Illegal
1	0	1	Read from Data Register
1	1	0	Write into Data Register

The bits in the Main Status Register are defined as follows:

BIT NUMBER	NAME	SYMBOL	DESCRIPTION
DB_0	FDD 0 Busy	D_0B	FDD number 0 is in the Seek mode.
DB_1	FDD 1 Busy	D_1B	FDD number 1 is in the Seek mode.
DB_2	FDD 2 Busy	D_2B	FDD number 2 is in the Seek mode.
DB_3	FDD 3 Busy	D_3B	FDD number 3 is in the Seek mode.
DB_4	FDC Busy	CB	A read or write command is in process.
DB_5	Non-DMA mode	NDM	The FDC is in the non-DMA mode. This bit is set only during the execution phase in non-DMA mode. Transition to "0" state indicates execution phase has ended.
DB_6	Data Input/Output	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO = "1" then transfer is from Data Register to the Processor. If DIO = "0", then transfer is from the Processor to Data Register.
DB_7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor.

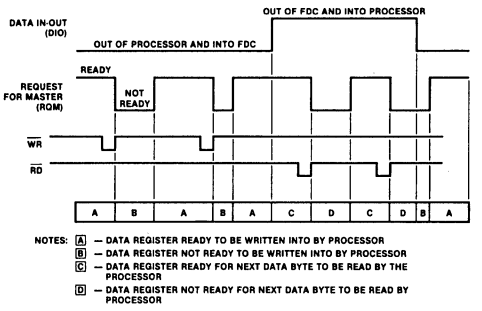
PIN DESCRIPTION

PIN NO.	PIN SYMBOL	I/O	CONNECTION TO	DESCRIPTION
1	RST	I	µP	Reset: Places FDC in idle state. Resets output lines to FDD to "0" (low)
2	$\overline{RD}$	I[1]	µP	Read: Control signal for transfer of data from FDC to Data Bus, when "0" (low)
3	$\overline{WR}$	I[1]	µP	Write: Control signal for transfer of data to FDC via Data Bus, when "0" (low)
4	$\overline{CS}$	I	µP	Chip Select: IC selected when "0" (low), allowing $\overline{RD}$ and $\overline{WR}$ to be enabled
5	A_0	I[1]	µP	Data/Status Reg Select: Selects Data Reg (A_0 = 1) or Status Reg (A_0 = 0) content be sent to Data Bus
6-13	DB_0-DB_7	I/O[1]	µP	Data Bus: Bidirectional 8-Bit Data Bus
14	DRQ	O	DMA	Data DMA Request: DMA Request is being made by FDC when DRQ "1"
15	$\overline{DACK}$	I	DMA	DMA Acknowledge: DMA cycle is active when "0" (low) and Controller is performing DMA transfer
16	TC	I	DMA	Terminal Count: Indicates the termination of a DMA transfer when "1" (high)
17	IDX	I	FDD	Index: Indicates the beginning of a disk track
18	INT	O	µP	Interrupt: Interrupt Request Generated by FDC
19	CLK	I		Clock: Single Phase 8 MHz Squarewave Clock
20	GND			Ground: D.C. Power Return

Note 1: Disabled when $\overline{CS}$ = 1.

PIN NO.	PIN SYMBOL	I/O	CONNECTION TO	DESCRIPTION
40	V_{CC}			D.C. POWER +5V
39	$\overline{RW}$/SEEK	O	FDD	Read Write/SEEK: When "1" (high) Seek mode selected and when "0" (low) Read/Write mode selected
38	LCT/DIR	O	FDD	Low Current/Direction: Lowers Write current on inner tracks in Read/Write mode, determines direction head will step in Seek mode
37	FR/STP	O	FDD	Fault Reset/Step: Resets fault FF in FDD in Read/Write mode, provides step pulses to move head to another cylinder in Seek mode
36	HDL	O	FDD	Head Load: Command which causes read/write head in FDD to contact diskette
35	RDY	I	FDD	Ready: Indicates FDD is ready to send or receive data
34	WP/TS	I	FDD	Write Protect/Two-Side: Senses Write Protect status in Read/Write mode, and Two Side Media in Seek mode
33	FLT/TRK0	I	FDD	Fault/Track 0: Senses FDD fault condition in Read/Write mode and Track 0 condition in Seek mode.
31,32	PS_1,PS_0	O	FDD	Precompensation (pre-shift): Write precompensation status during MFM mode. Determines early, late, and normal times.
30	WR DATA	O	FDD	Write Data: Serial clock and data bits to FDD
28,29	DS_1,DS_0	O	FDD	Drive Select: Selects FDD unit
27	HDSEL	O	FDD	Head Select: Head 1 selected when "1" (high) Head 0 selected when "0" (low)
26	MFM	O	PLL	MFM Mode: MFM mode when "1", FM mode when "0"
25	WE	O	FDD	Write Enable: Enables write data into FDD
24	VCO	O	PLL	VCO Sync: Inhibits VCO in PLL when "0" (low), enables VCO when "1"
23	RD DATA	I	FDD	Read Data: Read data from FDD, containing clock and data bits
22	DW	I	PLL	Data Window: Generated by PLL, and used to sample data from FDD
21	WR CLK	I		Write Clock: Write data rate to FDD FM = 500 kHz, MFM = 1 MHz, with a pulse width of 250 ns for both FM and MFM

The DIO and ROM bits in the Status Register indicate when Data is ready and in which direction data will be transferred on the Data Bus.



STATUS REGISTER TIMING

The 8272 is capable of executing 15 different commands. Each command is initiated by a multi-byte transfer from the processor, and the result after execution of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the 8272 and the processor, it is convenient to consider each command as consisting of three phases:

Command Phase: The FDC receives all information required to perform a particular operation from the processor.

Execution Phase: The FDC performs the operation it was instructed to do.

Result Phase: After completion of the operation, status and other housekeeping information are made available to the processor.

During Command or Result Phases the Main Status Register (described earlier) must be read by the processor before each byte of information is written into or read from the Data Register. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the 8272. Many of the commands require multiple bytes, and as a result the Main Status Register must be read prior to each byte transfer to the 8272. On the other hand, during the Result Phase, D6 and D7 in the Main Status Register must both be 1's (D6 = 1 and D7 = 1) before reading each byte from the Data Register. Note, this reading of the Main Status Register before each byte transfer to the 8272 is required in only the Command and Result Phases, and NOT during the Execution Phase.

During the Execution Phase, the Main Status Register need not be read. If the 8272 is in the Non-DMA Mode, then the receipt of each data byte (if 8272 is reading data from FDD) is indicated by an Interrupt signal on pin 18 (INT = 1). The generation of a Read signal ($\overline{RD} = 0$) will reset the Interrupt as well as output the Data onto the Data Bus. For example, if the processor cannot handle Interrupts fast enough (every 13 μs for MFM mode) then it may poll the Main Status Register and then bit D7 (RQM) functions just like the Interrupt signal. If a Write Command is in process then the $\overline{WR}$ signal performs the reset to the Interrupt signal.

If the 8272 is in the DMA Mode, no Interrupts are generated during the Execution Phase. The 8272 generates DRQ's (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both a $\overline{DACK} = 0$ (DMA Acknowledge) and a $\overline{RD} = 0$ (Read signal). When the DMA Acknowledge signal goes low ($\overline{DACK} = 0$) then the DMA Request is reset (DRQ = 0). If a Write Command has been programmed then a $\overline{WR}$ signal will appear instead of $\overline{RD}$. After the Execution Phase has been completed (Terminal Count has occurred) then an Interrupt will occur (INT = 1). This signifies the beginning of the Result Phase. When the first byte of data is read during the Result Phase, the Interrupt is automatically reset (INT = 0).

It is important to note that during the Result Phase all bytes shown in the Command Table must be read. The Read Data Command, for example, has seven bytes of data in the Result Phase. All seven bytes must be read in order to successfully complete the Read Data Command. The 8272 will not accept a new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result Phase.

The 8272 contains five Status Registers. The Main Status Register mentioned above may be read by the processor at any time. The other four Status Registers (ST0, ST1, ST2, and ST3) are only available during the Result Phase, and may be read only after successfully completing a command. The particular command which has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the 8272 to form the Command Phase, and are read out of the 8272 in the Result Phase, must occur in the order shown in the Command Table. That is, the Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result Phases are allowed. After the last byte of data in the Command Phase is sent to the 8272 the Execution Phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result Phase, the command is automatically ended and the 8272 is ready for a new command. A command may be aborted by simply sending a Terminal Count signal to pin 16 (TC = 1). This is a convenient means of ensuring that the processor may always get the 8272's attention even if the disk system hangs up in an abnormal manner.

POLLING FEATURE OF THE 8272

After the Specify command has been sent to the 8272, the Drive Select Lines DS0 and DS1 will automatically go into a polling mode. In between commands (and between step pulses in the SEEK command) the 8272 polls all four FDDs looking for a change in the Ready line from any of the drives. If the Ready line changes state (usually due to a door opening or closing) then the 8272 will generate an interrupt. When Status Register 0 (ST0) is read (after Sense Interrupt Status is issued), Not Ready (NR) will be indicated. The polling of the Ready line by the 8272 occurs continuously between instructions, thus notifying the processor which drives are on or off line.

TABLE 1. 8272 COMMAND SET

PHASE	R/W	DATA BUS D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	REMARKS
READ DATA			
Command	W	MT MFM SK 0 0 1 1 0	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
Execution			Data transfer between the FDD and main-system
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after command execution
	R	H	
	R	R	
	R	N	
READ DELETED DATA			
Command	W	MT MFM SK 0 1 1 0 0	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EC 1	
	W	GPL	
	W	DTL	
Execution			Data transfer between the FDD and main-system
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after Command execution
	R	H	
	R	R	
	R	N	
WRITE DATA			
Command	W	MT MFM 0 0 0 1 0 1	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
Execution			Data transfer between the main-system and FDD
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after Command execution
	R	H	
	R	R	
	R	N	
WRITE DELETED DATA			
Command	W	MT MFM 0 0 1 0 0 1	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
Execution			Data transfer between the FDD and main-system
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after Command execution
	R	H	
	R	R	
	R	N	

PHASE	R/W	DATA BUS D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	REMARKS
READ A TRACK			
Command	W	0 MFM SK 0 0 0 1 0	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
Execution			Data transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EOT
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after Command execution
	R	H	
	R	R	
	R	N	
READ ID			
Command	W	0 MFM 0 0 1 0 1 0	Commands
	W	0 0 0 0 0 HDS DS1 DS0	
Execution			The first correct ID information on the Cylinder is stored in Data Register
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information during Execution Phase
	R	H	
	R	R	
	R	N	
FORMAT A TRACK			
Command	W	0 MFM 0 0 1 1 0 1	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	N	Bytes/Sector
	W	SC	Sectors/Track
	W	GPL	Gap 3
	W	D	Filter Byte
Execution			FDC formats an entire cylinder
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	In this case, the ID information has no meaning
	R	H	
	R	R	
	R	N	
SCAN EQUAL			
Command	W	MT MFM SK 1 0 0 0 1	Command Codes
	W	0 0 0 0 0 HDS DS1 DS0	
	W	C	Sector ID information prior to Command execution
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	STP	
Execution			Data compared between the FDD and main-system
Result	R	ST 0	Status information after Command execution
	R	ST 1	
	R	ST 2	
	R	C	Sector ID information after Command execution
	R	H	
	R	R	
	R	N	

Note: 1. Symbols used in this table are described at the end of this section.

2. $A_0 = 1$ for all operations.

3. X = Don't care, usually made to equal binary 0.

TABLE 1. COMMAND SET (Continued)

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					STP				
Execution										Data compared between the FDD and main-system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID information after Command execution
	R					H				
	R					R				
	R					N				
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					C				Sector ID information prior Command execution
	W					H				
	W					R				
	W					N				
	W					EOT				
	W					GPL				
	W					STP				
Execution										Data compared between the FDD and main-system
Result	R					ST 0				Status information after Command execution
	R					ST 1				
	R					ST 2				
	R					C				Sector ID information after Command execution
	R					H				
	R					R				
	R					N				

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head retracted to Track 0
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R					ST 0				Status information at the end of each seek operation about the FDC
	R					PCN				
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT →							HUT	
	W	HLT →							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R					ST 3				Status information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W					NCN				
Execution										Head is positioned over proper Cylinder on Diskette
INVALID										
Command	W					Invalid Codes				Invalid Command Codes (NoOp — FDC goes into Standby State)
Result	R					ST 0				ST 0 = 80 (16)

TABLE 2. COMMAND MNEMONICS

SYMBOL	NAME	DESCRIPTION
A_0	Address Line 0	A_0 controls selection of Main Status Register (A_0 = 0) or Data Register (A_0 = 1).
C	Cylinder Number	C stands for the current selected Cylinder track number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a Sector.
D_7-D_0	Data Bus	8-bit Data Bus where D_7 is the most significant bit, and D_0 is the least significant bit.
DS0, DS1	Drive Select	DS stands for a selected drive number 0 or 1.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the Sector.
EOT	End of Track	EOT stands for the final Sector number of a Cylinder.
GPL	Gap Length	GPL stands for the length of Gap 3 (spacing between Sectors excluding VCO Sync Field).
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HDS	Head Select	HDS stands for a selected head number 0 or 1 (H = HDS in all command words).
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MFM	FM or MFM Mode	If MF is low, FM mode is selected and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be performed (a cylinder under both HD0 and HD1 will be read or written).
N	Number	N stands for the number of data bytes written in a Sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R stands for the Sector number, which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (1 to 16 ms in 1 ms increments). Stepping Rate applies to all drives (F = 1 ms, E = 2 ms, etc.).
ST 0 ST 1 ST 2 ST 3	Status 0 Status 1 Status 2 Status 3	ST 0-3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A_0 = 0). ST 0-3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2, then alternate sectors are read and compared.

COMMAND DESCRIPTIONS

During the Command Phase, the Main Status Register must be polled by the CPU before each byte is written into the Data Register. The DIO (DB6) and RQM (DB7) bits in the Main Status Register must be in the "0" and "1" states respectively, before each byte of the command may be written into the 8272. The beginning of the execution phase for any of these commands will cause DIO and RQM to switch to "1" and "0" states respectively.

READ DATA

A set of nine (9) byte words are required to place the FDC into the Read Data Mode. After the Read Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC outputs data (from the data field) byte-by-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous read function is called a "Multi-Sector Read Operation." The Read Data Command may be terminated by the receipt of a Terminal Count signal. Upon receipt of this signal, the FDC stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then at the end of the sector terminate the Read Data Command.

The amount of data which can be handled with a single command to the FDC depends upon MT (multi-track), MFM (MFM/FM), and N (Number of Bytes/Sector). Table 3 below shows the Transfer Capacity.

TABLE 3. TRANSFER CAPACITY

Multi-Track MT	MFM/FM MFM	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128) (26) = 3,328	26 at Side 0
0	1	01	(256) (26) = 8,656	or 26 at Side 1
1	0	00	(128) (52) = 6,656	26 at Side 1
1	1	01	(256) (52) = 13,312	
0	0	01	(256) (15) = 3,840	15 at Side 0
0	1	02	(512) (15) = 7,680	or 15 at Side 1
1	0	01	(256) (30) = 7,680	15 at Side 1
1	1	02	(512) (30) = 15,360	
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512) (16) = 8,192	8 at Side 1
1	1	03	(1024) (16) = 16,384	

The "multi-track" function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 0, Side 0 and completing at Sector L, Side 1 (Sector L = last sector on the side). Note, this function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, then DTL defines the data length which the FDC must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector, is not sent to the Data Bus. The FDC reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to 0FFH.

At the completion of the Read Data Command, the head is not unloaded until after Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

After reading the ID and Data Fields in each sector, the FDC checks the CRC bytes. If a read error is detected (incorrect CRC in ID field), the FDC sets the DE (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC also sets the DD (Data Error in Data Field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

If the FDC reads a Deleted Data Address Mark off the diskette, and the SK bit (bit D5 in the first Command Word) is not set (SK = 0), then the FDC sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector.

During disk data transfers between the FDC and the processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, and every 13 μs in the MFM Mode, or the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID Information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 4 shows the values for C, H, R, and N, when the processor terminates the Command.

TABLE 4. ID INFORMATION WHEN PROCESSOR TERMINATES COMMAND

MT	EOT	Final Sector Transferred to Processor	ID Information at Result Phase			
			C	H	R	N
0	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	C+1	NC	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	NC	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				
1	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	NC	LSB	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	LSB	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				

Notes: 1. NC (No Change): The same value as the one at the beginning of command execution.

2. LSB (Least Significant Bit): The least significant bit of H is complemented.

WRITE DATA

A set of nine (9) bytes are required to set the FDC into the Write Data mode. After the Write Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Fields. When the current sector number ("R"), stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC takes data from the processor byte-by-byte via the data bus, and outputs it to the FDD.

After writing data into the current sector, the Sector Number stored in "R" is incremented by one, and the next data field is written into. The FDC continues this "Multi-Sector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written then the remainder of the data field is filled with 00 (zeros).

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a read error (incorrect CRC) in one of the ID Fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high), and terminates the Write Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

The Write Command operates in much the same manner as the Read Command. The following items are the same; refer to the Read Data Command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag

- Head Unload Time Interval
- ID Information when the processor terminates command (see Table 2)
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC must occur every 31 μs in the FM mode, and every 15 μs in the MFM mode. If the time interval between data transfers is longer than this then the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Write Data Command.

WRITE DELETED DATA

This command is the same as the Write Data Command except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

READ DELETED DATA

This command is the same as the Read Data Command except that when the FDC detects a Data Address Mark at the beginning of a Data Field (and SK = 0 (low)), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, then the FDC skips the sector with the Data Address Mark and reads the next sector.

READ A TRACK

This command is similar to READ DATA Command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering the INDEX HOLE, the FDC starts reading all data fields on the track as continuous blocks of data. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multi-track or skip operations are not allowed with this command.

This command terminates when EOT number of sectors have been read. If the FDC does not find an ID Address Mark on the diskette after it encounters the INDEX HOLE for the second time, then it sets the MA (missing address mark) flag in Status Register 1 to a 1 (high), and terminates the command. (Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.)

READ ID

The READ ID Command is used to give the present position of the recording head. The FDC stores the values from the first ID Field it is able to read. If no proper ID Address Mark is found on the diskette, before the INDEX HOLE is encountered for the second time then the MA (Missing Address Mark) flag in Status Register 1 is set to a 1 (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a 1 (high) and the command is terminated.

FORMAT A TRACK

The Format Command allows an entire track to be formatted. After the INDEX HOLE is detected, Data is written on the Diskette: Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) Format are recorded. The particular format which will be written is controlled by the values programmed into N (number of bytes/sector), SC (sectors/cylinder), GPL (Gap Length), and D (Data Pattern) which are supplied by the processor during the Command Phase. The Data Field is filled with the Byte of data stored in D. The ID Field for each sector is supplied by the processor; that is, four data requests per sector are made by the FDC for C (Cylinder Number), H (Head Number), R (Sector Number) and N (Number of Bytes/Sector). This allows the diskette to be formatted with nonsequential sector numbers, if desired.

After formatting each sector, the processor must send new values for C, H, R, and N to the 8272 for each sector on the track. The contents of the R register is incremented by one after each sector is formatted, thus, the R register contains a value of R + 1 when it is read during the Result Phase. This incrementing and formatting continues for the whole track until the FDC encounters the INDEX HOLE for the second time, whereupon it terminates the command.

If a FAULT signal is received from the FDD at the end of a write operation, then the FDC sets the EC flag of Status Register 0 to a 1 (high), and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also the loss of a READY signal at the beginning of a command execution phase causes command termination.

Table 5 shows the relationship between N, SC, and GPL for various sector sizes:

TABLE 5. SECTOR SIZE RELATIONSHIPS

FORMAT	SECTOR SIZE	N	SC	GPL[1]	GPL[2]	REMARKS
FM Mode	128 bytes/Sector	00	$1A_{(16)}$	$07_{(16)}$	$1B_{(16)}$	IBM Diskette 1
	256	01	$0F_{(16)}$	$0E_{(16)}$	$2A_{(16)}$	IBM Diskette 2
	512	02	08	$1B_{(16)}$	$3A_{(16)}$	
FM Mode	1024 bytes/Sector	03	04	—	—	
	2048	04	02	—	—	
	4096	05	01	—	—	
MFM Mode	256	01	$1A_{(16)}$	$0E_{(16)}$	$36_{(16)}$	IBM Diskette 2D
	512	02	$0F_{(16)}$	$1B_{(16)}$	$54_{(16)}$	
	1024	03	08	$35_{(16)}$	$74_{(16)}$	IBM Diskette 2D
	2048	04	04	—	—	
	4096	05	02	—	—	
	8192	06	01	—	—	

Note: 1. Suggested values of GPL in Read or Write Commands to avoid splice point between data field and ID field of contiguous sections.

2. Suggested values of GPL in format command.

SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 6 shows the status of bits SH and SN under various conditions of SCAN.

TABLE 6. SCAN STATUS CODES

COMMAND	STATUS REGISTER 2		COMMENTS
	BIT 2 = SN	BIT 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 27 μs (FM Mode) or 13 μs (MFM Mode). If an Overrun occurs the FDC terminates the command.

SEEK

The read/write head within the FDD is moved from cylinder to cylinder under control of the Seek Command. The FDC compares the PCN (Present Cylinder Number) which is the current head position with the NCN (New Cylinder Number), and performs the following operation if there is a difference:

PCN < NCN: Direction signal to FDD set to a 1 (high), and Step Pulses are issued. (Step In.)

PCN > NCN: Direction signal to FDD set to a 0 (low), and Step Pulses are issued. (Step Out.)

The rate at which Step Pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each Step Pulse is issued NCN is compared against PCN, and when NCN = PCN, then the SE (Seek End) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

During the Command Phase of the Seek operation the FDC is in the FDC BUSY state, but during the Execution Phase it is in the NON BUSY state. While the FDC is in the NON BUSY state, another Seek Command may be issued, and in this manner parallel seek operations may be done on up to 4 Drives at once.

If an FDD is in a NOT READY state at the beginning of the command execution phase or during the seek operation, then the NR (NOT READY) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

RECALIBRATE

This command causes the read/write head within the FDD to retract to the Track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the Track 0 signal from the FDD. As long as the Track 0 signal is low, the Direction signal remains 1 (high) and Step Pulses are issued. When the Track 0 signal goes high, the SE (SEEK END) flag in Status Register 0 is set to a 1 (high) and the command is terminated. If the Track 0 signal is still low after 77 Step Pulses have been issued, the FDC sets the SE (SEEK END) and EC (EQUIPMENT CHECK) flags of Status Register 0 to both 1s (highs), and terminates the command.

The ability to overlap RECALIBRATE Commands to multiple FDDs, and the loss of the READY signal, as described in the SEEK Command, also applies to the RECALIBRATE Command.

SENSE INTERRUPT STATUS

An Interrupt signal is generated by the FDC for one of the following reasons:

1. Upon entering the Result Phase of:
   a. Read Data Command
   b. Read a Track Command
   c. Read ID Command
   d. Read Deleted Data Command
   e. Write Data Command
   f. Format a Cylinder Command
   g. Write Deleted Data Command
   h. Scan Commands
2. Ready Line of FDD changes state
3. End of Seek or Recalibrate Command
4. During Execution Phase in the NON-DMA Mode

Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. However, interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt Status Command. This command when issued resets the interrupt signal and via bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt.

TABLE 7. SEEK, INTERRUPT CODES

SEEK END BIT 5	INTERRUPT CODE		CAUSE
	BIT 6	BIT 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

Neither the Seek or Recalibrate Command have a Result Phase. Therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them and to provide verification of the head position (PCN).

SPECIFY

The Specify Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the Execution Phase of one of the Read/Write Commands to the head unload state. This timer is programmable from 16 to 240 ms in increments of 16 ms (01 = 16 ms, 02 = 32 ms . . . . OF = 240 ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1 ms, E = 2 ms, D = 3 ms, etc.). The HLT (Head Load Time) defines the time between when the Head Load signal goes high and when the Read/Write operation starts. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2 ms, 02 = 4 ms, 03 = 6 ms . . . . FE = 254 ms).

The time intervals mentioned above are a direct function of the clock (CLK on pin 19). Times indicated above are for an 8 MHz clock, if the clock was reduced to 4 MHz (mini-floppy application) then all time intervals are increased by a factor of 2.

The choice of DMA or NON-DMA operation is made by the ND (NON-DMA) bit. When this bit is high (ND = 1) the NON-DMA mode is selected, and when ND = 0 the DMA mode is selected.

SENSE DRIVE STATUS

This command may be used by the processor whenever it wishes to obtain the status of the FDDs. Status Register 3 contains the Drive Status information.

INVALID

If an invalid command is sent to the FDC (a command not defined above), then the FDC will terminate the command. No interrupt is generated by the 8272 during this condition. Bit 6 and bit 7 (DIO and RQM) in the Main Status Register are both high ("1") indicating to the processor that the 8272 is in the Result Phase and the contents of Status Register 0 (STO) must be read. When the processor reads Status Register 0 it will find a 80H indicating an invalid command was received.

A Sense Interrupt Status Command must be sent after a Seek or Recalibrate interrupt, otherwise the FDC will consider the next command to be an Invalid Command.

In some applications the user may wish to use this command as a No-Op command, to place the FDC in a standby or no operation state.

TABLE 8. STATUS REGISTERS

BIT NO.	BIT NAME	BIT SYMBOL	DESCRIPTION
		STATUS REGISTER 0	
D_7	Interrupt Code	IC	$D_7 = 0$ and $D_6 = 0$ Normal Termination of Command, (NT). Command was completed and properly executed.
D_6			$D_7 = 0$ and $D_6 = 1$ Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed.
			$D_7 = 1$ and $D_6 = 0$ Invalid Command issue, (IC). Command which was issued was never started.
			$D_7 = 1$ and $D_6 = 1$ Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK Command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault Signal is received from the FDD, or if the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US 1	These flags are used to indicate a
D_0	Unit Select 0	US 0	Drive Unit Number at Interrupt
		STATUS REGISTER 1	
D_7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set.
D_6			Not used. This bit is always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set.
D_4	Over Run	OR	If the FDC is not serviced by the main-systems during data transfers, within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN Command, if the FDC cannot find the Sector specified in the IDR Register, this flag is set.
			During executing the READ ID Command, if the FDC cannot read the ID field without an error, then this flag is set.
			During the execution of the READ A Cylinder Command, if the starting sector cannot be found, then this flag is set.
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A Cylinder Command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set.
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set.
		STATUS REGISTER 2	
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During executing the READ DATA or SCAN Command, if the FDC encounters a Sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related with the ND bit, and when the contents of C on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution, the SCAN Command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During executing the SCAN Command, if the FDC cannot find a Sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related with the ND bit, and when the content of C on the medium is different from that stored in the IDR and the content of C is FF, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.
		STATUS REGISTER 3	
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RDY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Side	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of Side Select signal to the FDD.
D_1	Unit Select 1	US 1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US 0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature ..... −10°C to +70°C
Storage Temperature ..... −40°C to +125°C
All Output Voltages ..... −0.5 to +7 Volts
All Input Voltages ..... −0.5 to +7 Volts
Supply Voltage V_{CC} ..... −0.5 to +7 Volts
Power Dissipation ..... 1 Watt

*T_A = 25°C

COMMENT: *Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

DC CHARACTERISTICS

T_A = 0°C to +70°C; V_{CC} = +5V ± 5%

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		MIN	MAX		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{IL}	(CLK & WR CLK)	−0.5	0.65	V	
V_{IH}	(CLK & WR CLK)	2.4	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4	V_{CC}	V	I_{OH} = −200 μA
I_{CC}	V_{CC} Supply Current		150	mA	
I_{IL}	Input Load Current (All Input Pins)		10 −10	μA μA	V_{IN} = V_{CC} V_{IN} = 0V
I_{LOH}	High Level Output Leakage Current		10	μA	V_{OUT} = V_{CC}
I_{LOL}	Low Level Output Leakage Current		−10	μA	V_{OUT} = +0.45V

CAPACITANCE

T_A = 25°C; f_c = 1 MHz; V_{CC} = 0V

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		MIN	MAX		
$C_{IN(\Phi)}$	Clock Input Capacitance		20	pF	All Pins Except Pin Under Test Tied to AC Ground
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS

$t_A = 0°C$ to $70°C$, $V_{CC} = +5.0V \pm 5\%$

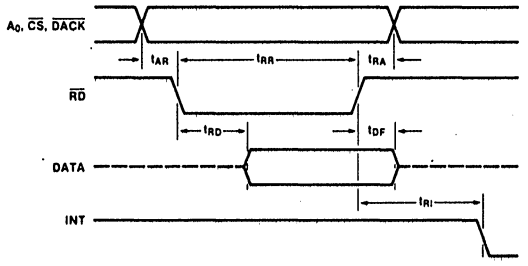
SYMBOL	PARAMETER	TYP [1]	MIN	MAX	UNIT	TEST CONDITIONS
t_{CY}	Clock Period		125		ns	
t_{CH}	Clock High Period		40		ns	
t_{RST}	Reset Width		14		t_{CY}	
Read Cycle						
t_{AR}	Select Setup to $\overline{RD}\downarrow$		0		ns	
t_{RA}	Select Hold from $\overline{RD}\uparrow$		0		ns	
t_{RR}	$\overline{RD}$ Pulse Width		250		ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$			200	ns	$C_L = 100$ pF
t_{DF}	Output Float Delay		20	100	ns	$C_L = 100$ pF
Write Cycle						
t_{AW}	Select Setup to $\overline{WR}\downarrow$		0		ns	
t_{WA}	Select Hold from $\overline{WR}\uparrow$		0		ns	
t_{WW}	$\overline{WR}$ Pulse Width		250		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$		150		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$		5		ns	
Interrupts						
t_{RI}	INT Delay from $\overline{RD}\uparrow$			500	ns	
t_{WI}	INT Delay from $\overline{WR}\uparrow$			500	ns	
DMA						
t_{RQCY}	DRQ Cycle Period		13		µs	
t_{AKRQ}	$\overline{DACK}\downarrow$ to DRQ↓			200	ns	
t_{RQR}	DRQ↑ to $\overline{RD}\downarrow$		800		ns	8 MHz clock
t_{RQW}	DRQ↑ to $\overline{WR}\downarrow$		250		ns	8 MHz clock
t_{RQRW}	DRQ↑ to $\overline{RD}\uparrow$ or $\overline{WR}\uparrow$			12	µs	8 MHz clock
FDD Interface						
t_{WCY}	WCK Cycle Time	2 or 4			µs	MFM = 0 Note 2
		1 or 2				MFM = 1
t_{WCH}	WCK High Time	250	80	350	ns	
t_{CP}	Pre-Shift Delay from WCK↑		20	100	ns	
t_{CD}	WDA Delay from WCK↑		20	100	ns	
t_{WDD}	Write Data Width		$t_{WCH} - 50$		ns	
t_{WE}	WE↑ to WCK↑ or WE↓ to WCK↓ Delay		20	100	ns	
t_{WWCY}	Window Cycle Time	2			µs	MFM = 0
		1				MFM = 1
t_{WRD}	Window Setup to RDD↑		15		ns	
t_{RDW}	Window Hold from RDD↓		15		ns	
t_{RDD}	RDD Active Time (HIGH)		40		ns	
FDD SEEK/ DIRECTION/ STEP						
t_{US}	$US_{0,1}$ Setup to $\overline{RW}$/SEEK↑		12		µs	8 MHz clock
t_{SU}	$US_{0,1}$ Hold from $\overline{RW}$/SEEK↓		15		µs	
t_{SD}	$\overline{RW}$/SEEK Setup to LCT/DIR		7		µs	
t_{DS}	$\overline{RW}$/SEEK Hold from LCT/DIR		30		µs	
t_{DST}	LCT/DIR Setup to FR/STEP↑		1		µs	
t_{STD}	LCT/DIR Hold from FR/STEP↓		24		µs	
t_{STU}	$DS_{0,1}$ Hold from FR/Step↓		5		µs	
t_{STP}	STEP Active Time (High)	5			µs	
t_{SC}	STEP Cycle Time		33		µs	Note 3
t_{FR}	FAULT RESET Active Time (High)		8	10	µs	
t_{IDX}	INDEX Pulse Width	625			µs	
t_{TC}	Terminal Count Width		1		t_{CY}	

NOTES:

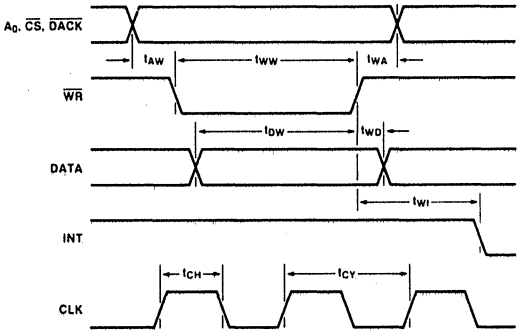
1. Typical values for $T_A = 25°C$ and nominal supply voltage.
2. The former values are used for standard floppy and the latter values are used for mini-floppies.
3. $t_{SC} = 33\mu s$ min. is for different drive units. In the case of same unit, t_{SC} can be ranged from 1 ms to 16 ms with 8 MHz clock period, and 2 ms to 32 ms with 4 MHz clock, under software control.

TIMING WAVEFORMS

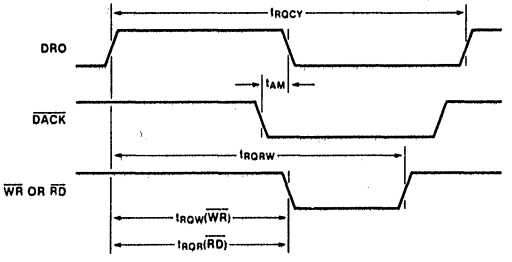
PROCESSOR READ OPERATION



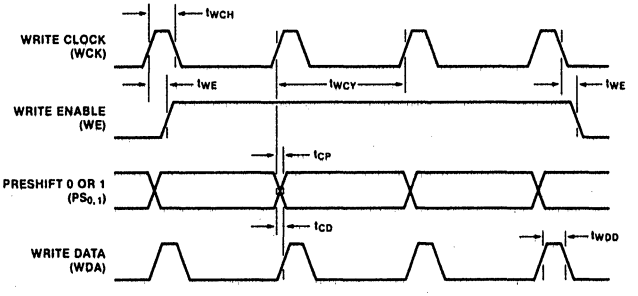
PROCESSOR WRITE OPERATION



DMA OPERATION

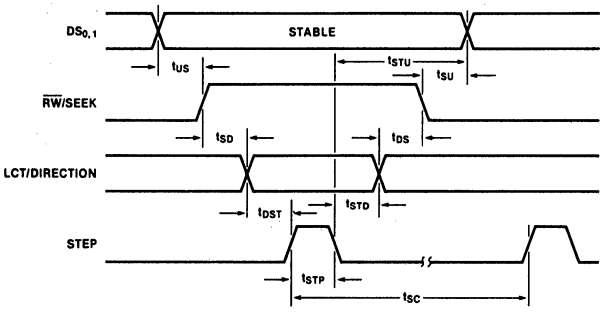


FDD READ OPERATION

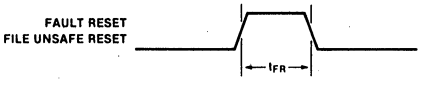


	PRESHIFT 0	PRESHIFT 1
NORMAL	0	0
LATE	0	1
EARLY	1	0
INVALID	1	1

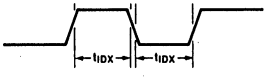
SEEK OPERATION



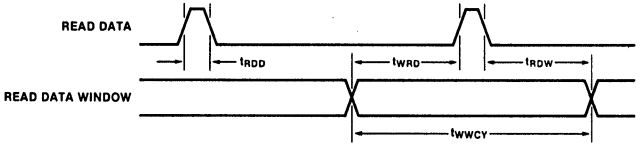
FLT RESET



INDEX

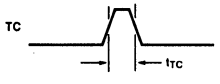


FDD READ OPERATION

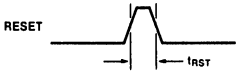


NOTE: EITHER POLARITY DATA WINDOW IS VALID.

TERMINAL COUNT

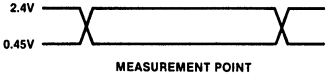


RESET

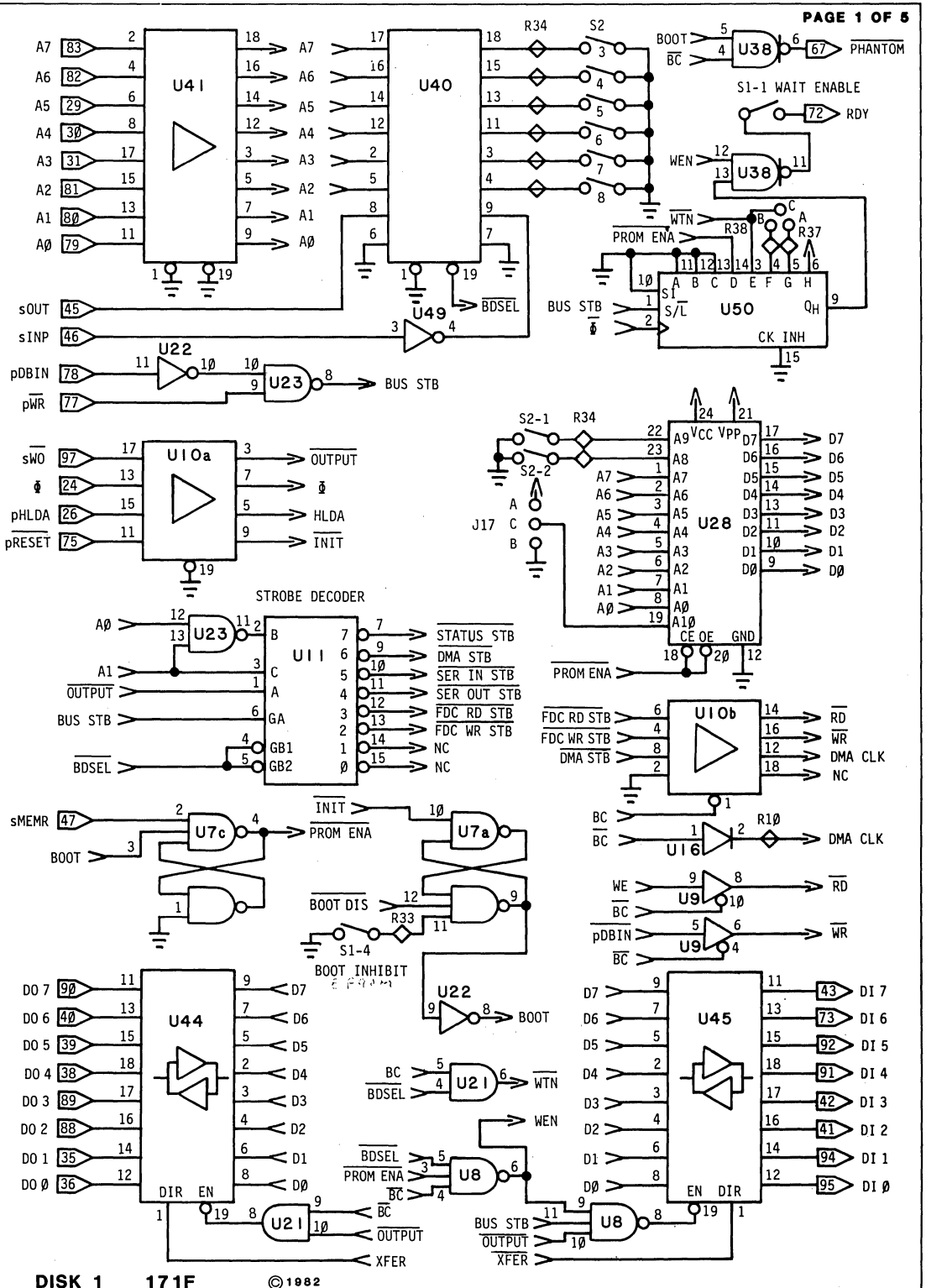


A.C. TIMING MEASUREMENT CONDITIONS

INPUT WAVEFORM

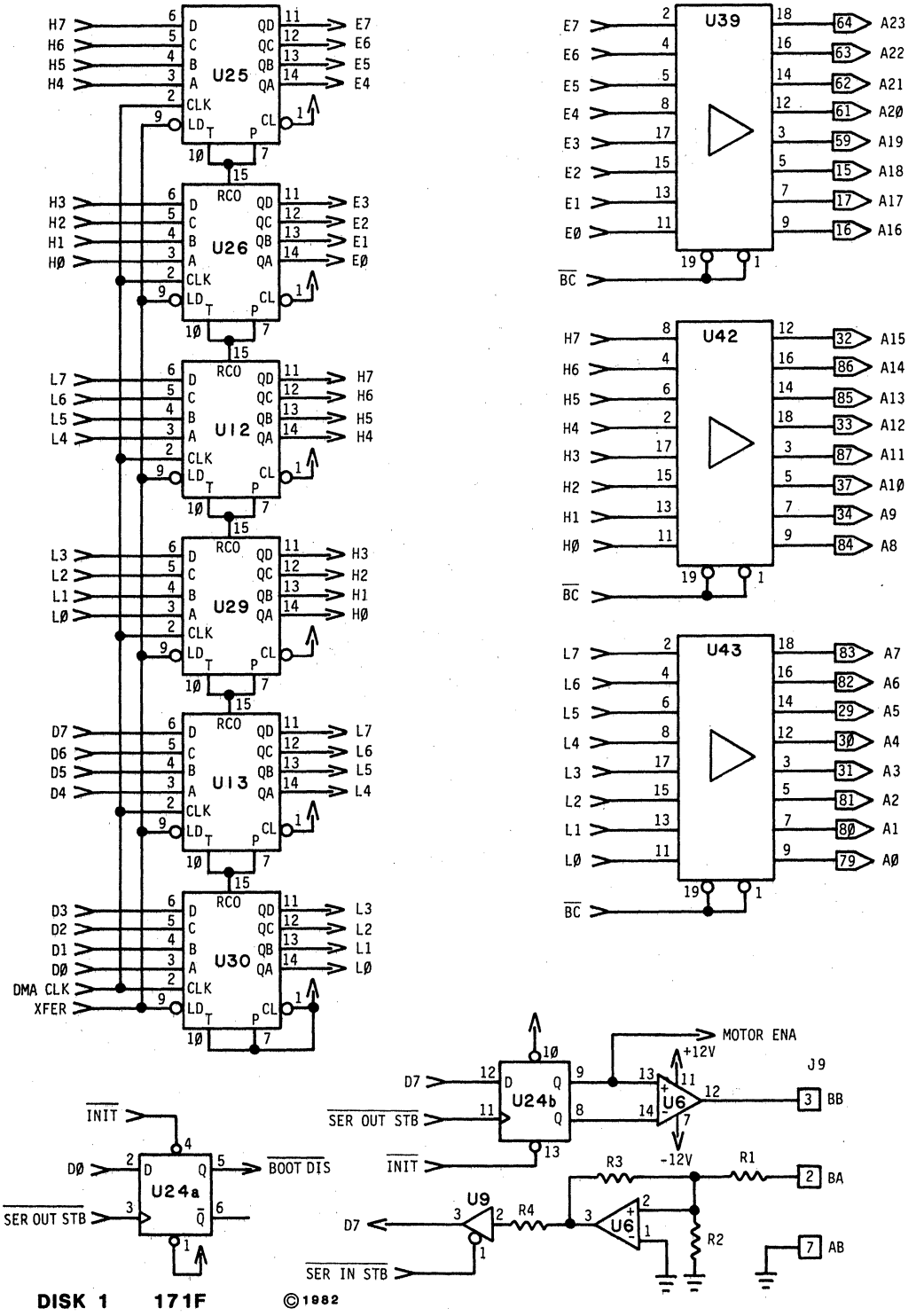


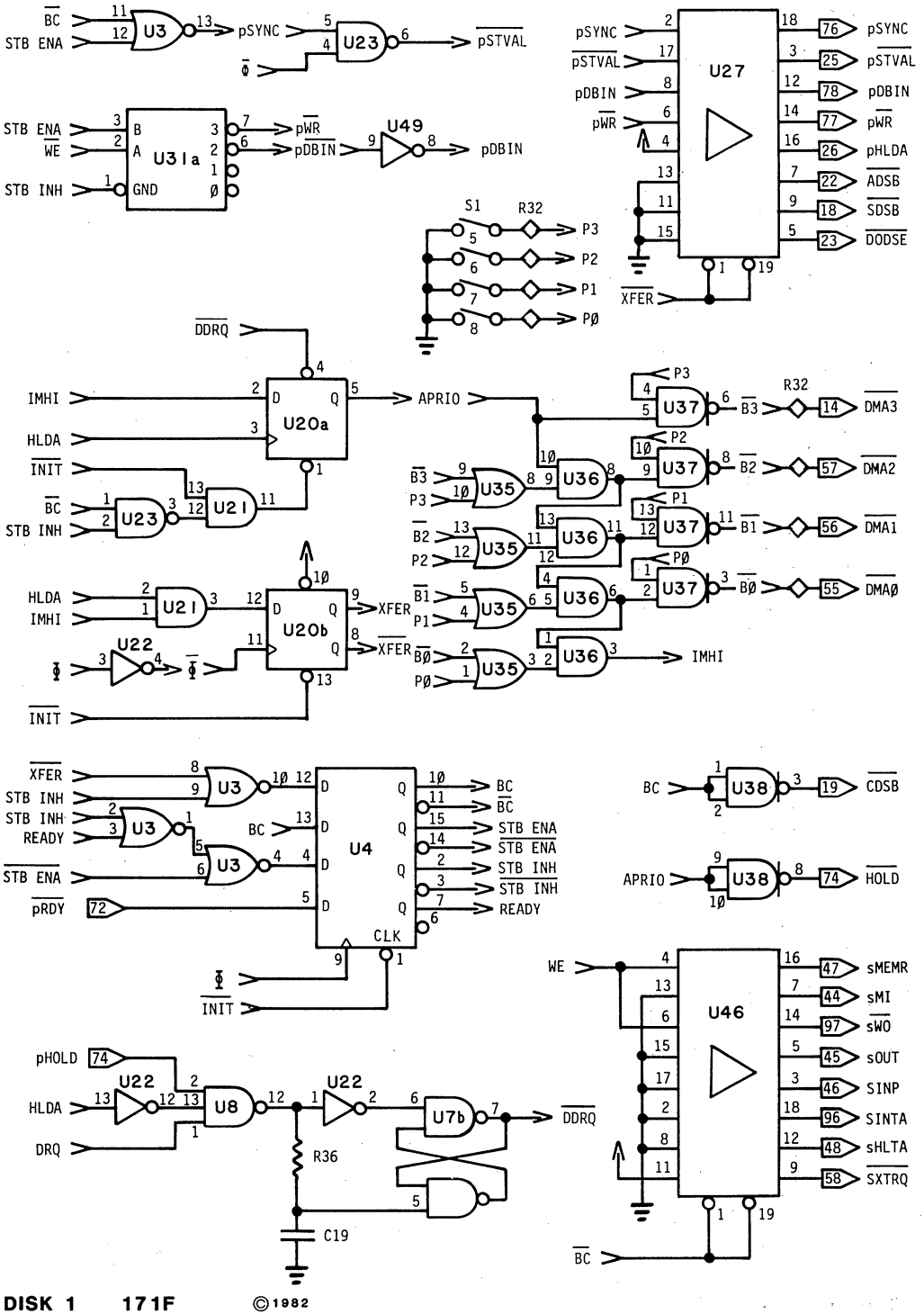
MEASUREMENT POINT
INPUT: 2.0V ∞ 0.8V
OUTPUT: 2.0V ∞ 0.8V

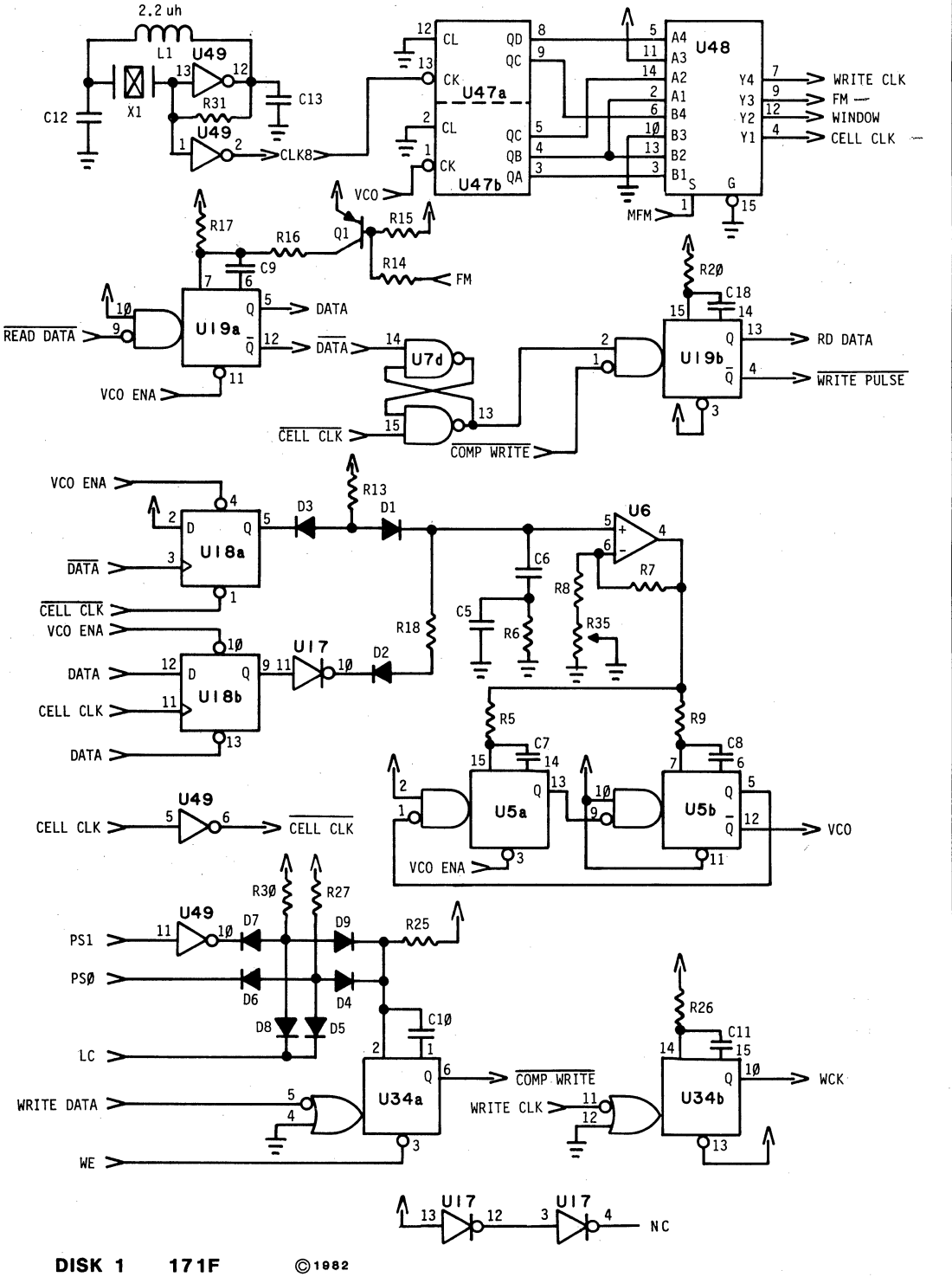


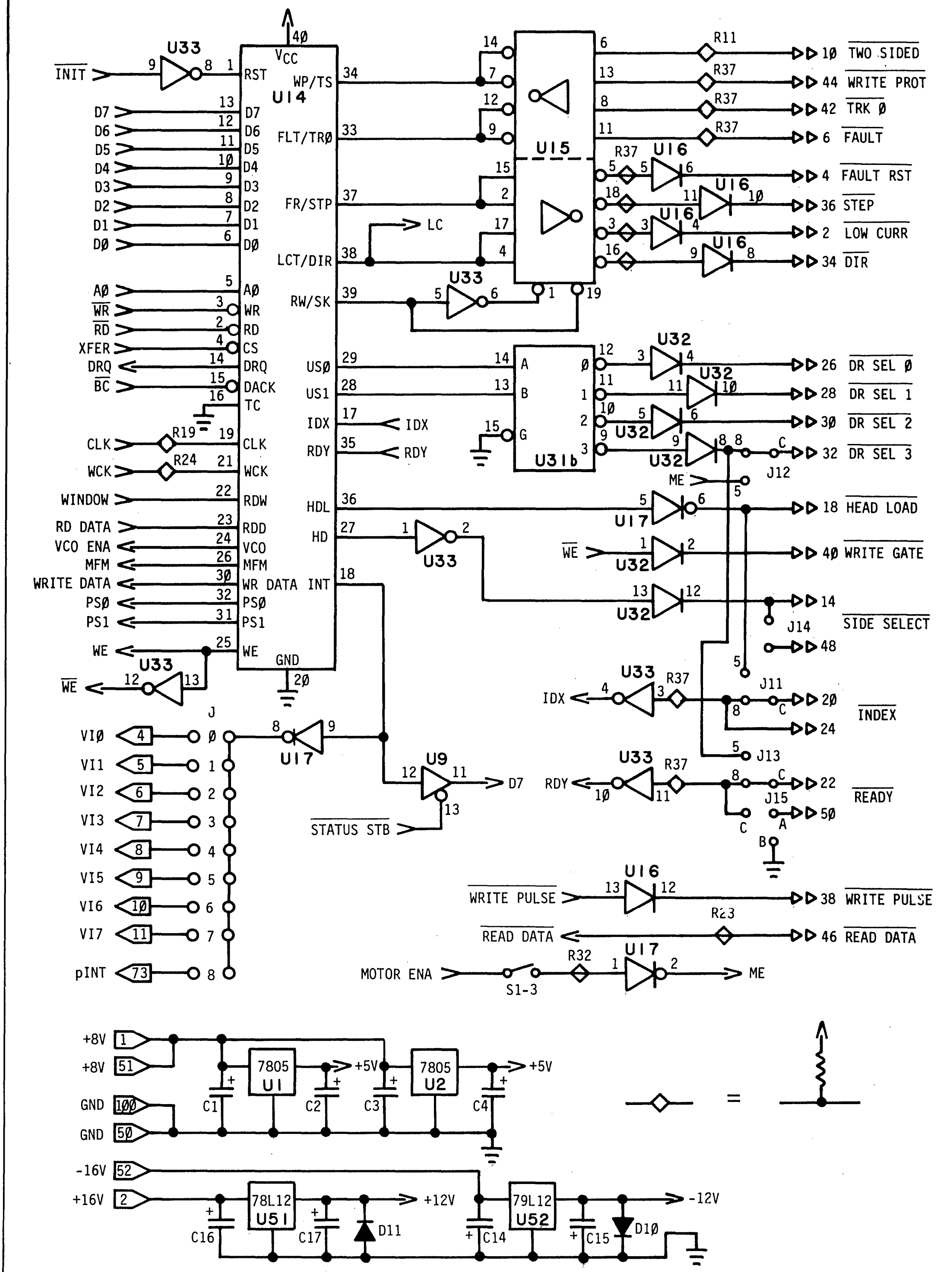
DISK 1 171F ©1982

COMPUPRO division GODBOUT ELECTRONICS


U25
U26
U12
U29
U13
U30
H7
H6
H5
H4
H3
H2
H1
HØ
L7
L6
L5
L4
L3
L2
L1
LØ
D7
D6
D5
D4
D3
D2
D1
DØ
DMA CLK
XFER
E7
E6
E5
E4
E3
E2
E1
EØ
U39
U42
U43
A23
A22
A21
A20
A19
A18
A17
A16
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
AØ
BC
U24a
U24b
U9
U6
INIT
SER OUT STB
SER IN STB
BOOT DIS
MOTOR ENA
+12V
-12V
R1
R2
R3
R4
J9
BB
BA
AB


BC
STB ENA
U3
pSYNC
U23
pSTVAL
STB ENA
WE
STB INH
U31a
pWR
pDBIN
U49
pDBIN
pSYNC
pSTVAL
pDBIN
pWR
U27
pSYNC
pSTVAL
pDBIN
pWR
pHLDA
ADSB
SDSB
DODSE
S1
R32
P3
P2
P1
PØ
XFER
DDRQ
IMHI
HLDA
INIT
BC
STB INH
U20a
U21
U23
APRIO
U37
U36
U35
B3
B2
B1
BØ
DMA3
DMA2
DMA1
DMAØ
U21
U20b
U22
XFER
IMHI
XFER
INIT
STB INH
STB INH
READY
STB ENA
pRDY
U3
U4
BC
STB ENA
STB INH
READY
CLK
INIT
BC
U38
CDSB
APRIO
U38
HOLD
WE
U46
sMEMR
sMI
sWO
sOUT
SINP
SINTA
sHLTA
SXTRQ
pHOLD
HLDA
DRQ
U22
U8
U22
U7b
DDRQ
R36
C19
BC


2.2 uh
L1
U49
C12
X1
R31
C13
CLK8
U47a
U47b
U48
WRITE CLK
FM
WINDOW
CELL CLK
VCO
MFM
R17
R16
Q1
R15
R14
FM
C9
R2Ø
C18
U19a
U19b
READ DATA
DATA
VCO ENA
U7d
CELL CLK
COMP WRITE
RD DATA
WRITE PULSE
R13
D3
D1
U18a
U18b
U17
D2
R18
U6
C6
R7
R8
C5
R6
R35
R5
R9
C7
C8
U5a
U5b
VCO
R3Ø
R27
R25
PS1
PSØ
D7
D9
D6
D4
D8
D5
C1Ø
LC
WRITE DATA
WE
U34a
U34b
R26
C11
WCK
NC


U33
U14
INIT
RST
VCC
WP/TS
FLT/TRØ
FR/STP
LCT/DIR
RW/SK
USØ
US1
IDX
RDY
HDL
HD
INT
GND
D7
D6
D5
D4
D3
D2
D1
DØ
AØ
WR
RD
XFER
CS
DRQ
BC
DACK
TC
CLK
WCK
WINDOW
RDW
RD DATA
RDD
VCO ENA
VCO
MFM
WRITE DATA
WR DATA
PSØ
PS1
WE
U15
U16
U31b
U32
U17
U9
R11
R19
R24
R37
R32
R23
LC
1Ø TWO SIDED
44 WRITE PROT
42 TRK Ø
6 FAULT
4 FAULT RST
36 STEP
2 LOW CURR
34 DIR
26 DR SEL Ø
28 DR SEL 1
3Ø DR SEL 2
32 DR SEL 3
J12
ME
18 HEAD LOAD
4Ø WRITE GATE
14
J14
SIDE SELECT
48
J11
2Ø
24
INDEX
J13
22
J15
5Ø
READY
STATUS STB
J
VIØ
VI1
VI2
VI3
VI4
VI5
VI6
VI7
pINT
73
WRITE PULSE
38 WRITE PULSE
READ DATA
46 READ DATA
MOTOR ENA
S1-3
+8V
GND
7805
U1
U2
+5V
C1
C2
C3
C4
-16V
+16V
78L12
U51
79L12
U52
+12V
-12V
C16
C17
D11
C14
C15
D1Ø

PARTS LIST

SEMICONDUCTORS UNIT#	PART#
U1	7805
U2	7805
U3	74LS02
U4	74LS175
U5	74LS221
U6	4136
U7	74LS279
U8	74LS10
U9	74LS125
U10	74LS244
U11	74LS138
U12	74LS163
U13	74LS163
U14	765/8272
U15	74LS240
U16	7407
U17	7406
U18	74LS74
U19	74LS221
U20	74LS74
U21	74LS08
U22	74LS04
U23	74LS00
U24	74LS74
U25	74LS163
U26	74LS163
U27	74LS244
U28	2758/2716
U29	74LS163
U30	74LS163
U31	74LS139
U32	7407
U33	74LS04
U34	96LS02
U35	74LS32
U36	74LS08
U37	74LS38
U38	74LS38
U39	74LS244
U40	25LS2521
U41	74LS244
U42	74LS244
U43	74LS244
U44	74LS245
U45	74LS245

SEMICONDUCTORS UNIT#	PARTS#
U46	74LS244
U47	74LS393
U48	74LS157
U49	74LS04
U50	74LS165
U51	78L12
U52	79L12

RESISTOR	
R1	4.7K OHM
R2	4.7K OHM
R3	33K OHM
R4	4.7K OHM
R5*	1.5K OHM
or	3.0K OHM
R6*	1.0K OHM
or	750 OHM
R7	47K OHM
R8	33K OHM
R9*	1.5K OHM
or	3.0K OHM
R10	1.2K OHM
R11	150 OHM
R12	150 OHM SIP
R13*	3.9K OHM
or	2.2K OHM
R14	4.7K OHM
R15	4.7K OHM
R16*	5.1K OHM
or	10K OHM
R17*	5.1K OHM
or	10K OHM
R18*	3.9K OHM
or	2.2K OHM
R19	1.2K OHM
R20*	1.8K OHM
or	3.6K OHM
R21	not used
R22	not used
R23	150 OHM
R24	1.2K OHM
R25	15K OHM
R26	2.4K OHM
R27	75K OHM

RESISTOR	
R28	not used
R29	not used
R30	75K OHM
R31	1.0K OHM
R32	5.1K OHM SIP
R33	4.7K OHM
R34	5.1K OHM SIP
R35	10K TRIMPOT
R36	330 OHM
R37	4.7K OHM
R38	4.7K OHM

CAPACITOR	
C1-C4	39 uF
C5	.01 uF
C6	.1 uF
C7	220 pF
C8	220 pF
C9	270 pF
C10	220 pF
C11	220 pF
C12	250 pF
C13	600 pF
C14-C17	1.0 uF
C18	220 pF
C19	.0027 uF
(31)	.01 uF

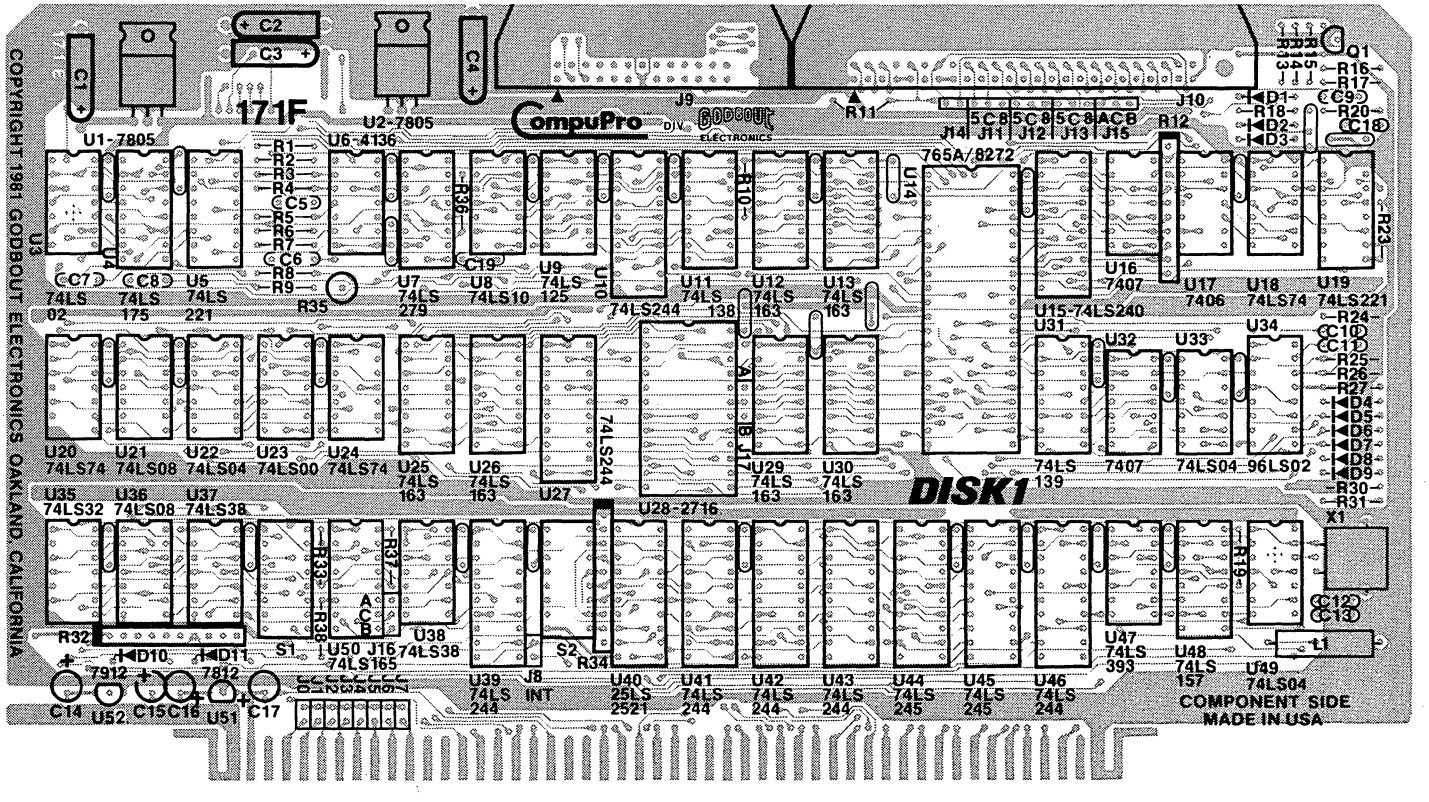
CRYSTAL	
X1*	8.00 MHz
or	4.00 MHz

INDUCTOR	
L1*	2.2 uH
	10 uH

TRANSISTOR	
Q1	2N3906

DIODE	
D1-D9	809-36
D10-D11	1N914

*part alteration for 5.25" drives



COMPONENT LAYOUT

SOFTWARE SECTION

1.0 INTRODUCTION

The purpose of this manual is to describe the software supplied with the Godbout Electronics / CompuPro Disk 1 Floppy Disk Controller for the standard IEEE 696 (S-100) bus. In addition, a section of this manual is devoted to describing how a modified CompuPro CP/M 2.2 CBIOS can be integrated into the CP/M system.

This manual is written for purchaser who are familiar with the CP/M BIOS customizing techniques. If CP/M 2.2 was purchased with the Disk 1 controller then all corrective patches have been installed. On the other hand if the CP/M was purchased for a different controller then be sure to obtain and install all CP/M corrective patches.

The information contained within the document is divided as follows:

a. Software User's Guide
b. Software Features
c. Software Internal Design
d. CBIOS Customization Guide

The Software User's Guide describes how to use the following software packages:

a. FORMAT.com --disk formatter
b. COPY.com / CONVERT.com -- disk copy utility
c. DSK.com -- CP/M 1.4 compatibility toggle

The next section describes the external features of the supplied software. This section describes the enhancements and assumptions made by the CompuPro CBIOS routines and utility routines.

The Software Internal Design section describes the design of the supplied software. This section should help the user understand how the software components work; thus enabling the purchaser to modify the software for his hardware configuration.

The final section has a step-by-step method describing how to modify and integrate the CompuPro Disk 1 CBIOS into a CP/M system.

2.0 APPLICABLE DOCUMENTS

The reader should be familiar with the following documents.

a. NEC uPD765 Floppy Disk Controller Application Note
b. CompuPro Disk 1 Floppy Disk Controller Description
c. An Introduction to CP/M Features and Facilities
d. CP/M 2.0 User's Guide for CP/M 1.4 Owners
e. CP/M 2.0 Alteration Guide
f. CP/M 2.0 Interface Guide

3.0 SOFTWARE USER'S GUIDE

The following paragraphs describe the operating features for the supplied utility programs.

3.1 FORMAT.com -- Floppy Disk Format Utility

The FORMAT.com utility program is supplied so that purchasers of the Disk 1 board can change the density of his disk (data is destroyed). The FORMAT.com utility formats floppy disk in IBM compatible formats; not all disk controllers are IBM compatible.

The FORMAT utility contains the Disk1 board interface routines allowing the user to run under a current IEEE 696 CP/M 2.2 system.

The FORMAT utility has simple and straight forward operating instructions. The user is prompted for input as needed to control the utilities flow. The user initiates the FORMAT utility by entering the following:

```
FORMAT
or
FORMAT drive
```

If the drive is not specified on the command line then the FORMAT utility prompts the user with the following line:

```
Specify drive (A: - D:) :
```

The user now enters the drive (A thru D) and the FORMAT utility proceeds.

After the user has specified the drive to be formatted, FORMAT attempts to determine the specified disks format. The disk format or lack there of is displayed for the user and a message requesting the new format selection is displayed. The format of these displays are as follows:

```
Disk is Formatted as 1024 byte sectors.

Select Disk format mode (0,1,2,3):
   0 = 128          2 = 512
   1 = 256          3 = 1024     > _
```

The user now enter the disk formatting selection and the FORMAT utility begins formatting the disk. As each track is formatted an F appears on the screen (total of 77). After formatting all the tracks the FORMAT utility begins a verify operation. This verify operation informs the user of possible bad spots on the floppy disk. These displays and a description of the output follows:

```
Confirm ready for format on disk drive B (y)_

          1         2                  6         7
01234567890123456789    . . .    01234567890123456
FFFFFFFFFFFFFFFFFFFF    . . .    FFFFFFFFFFFFFFFFF
 VVVVVVSSSVVVVEEEVVVV   . . .    VVVVVVVVVSVVSVVEEE
```

These symbols have the following meaning:

F -- successful format operation.	S -- error occrued by retry worked.
V -- read verified.	E -- hard failure.

After the verification, the FORMAT utility asks the user if another disk is to be formatted. The user response will result in one of three actions as follows:

1. Return to CP/M
2. Formatting another disk with the same parameters.
3. Change formatting parameters.

The FORMAT utility uses a lot of user interactions to control the utilities flow. The user can get a general idea of the inputs required by entering the following command line:

```
FORMAT ?
```

This will result in the following display:

```
FORMAT will format a diskette to be used with
the CompuPro disk subsystem.

The FORMAT program is initiated by:

        FORMAT <dr:><cr>

Where <cr> is the Carriage Return.
If the drive <dr:> is not specified, it will be prompted for.
```

3.2 COPY.com / CONVERT.com -- Disk Copy Utilities.

The COPY and CONVERT.com utility programs perform diskette copy functions. The CONVERT program inverts the data while copying, because some controllers record the data inverted from normal form.

These programs are intended to be used to copy an entire diskette to another diskette. They perform this by reading in a track of data, write it out, reading it back in and comparing.

The user must tell these programs the following information:

1. Area of Diskette to be copied.
2. Source drive.
3. Destination drive.

These programs get this information by prompting the user with the following messages:

```
CompuPro COPY Utility Version 2.X.

Do you want to copy:
```

```
SYSTEM tracks only?      (type S)
DATA tracks only?        (type D)
ALL of the disk?         (type A)
Exit back to system?     (type X) _

Source drive?  (A, B, C, or D) _
Destination drive?  (A, B, C, or D) _

Put source disk on X
Put destination disk on Y
Then type <return>
```

As with the FORMAT program the following is available:

```
COPY ?
```

This will result in the following display:

```
Copy will copy disks on the CompuPro
disk subsystem.  To start, type:

       COPY <portion><cr>

Where <cr> is Carriage Return
and <portion> is S = system, D = data or
A = all.  If not entered, a prompting message
will be output.

The source and destination drive prompts occur.
```

A similar message will be output by the CONVERT program.

3.3 DSK.com -- CP/M 1.4 Compatibility Toggle.

The DSK routine is used to toggle 256 byte double density disks between CP/M 1.4 and CP/M 2.0. The differences between the two modes only affects files which are larger than 16K bytes long. There are no real operating instructions and the code is self explanatory.

4.0 SOFTWARE FEATURES

This section describes the features and the assumptions made in the following routines:

- DISK1 ROM / BOOT
- DISK1 CBIOS
- DISK1 FORMAT
- COPY / CONVERT Programs

The above programs assure a disk which has been formatted by the DISK1 FORMAT utility. The DISK1 FORMAT utility always formats side 0 cylinder 0 in 128 bytes sectors, FM, 26 sectors per track. The rest of the disk is formatted in one of the following optional formats:

- 128 byte sectors, FM, 26 sectors per track.
- 256 byte sectors, MFM, 26 sectors per track.
- 512 byte sectors, MFM, 15 sectors per track.
- 1024 byte sectors, MFM, 8 sectors per track.

Also note that due to the size of the DISK1 CBIOS; a CP/M System disk must be recorded in one of the MFM modes.

4.1 CompuPro DISK1 ROM / BOOT

The CompuPro DISK1 ROM code must reside within 256 bytes and contain no memory data references. The ROM code is therefore straight line code performing the following functions:

- Reads the first four sectors (BOOT program).
- Disables the ROM.
- Begin executing the BOOT program.

The CompuPro DISK1 ROM will probably not have to be modified while the BOOT code will vary depending upon the memory configuration.

The BOOT program resides in the first four sectors of the disk (512 bytes). It is read by the ROM code into location 0100h and performs the following functions:

- Reads in the CBIOS.
- Jumps to the Cold Start entry of the CBIOS.

The CBIOS resides in the 5th thru 26th sectors of the disk. The code for the ROM and BOOT is very similar and assumes FM disk read (side 0 head 0 is always FM).

The supplied MOVCPM utility program performs the relocation of the BOOT code depending upon the memory size specification.

4.2 CompuPro DISK1 CBIOS

The CompuPro DISK1 CP/M CBIOS implements all of the jump vectors described in the CP/M 2.2 Alteration Guide. An additional jump vector was added to allow for the extended addressing available on the IEEE 696 bus.

Since the CompuPro CBIOS uses Blocker / Deblocker code to handle the MFM recorded disks; assumptions are required because CP/M 2.2 does not inform the CBIOS when to flush the disk buffer. Floppy disks are removable media and before a disk can be safely removed the disk buffer must be flushed. Before a change of disk is performed one of the following sequences must be performed:

1. A warm start.
2. Closing all files and a disk reset.
3. Call the CBIOS HOME routine.

Failure to perform one of the above sequences might result in disk data being written on the newly inserted disk.

4.2.1 Cold Start

The Cold Start code performs the following functions:

- Initialize, Location 3 -- IOBYTE
- Location 4 -- Current Flexible disk
- Outputs a sign-on message.
- Performs the Warm Boot function.

4.2.2 Warm Boot

The Warm Boot code loads the CP/M CCP and BDOS modules and transfer control to the CCP.

4.2.3 Unit Record

The Standard Unit Record Input / Output routines are:

Console Status.	List Output.
Console Input.	List Status.
Console Output.	Punch Output.
	Reader Input.

The code included in the CompuPro DISK1 CBIOS assumes the purchaser has a CompuPro Interfacer 1. The Console ports are 0 and 1 while the reader, punch, and list are ports 2 and 3. Other serial Input / Output controllers are quite similar and require very little modifications, however parallel or memory mapped video is a different subject and extensive modifications might be required.

4.2.4 Home

The HOME routine flushes the BLOCKER / DEBLOCKER buffer and sets the selected track to zero (0).

4.2.5 Select Disk

The Select Disk routine performs the following functions:

- Checks for Valid drive selection.
- Computes the specified drive's DPH address.
- If this is a "first time call" then the disk type is determined.

4.2.6 SET Track

Sets the selected track to the specified value for a subsequent disk transfer.

4.2.7 Set Sector

Sets the selected sector to the specified value for a subsequent disk transfer.

4.2.8 Set Disk Memory Address

Sets the disk memory buffer address for the next disk transfer.

4.2.9 Set Extended Address

This routine allows the user to set the high order byte of the DMA address. This allows the CP/M user to assess the entire memory address range available on the IEEE 696 bus.

4.2.10 Set Number of Sectors

This routine allows the user to access more than one sector on a track at a time. This feature is only useful for individuals who perform direct transfers with the CompuPro CBIOS.

4.2.11 READ from disk

A 128 byte record is transfered from the disk previously specified by the Select Disk, Set Track, and Set Sector routines, into the memory buffer defined by the Set Disk memory address routine.

4.2.12 Write to disk

A 128 byte record is transfered from the memory buffer to the previously selected disk track and sector.

4.2.13 Translate Sector Number

A sector translation function is performed on the specified sector using the translation table specified the the caller.

4.3 DISK1 FORMAT

The DISK1 FORMAT utility has some important features. One of these features is the automatic determination of the number of sides the disk supports. The FORMAT utility will adjust the formatting parameters for a two-sided disk.

Another feature of the DISK1 FORMAT utility is to perform a read verify which gives the user a check on the ability of the disk to retain data. We have found disks which will have read errors at one density but not a another. If a disk has hard or soft errors at all densities, then this could indicate one of the following problems:

1. Disk media is bad.
2. Drive heads are dirty.
3. Drive is out of adjustment.
4. Controller is out of adjustment.

The ability of the FORMAT program to repeat the identical format on another disk saves time when formatting batches of flexible disks.

Some features the purchaser might want to add to the DISK1 FORMAT utility are:

- Formatting System Tracks only.
- Formatting Selected Tracks only.

4.4 COPY / CONVERT Utilities

The COPY routine performs a track by track copy of the source disk to the user specified destination disk. The CONVERT utility is provided to support the user that currently has disks that have been written with controllers that invert the data before written (i.e. Western Digital usage).

Both COPY and CONVERT use the standard CBIOS call formats for reading and writing the data. The only know restriction with these programs is that the format of the disk must be compatible with the NEC definition for FM and MFM.

5.0 SOFTWARE INTERNAL DESIGN

This section describes the general flow and philosophy of the supplied components.

5.1 CompuPro DISK1 ROM / BOOT

The DISK1 ROM and BOOT routines are vary straight forward. They are both straight line code routines with few loops. The important item to remember when reviewing the listing is the following system disk layout:

Cylinder 0, Head 0 - FM recording; 128 byte sectors 1-4 reserved for CompuPro DISK1 BOOT. Sectors 5-26 reserved for CompuPro DISK1 CBIOS

Cylinder 0, Head 1 - MFM recording; 256, 512, or 1024 byte sectors. Sectors 1-n not used.

Cylinder 1,Head 0 - MFM; 256, 512, or 1024 byte sectors. Sectors 1-n reserved for CCP and BDOS

Cylinder 1, Head 1 - MFM; 256, 512, or 1024 byte sectors. Sectors 1-n not used.

The value of 'n' which equals sectors per track is: 26 for 256 byte sectors, 15 for 512 byte sectors, 8 for 1024 byte sectors.

5.2 CompuPro DISK1 CBIOS

The CompuPro DISK1 CBIOS quite literally makes CP/M work. The CompuPro DISK1 CBIOS uses a modified version of the CP/M DEBLOCK routines. The modification stems from the fact that multiple sector formats are supported. All CP/M disk transfers are 128 bytes in length, and a method for combining the 128 byte transfers into a single, transfer is required.

The description of tables and variables used in the CompuPro DISK1 CBIOS are key to its understanding; therefore the following paragraphs describe the key tables and variables.

The DPBASE table contains a CP/M Disk Parameter Header (DHP) for each logical disk drive. The DPH is described in the CP/M 2.0 Alteration Guide, but a few comments here are appropriate. The PDH is used mostly by CP/M , but four entries are of interest to the CompuPro DISK1 CBIOS -- the Translation Table Address, the Disk Parameter Block (DPB) Address, the Check Vector Address, and the Allocation Vector Address. The Check Vector Address and Allocation Vector

Address are mentioned here, because storage within the CompuPro DISK1 CBIOS must be reserved for these CP/M tables.

The Translation Table Address is used to translate CP/M consecutive sectors into software interlaced external sectors. In order to maintain compatibility between single density disks, a six sector interlace table is used; but for other sector formats different tables are used.

The most important entry within the DPH is the Disk Parameter Block. This address points to a table -- one table for each disk type -- which describes the storage characteristics of the associated disk type. The CP/M 2.0 Alteration Guide describes the entries within the DPB, but the Guide does not describe why the CompuPro DISK1 CBIOS has defined the DPB values. The format of the CompuPro DISK1 CBIOS DPBs follows:

Disk type definition blocks for each particular mode. The format of these areas are as follows:

```
 8 bit  =  Disk type code
16 bit  =  Sectors per track
 8 bit  =  Block shift
 8 bit  =  BS mask
 8 bit  =  Extent mask
16 bit  =  Disk size/1024 - 1
16 bit  =  Directory size
16 bit  =  Allocation for directory
16 bit  =  Check area size
16 bit  =  Offset to first track
```

The Sector translation tables contain values which are CP/M compatible or compatible with other popular CP/M implementations.

Now that the description of the tables is complete, the key variables need to be described. The variables contain information about various stages of a disk transfer. The following variables:

```
ACTDSK  --  Disk Selection Value
ACTTRK  --  Track Number
ACTTYP  --  Disk Type
ACTSEC  --  Sector in Track
```

The information about the data contained within the host disk deblocking buffer is contained in a corresponding set of variables called HSTDSK, HSTTYP, HSTTRK, and HSTSEC. The SEKxxx variables contain the initial CP/M data transfer request. The variables might result in an actual physical transfer, or the data requested could be contained in the host disk deblocking buffer. Each time CP/M calls the WRITE routine, the C register contains a value which indicates on of the following:

```
0  --  Write to a allocated block
1  --  Write to the directory
2  --  First write to a previously unallocated block
```

How the CompuPro DISK1 CBIOS uses these values and other variables reduces the number of unnecessary pre-reads.

The following paragraphs describe the differences between the CBIOS described in the CP/M 2.0 Alteration Guide and the CompuPro DISK1 CBIOS implementation. The reader should be especially familiar with Sections 6 and 12.

The SECTRAN routine differs from the Alteration Guide by testing register DE. If register DE is zero, then no sector translation is performed. Otherwise, DE contains the translation table address.

Since the CompuPro DISK1 CBIOS attempts to detect the density and number of sides of a floppy disk contains, the SELDSK diverges significantly from the Alteration Guide. When the SELDSK is invoked by CP/M, the disk selection value is saved. SELDSK calculates the DPH address, and the least significant of DE is tested for zero. If the bit is non-zero, then the disk type is extracted and saved, and the DPH address is returned.

The more complicated process begins when the bit is zero. The SELDSK routine calls TREAD to try and determine the floppy disk type. If TREAD returns with the zero flag set, then the type was determined. SELDSK then computes the appropriate DPB address and initializes the translation table and DPB address in the DPH.

The READ routine appears simplified, but this is a result of modularizing its code. This routine first calls the CHKBKD routine, which checks the disk type for single density floppy. If the disk is a single density floppy, then the READ routine jumps to the physical sector reading routine FINAL. If the disk is not a single density floppy, then the host buffer is filled by the FILL routine, and the appropriate 128 byte sector is moved to the user's buffer.

The WRITE routine is significantly more complicated than the READ routine. This complication results from the attempt to reduce the number of unnecessary pre-reads of the disk.

The WRITE routine calls CHKBKD to determine the disk type. If the disk type is a single density floppy, a jump to the physical WRITE routine is performed. If the disk type is not a single density, then the logic parallels the WRITE routine documented in the Alteration Guide.

CompuPro DISK1 CBIOS is written fairly straight forward, and only two places are "sneaky". The first sneaky portion is in the FILL routine. The FILL routine returns two parameters on the stack: the host buffer address, and the caller's buffer address. The code to place these two addresses into the stack starts with the XTHL instruction. The other "sneaky" code is in SETUP. SETUP pushes the transfer routine address onto the stack so that the EXEC routine jumps directly to the appropriate disk transfer routine.

This concludes the description of the CompuPro DISK1 CBIOS. If the reader wishes more detailed information, he should refer to the listings and source of the supplied CompuPro DISK1 CBIOS.

6.0 CompuPro CBIOS Customization Guide

This portion of the manual describes a step-by-step method for installing and customizing the CompuPro released CBIOS. The following paragraphs and operations must be performed in a sequential manner.

6.1 Creating the CompuPro BOOT.hex File

The CompuPro BOOT.hex file is created by assembling the GBBOOT.asm file. Before assembling two equates must be altered: MSIZE and the controller's base port address. The released configuration has the following default values preset into the source:

```
MSIZE = 20k
Controller Port Numbers = C0h,...,C3h
```

The supplied MOVCPM.com will alter the memory size dependent locations and if the controller's port is set to the default this step can be by-passed.

6.2 Creating the CompuPro GBBIOS.hex file

If alterations to the CompuPro CBIOS are required; then the controller's port and MSIZE values must be set. Alterations to the CompuPro CBIOS are required if the user wishes to support unit record (CONSOLE, LIST, PUNCH and READER) which do not use the CompuPro "bit-banger", System Support 1, Interfacer 1/2, or Interfacer 3/4 Boards. The use of these serial interfaces are included within the released CompuPro CBIOS.

After the alterations to the source file(s) are made then a check to insure the resultant CBIOS is not larger than 1000 bytes is required. If the CBIOS is larger than 1000 bytes then the BIOSLN value must be altered appropriately. In addition, a cell within the MOVCPM.com file must be altered. The altered cell contains the CBIOS length (rounded to next 256 byte page). The location of this cell is 805H and 806H as shown in the following figure.

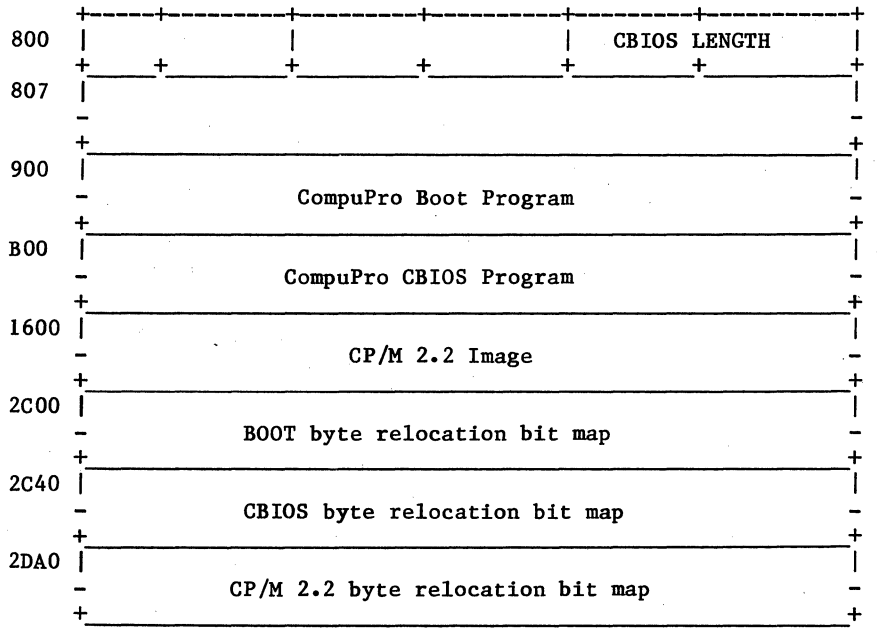
6.3 Understanding the CompuPro MOVCPM.com

The next step in creating the modified CompuPro CP/M 2.2 system is the understanding of how MOVCPM.com works and the key to how MOVCPM.com works is its structure.

The MOVCPM.com file is composed of two parts: a relocating program and a Page Relocatable (PRL) file. The relocating program occupies the first 700h bytes of the MOVCPM.com file and is of little concern. The PRL file on the other hand is of the utmost concern, because it contains not only the binary image of the CP/M system but also the relocating information.

A PRL file allows any byte in the program to have a constant value added to it. This constant is the page offset and is added to the high order byte of the instruction address. That is a sequence of byte C3 25 14 would have the page offset added to the 14 thus "relocating" jump target by a number of pages. In order for the relocating program to know which bytes to add the constant page offset to a bit map is appended to the binary image, one bit for each byte. In addition, there is a header which contains the binary portions length and offset. The header for the CP/M PRL has an additional field indicating the CBIOS's length.

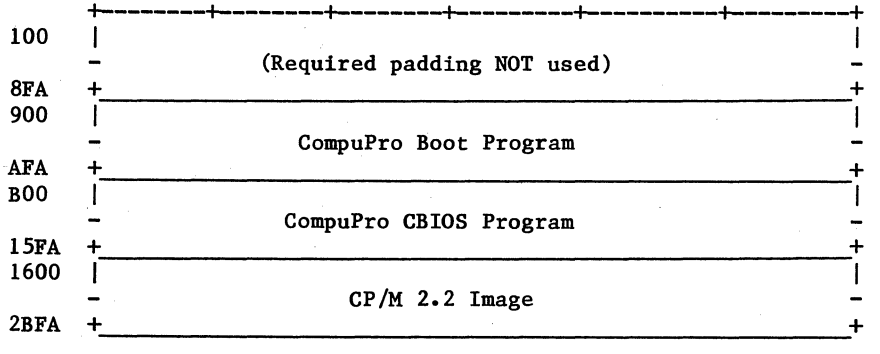
The format of the CP/M PRL portion of the MOVCPM.com file is depicted below:



It is important to note here that the above description applies only to the CompuPro released MOVCPM.com and not to other vendors. The unique way the CompuPro System Disks are layed out requires a unique MOVCPM.com and SYSGEN.com.

6.4 Creating the CPMxx.com file

The creation of the CPMxx.com file is a multi-step operation. The goal of this operation is to create a binary image which can be used by the SYSGEN.com program. The format of this image can be "SAVED" on a disk file for later input to the SYSGEN program. The format of this image is as follows:



In order to create this binary image MOVCPM, DDT, GBBOOT.hex, and GBBIOS.hex will be used. The following paragraphs describe the use of these files to create the required binary image.

The first step in building the binary image is to create a CP/M 2.2 relocated for the appropriate memory size. In the following examples we will illustrate using a 32K CP/M.

```
A>MOVCPM 32 *

CONSTRUCTING 32K CP/M vers 2.2
READY FOR "SYSGEN" OR
"SAVE 43 CPM32.com"
A>SAVE 43 CPM32.com
```

We have now built a standard released CompuPro 32K CP/M 2.2 binary image on the file CPM32.com. The next step is to combine the GBBOOT.hex, GBBIOS.hex, and CPM32.com files into a new binary image. To do this one enters the following:

```
A>DDT CPM32.COM
DDT VERS 2.2
NEXT PC
2C00 0100
-LB00
  0B00    JMP    72B3
  0B03    JMP    72C3
    .       .      .
    .       .      .
    .       .      .
-f100,15FF,0   ---------------> Clear out old image
-iGBBOOT.HEX
-r800      --------------------> Bias of 800h
NEXT PC
2C00 0000
-iGBBIOS.HEX
-h0B00,7000  -----------------> Compute bias
-7B00 9B00
-r9B00

NEXT PC
2C00 0000
```

The next step is to do a quick verify that everything is in its proper place.

```
-L0900   ---------------> list first few locations of BOOT.
  .
  .
  .
-L0B00   ---------------> list first few locations of CBIOS.
  .
  .
  .
```

Now exit and save the binary image on a file as follows:

```
-G0   ------------------> exit via jump to warm boot.
A>SAVE 43 CPM32X.COM
```

The new CP/M binary image is now on the CPM32X.com file and is ready for input into the SYSGEN program as follows:

```
A>SYSGEN CPM32X.com
SYSGEN Version 2.2B

Destination drive name (or RETURN to terminate). B <-- drive #
Destination on B, the type return.
Function complete.  <-- new system disk has now been made
Destination drive name (or RETURN to terminate).  <-- RETURN to exit
```

The system has now been written out on the specified disk and is ready for checkout. Some hints about checking out a new CP/M system. Be sure and check out the disk functions on a test disk before inserting a disk with needed information. Nothing is more upsetting than wiping out a disk by accident.

ALTERATION GUIDE FOR INTERFACER 1/2 I/O ROUTINES

The current version of CP/M-80 has routines that support the serial ports on **INTERFACER 1/2** boards as the CONSOLE, LIST, and UL1 devices. The PUNCH and READER routines are covered through the LIST port. The port addresses for these devices are shown in the table below. Since the **INTERFACER 1/2** boards are configured under hardware control by programming shunts, the baud rates and serial parameters are not specified in the following code section.

DEVICE	PORT BASE	EQUATE
CONSOLE	00H	GBP0:
LIST	02H	GBP1:
READER/PUNCH	02H	GBP1:
UL1: LIST	04H	GBP2:

To alter the base port addresses, simply alter the equate statement value to the desired base.

INTERFACER 1/2 I/O ROUTINES

```
;       CompuPro Interfacer board equates.

GBP0:   EQU     0               ;Serial port zero
GBP1:   EQU     2               ;Serial port one
GBP2:   EQU     4               ;Serial port two
GBDATA: EQU     0               ;Data on even I/O unit
GBSTAT: EQU     1               ;Status on odd I/O unit
GBTBMT: EQU     00000001b       ;Transmit buffer empty
GBDAV:  EQU     00000010b       ;Data available
GBOPT:  EQU     00000100b       ;Optional status line
GBPE:   EQU     00001000b       ;Parity error
GBOR:   EQU     00010000b       ;Overrun error
GBFE:   EQU     00100000b       ;Framing error
GBCC:   EQU     01000000b       ;RS 232 CC input
GBCB:   EQU     10000000b       ;RS 232 CB input
GBRIE:  EQU     00000001b       ;Receiver interrupt enable
GBTIE:  EQU     00000010b       ;Transmitter interrupt enable
GBCD:   EQU     00000100b       ;RS 232 CD output
GBCA:   EQU     00001000b       ;RS 232 CA output
GBTSB:  EQU     00010000b       ;Number of stop bits
GBNP:   EQU     00100000b       ;No parity
GBEPS:  EQU     01000000b       ;Even parity
GBNBI:  EQU     10000000b       ;number of bits/character

;
;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the following
;       values in the A register.
;
;       EXIT    A = 0 (zero), means no character currently ready to read.
;               A = FFh (255), means character currently ready to read.
CONST:  IN      GBP0+GBSTAT     ;Input from port
        ANI     GBDAV           ;Mask data available
        RZ                      ;If data not available
```

```
        ORI     0FFh
        RET
;
;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing the high order
;       bit.  If no character currently ready to read then wait for a character
;       to arrive before returning.
;
;       EXIT    A = character read from terminal.
CONIN:  IN      GBP0+GBSTAT
        ANI     GBDAV
        JZ      CONIN           ;If data not available
        IN      GBP0+GBDATA
        ANI     7Fh
        RET
;
;       C O N S O L E   O U T P U T
;
;       Send a character to the console.  If the console is not ready to receive
;       a character wait until the console is ready.
;
;       ENTRY   C = ASCII character to output to console.
CONOUT: IN      GBP0+GBSTAT
        ANI     GBTBMT
        JZ      CONOUT          ;If transmit buffer not empty
        MOV     A,C
        OUT     GBP0+GBDATA
        RET
;
;       P u n c h   O u t p u t.
;
;       Send a character to the punch device.  If no punch device exists
;       then immediately return.
;
;       ENTRY   C = ASCII character to output.
PUNCH:  IN      GBP1+GBSTAT
        ANI     GBTBMT
        JZ      PUNCH           ;If transmit buffer full
        MOV     A,C
        OUT     GBP1+GBDATA
        RET
;
;       R e a d e r   I n p u t.
;
;       Read the next character from the currently assigned reader device
;       into the A register.
;
;       EXIT    A = character read from the reader device.
READER: IN      GBP1+GBSTAT     ;Input from port
        ANI     GBDAV           ;Mask data available
        JZ      READER          ;If data not available
        IN      GBP1+GBDATA
        RET
;
```

```
;       L i s t   O u t p u t.
;
;       Send a character to the list device.  If the list device is not ready
;       to receive a character wait until the device is ready.
;
;       ENTRY   C = ASCII character to be output.

LIST:   LDA     IOBYTE          ;Get IOBYTE status
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        JZ      UL1
LIST1:  IN      GBP1+GBSTAT     ;Get status
        ANI     GBCC+GBTBMT
        SUI     GBTBMT
        JNZ     LIST1
        MOV     A,C
        OUT     GBP1+GBDATA
        RET
;
;       U L 1 :    L I S T   O U T P U T
;
;       Send a character to the second interfacer
;

UL1:    IN      GBP2+GBSTAT     ;Get status
        ANI     GBCC+GBTBMT
        SUI     GBTBMT
        JNZ     UL1
        MOV     A,C
        OUT     GBP2+GBDATA
        RET
;
;       L i s t   S t a t u s.
;
;       Return the ready status for the list device.
;
;       EXIT    A = 0 (zero), list device is not ready to accept another
;               character.
;               A = FFh (255), list device is ready to accept a character.

LISTST: LDA     IOBYTE
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        JZ      UL1ST
        IN      GBP1+GBSTAT
LSTAT:  ANI     GBCC+GBTBMT
        SUI     GBTBMT
        RZ                      ;If ready
        ORI     0FFh
        RET
UL1ST:  IN      GBP2+GBSTAT
        JMP     LSTAT

        END
```

ALTERATION GUIDE FOR SYSTEM SUPPORT 1 I/O ROUTINES

The current version of CP/M-80 has routines that support the serial port on the **SYSTEM SUPPORT 1** as the CONSOLE device and the serial ports on **INTERFACER 1/2** boards as the LIST, and UL1 devices. The PUNCH and READER routines are covered through the LIST port. The port addresses for these devices are shown in the table below. The **SYSTEM SUPPORT 1** board has a software programmable USART that must be initialized before it can be used. To alter the baud rates and the serial parameters, you must substitute new mode and command words in the sTINIT routine of the CONSOLE INITIALIZATION section below. (Refer to your **SYSTEM SUPPORT 1** manual for these values.) Since the **INTERFACER 1/2** boards are configured under hardware control by programming shunts, refer to the proper manual for modification settings.

Since the CONSOLE is the only device supported on the **SYSTEM SUPPORT 1** board, this code only covers the CONSOLE I/O routines. To alter the LIST, READER/PUNCH, or UL1 code, refer to the **ALTERATION GUIDE FOR INTERFACER 1/2 I/O ROUTINES.**

DEVICE	PORT BASE	EQUATE
CONSOLE	50H	GBSS:
LIST	02H	GBP1:
READER/PUNCH	02H	GBP1:
UL1: LIST	04H	GBP2:

To alter the base port addresses, simply alter the equate statement value to the desired base.

SYSTEM SUPPORT 1 CONSOLE I/O ROUTINE

```
        IF      SYSSUP
;       CompuPro System Support 1 equates.

GBSS:   EQU     50h             ;System Support starting port
GBMP0:  EQU     GBSS+0          ;Master PIC port 0
GBMP1:  EQU     GBSS+1          ;Master PIC port 1
GBSP0:  EQU     GBSS+2          ;Slave PIC port 0
GBSP1:  EQU     GBSS+3          ;Slave PIC port 1
GBT0:   EQU     GBSS+4          ;Timer number 0
GBT1:   EQU     GBSS+5          ;Timer number 1
GBT2:   EQU     GBSS+6          ;Timer number 2
GBTC:   EQU     GBSS+7          ;Timer control port
GBFPPD: EQU     GBSS+8          ;Floating point processor data port
GBFPPC: EQU     GBSS+9          ;Floating point processor command port
GBCLKC: EQU     GBSS+10         ;Clock command port
GBCLKD: EQU     GBSS+11         ;Clock data port
GBUD:   EQU     GBSS+12         ;Uart data port
GBUS:   EQU     GBSS+13         ;Uart status port
GBUM:   EQU     GBSS+14         ;Uart modem port
GBUC:   EQU     GBSS+15         ;Uart command port
SSDAV:  EQU     00000010b       ;System Support Data Available
SSTBMT: EQU     00000001b       ;System Support Transmit Buffer Empty
;
;       C O N S O L E   I N I T I A L I Z A T I O N
;
;       This routine performs the initialization required by
;       the System Support USART.
```

```
;
sTINIT:
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, even, 2 stops
        OUT     GBUM            ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBUM            ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBUC            ;Set up command port
        RET
;
;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the following
;       values in the A register.
;
;       EXIT    A = 0 (zero), means no character currently ready to read.
;               A = FFh (255), means character currently ready to read.
sCONST:
        IN      GBUS            ;Input from port
        ANI     SSDAV           ;Mask data available
        RZ                      ;If data not available
        ORI     0FFH
        RET
;
;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing the high order
;       bit.  If no character currently ready to read then wait for a character
;       to arrive before returning.
;
;       EXIT    A = character read from terminal.
sCONIN:
        IN      GBUS            ;Get status from uart
        ANI     SSDAV
        JZ      sCONIN
        IN      GBUD
        ANI     7Fh
        RET
;
;       C O N S O L E   O U T P U T
;
;       Send a character to the console.  If the console is not ready to receive
;       a character wait until the console is ready.
;
;       ENTRY   C = ASCII character to output to console.
sCONOUT:
        IN      GBUS            ;Get uart status
        ANI     SSTBMT          ;Test if buffer empty
        JZ      sCONOUT
        MOV     A,C
        OUT     GBUD
        RET

        ENDIF
```

ALTERATION GUIDE FOR INTERFACER 3/4 I/O ROUTINES

The current version of CP/M-80 has routines that support the serial ports on the **INTEFACER 3/4** as the CONSOLE, LIST, and UL1 devices. The PUNCH and READER routines are covered through the LIST port of the **INTEFACER 1/2** code. The port addresses for the **INTERFACER 3/4** are 10H-17H. The USER number for these devices are shown in the table below. The **INTERFACER 3/4** board has software programmable USARTs that must be initialized before it can be used. To alter the baud rates and the serial parameters, you must substitute new mode and command words in the I3INIT routine of the CONSOLE INITIALIZATION section below. (Refer to your **INTERFACER 3/4** manual for these values.)

DEVICE	PORT BASE	USER NUMBER	EQUATE
CONSOLE	10H	#7	CON:
LIST	10H	#6	PRN:
READER/PUNCH	02H	NONE	GBP1:
UL1: LIST	10H	#5	ULS:

To alter the base port addresses, simply alter the GBI3: equate statement value to the desired base. To alter the USER NUMBER, simply alter the CON:, PRN:, or ULS: equate statement to the desired USER.

INTERFACER 3/4 CONSOLE I/O ROUTINE

```
        IF      INTER3
;       CompuPro Interfacer 3/4 support routines
GBI3:   EQU     10h             ;Interfacer 3/4 Base address
GBI3D:  EQU     GBI3+0          ;Uart data location
GBI3S:  EQU     GBI3+1          ;Uart status
GBI3M:  EQU     GBI3+2          ;Uart mode register
GBI3C:  EQU     GBI3+3          ;Uart command register
GBI3U:  EQU     GBI3+7          ;User select register
GBI3DV: EQU     00000010b       ;Interfacer 3/4 Data Available
GBI3MT: EQU     00000001b       ;Interfacer 3/4 Transmit Buffer Empty
GBI3DS: EQU     10000000b       ;Interfacer 3/4 Data Set Ready
CON:    EQU     7               ;Interfacer 3/4 Console Select
PRN:    EQU     6               ;Interfacer 3/4 Printer Select
ULS:    EQU     5               ;Interfacer 3/4 UL1 Select
;
;       C O N S O L E   I N I T I A L I Z A T I O N
;
;       This routine performs the initialization required by the Interfacer 3/4.
;
I3INIT:
        MVI     A,CON           ;Console select
        OUT     GBI3U           ;Select Uart 7
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port
        MVI     A,PRN           ;Printer Select
        OUT     GBI3U           ;Select Uart 0
```

```
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port
        MVI     A,ULS           ;User list 1 Select
        OUT     GBI3U           ;Select Uart 0
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port
        RET
;
;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the following
;       values in the A register.
;
;       EXIT    A = 0 (zero), means no character currently ready to read.
;               A = FFh (255), means character currently ready to read.
I3CONST:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Input from port
        ANI     GBI3DV          ;Mask data available
        RZ                      ;If data not available
        ORI     0FFH
        RET
;
;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing the high order
;       bit.  If no character currently ready to read then wait for a character
;       to arrive before returning.
;
;       EXIT    A = character read from terminal.
I3CONIN:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Get status from uart
        ANI     GBI3DV
        JZ      I3CONIN
        IN      GBI3D
        ANI     7Fh
        RET
;
;       C O N S O L E   O U T P U T
;
;       Send a character to the console.  If the console is not ready to receive
;       a character wait until the console is ready.
```

```
;
;       ENTRY   C = ASCII character to output to console.
I3CONOUT:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Get uart status
        ANI     GBI3MT          ;Test if buffer empty
        JZ      I3CONOUT
        MOV     A,C
        OUT     GBI3D
        RET
;
;       L i s t   O u t p u t.
;
;       Send a character to the list device.  If the list device is not ready
;       to receive a character wait until the device is ready.
;
;       ENTRY   C = ASCII character to be output.
I3LIST: LDA     IOBYTE          ;Get IOBYTE status
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        MVI     A,ULS
        JZ      I3UL1
        MVI     A,PRN
I3UL1:  OUT     GBI3U
I3LST1: IN      GBI3S
        ANI     GBI3MT+GBI3DS
        SUI     GBI3MT+GBI3DS
        JNZ     I3LST1
        MOV     A,C
        OUT     GBI3D
        RET
;
;       L i s t   S t a t u s.
;
;       Return the ready status for the list device.
;
;       EXIT    A = 0 (zero), list device is not ready to accept another charac.
;               A = FFh (255), list device is ready to accept a character.
I3LST:  LDA     IOBYTE
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        MVI     A,ULS
        JZ      I3LS1
        MVI     A,PRN
I3LS1:  OUT     GBI3U
        IN      GBI3S
        ANI     GBI3MT+GBI3DS
        SUI     GBI3MT+GBI3DS
        MVI     A,0FFH
        RZ
        XRA     A
        RET

        ENDIF
```

DISCLAIMER

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

```
;CBIOS - Sorcim CompuPro CP/M 2.2 CBIOS.
;
;       +----------------------+
;       |                      |
;       |      C B I O S       |
;       |                      |
;       +----------------------+
;
;
;       CompuPro                        Sorcim Corp.
;       Oakland, CA                     Santa Clara, CA
;
;       This product is a copyright program product of
;       Sorcim Corp. and is supplied for use with the CompuPro
;       Disk controllers.
;
;       Version number: 2.2K
; 80-10-27      Ver 2.2A
; 81-03-05      Ver 2.2E        Seek fixes. dwd.
; 81-03-30      Ver 2.2F        Add Bit Banger.  phm.
; 81-04-25      Ver 2.2G        Add System Support, two list devs. bdg.
; 81-05-21      Ver 2.2H        Add flushing fixes. dwd
; 81-09-08      Ver 2.2J        Add Interfacer 3 support routines. bdg.
; 81-10-26      Ver 2.2K        Add list device for interfacer 3. bdg.
;                                       Fix deblocking in SETTRK
;
;       The following code is supplied to customers who
;       purchase a hard/floppy disk system from CompuPro.
;       The intent of the following code is to illustrate
;       how to create a CBIOS for the user supplied
;+      CP/M 2.2. Lines of code beginning with ";+" were
;+      modified for assembly with Digital Research's ASM
;+      assembler. Syntax changes and comments can be found
;+      near the modified lines.
;
;       CompuPro IEEE 696 Floppy Disk Controller.
;       CompuPro IEEE 696 chassis and motherboards.
;       CompuPro IEEE 696 Dual Processor board.
;       CompuPro IEEE 696 RAM 20 boards (2).
;       CompuPro IEEE 696 Interfacer I (assigned port 0 thru 3).

OPARM:  EQU     32          ;Memory size in Kbytes, or Bios load address

VERS:   EQU     22
CBIOSV: EQU     11          ;CBIOS revision level (2.2K)
```

```
FALSE:  EQU     0
TRUE:   EQU     NOT FALSE
Z80:    EQU     FALSE           ;Set to true if processor is Z80.
C8080:  EQU     not Z80         ;Otherwise processor is 8080 type.
HARD:   EQU     FALSE           ;Set to true if hard disk code
BANG:   EQU     TRUE            ;False to eliminate Bit Banger.
SYSSUP: EQU     TRUE            ;Set to true to include System Support code.
INTER3: EQU     TRUE            ;Set to true to include Interfacer 3 support

BIOSLN: EQU     01000H          ;Bios length

CR:     EQU     0Dh
LF:     EQU     0Ah
DELCNT: EQU     5*1000          ;Delay count for 5 Mhz CPU


K:      EQU     1024
;+     IF      OPARM < (64+2)  ;If absolute
option: set     false
        if      oparm/(64+1)
option: set     true
        endif
        if      not option

MSIZE:  EQU     OPARM           ;Size of CP/M memory
BIOS:   EQU     MSIZE*K-BIOSLN  ;Start of CP/M jump table
LWAMEM: EQU     MSIZE*K-1
        ENDIF
;+     IF      OPARM > (64+1)  ;If PRL generation
        if      option

MSIZE:  EQU     (OPARM+BIOSLN)/K        ;Size of CP/M memory
BIOS:   EQU     OPARM           ;Start of CP/M jump table
LWAMEM: EQU     OPARM+BIOSLN-1
        ENDIF
BDOS:   EQU     BIOS-0E00h+6    ;BDOS entry point
CCP:    EQU     BIOS-1600h


;       Page Zero Definitions.

IOBYTE: EQU     3               ;Location of IOBYTE
CDISK:  EQU     4               ;Location of current disk
BIORAM: EQU     40h             ;16 ram cells
OPTS:   EQU     BIORAM          ;GBC DISK1 board switch options
;               BIORAM+1        ;GBC (cell before TICK)
TICK:   EQU     BIORAM+2        ;GBC Sample period
DBUF:   EQU     80h             ;Default sector buffer


;       CP/M to host disk constants

HSTSIZ: EQU     1024            ;Blocking/Deblocking buffer size
CPMSIB: EQU     1024/128        ;Standard sectors in block
FPYSIB: EQU     2048/128        ;Sectors in floppy disk block
```

```
;       CP/M disk control block equates which define the
;       disk types and maximum storage capability of each
;       disk type.

DSKS1:  EQU     0       ;Single density, single sided.
DSKS2:  EQU     1       ;Single density, double sided.
DSKD1:  EQU     2       ;Double density, single sided.
DSKD2:  EQU     3       ;Double density, double sided.
DSKD3:  EQU     4       ;Double density, single sided
DSKD4:  EQU     5       ;Double density, double sided
DSKD5:  EQU     6       ;Double density, single sided
DSKD6:  EQU     7       ;Double density, double sided
MAXFTP: EQU     DSKD6   ;Maximum floppy type
DSK8S1: EQU     8       ;SA 1002 and first half SA 1004
DSK8S2: EQU     9       ;last half SA 1004.
DSK8M1: EQU     10      ;First half of Memorex 8 inch drive
DSK8M2: EQU     11      ;Last half of Memorex 8 inch drive

S1DSM:  EQU     ((77-2)*26)/CPMSIB
S2DSM:  EQU     ((77-2)*2*26)/FPYSIB
D1DSM:  EQU     ((77-2)*2*26)/FPYSIB
D2DSM:  EQU     ((77-2)*2*2*26)/FPYSIB
D3DSM:  EQU     ((77-2)*4*15)/FPYSIB
D4DSM:  EQU     ((77-2)*2*4*15)/FPYSIB
D5DSM:  EQU     ((77-2)*8*8)/FPYSIB
D6DSM:  EQU     ((77-2)*2*8*8)/FPYSIB


;       BDOS constants on entry to write

WRALL:  EQU     0               ;write to allocated
WRDIR:  EQU     1               ;write to directory
WRUAL:  EQU     2               ;write to unallocated

;       Macro for generating Control Blocks for disk drives
;       The format of these disk control blocks are as follows:
;       16 bits = -> translation table.
;       48 bits = Work area for CP/M.
;       16 bits = -> DIRBUF.
;       16 bits = -> Parameter block.
;       16 bits = -> check vector.
;       16 bits = -> allocation vector.

NDSK:   SET     0               ;Number of disk drives
NOHRD:  SET     0               ;Number of hard disk drives
NOFDD:  SET     0               ;Number of floppy disk drives
ALVS:   SET     0               ;Allocation vector size
CSVS:   SET     0               ;Check vector size

;       The following jump table defines the entry points
;       into the CBIOS for use by CP/M and other external
;       routines; therfore the order of these jump cannot
;       be modified.  The location of these jumps can only
;       be modified by 400h locations, which is a restriction
;       of MOVCPM.
```

```
        ORG     BIOS

        JMP     CBOOT           ;Cold boot
        JMP     WBOOT           ;Warm boot
J$CST   JMP     CONST           ;Console status (input)
J$CIN   JMP     CONIN           ;Console input
J$COUT  JMP     CONOUT          ;Console output
J$LIST  JMP     LIST            ;List output
        JMP     PUNCH           ;Punch output
        JMP     READER          ;Reader input
        JMP     HOME            ;Set track to zero
        JMP     SELDSK          ;Select disk unit
        JMP     SETTRK          ;Set track
        JMP     SETSEC          ;Set sector
        JMP     SETDMA          ;Set Disk Memory Address
        JMP     READ            ;Read from disk
        JMP     WRITE           ;Write onto disk
J$LST   JMP     LISTST          ;List status (output)
        JMP     SECTRN          ;Translate sector number
        JMP     SETNUM          ;Set number of sectors to read
        JMP     SETXAD          ;Set extended address bank.

;       Extended bank addresses.

CXADR:  EQU     00h             ;CP/M Bank
BXADR:  EQU     00h             ;BIOS Bank


;       Disk Input / Output port assignments.

FDPORT  EQU     0C0h            ;Base port address for Controller
FDCS    EQU     FDPORT          ;Status register
FDCD    EQU     FDPORT+1        ;Data register
DMA     EQU     FDPORT+2        ;Dma address (when write)
INTS    EQU     FDPORT+2        ;Status Register (when read)
SER     EQU     FDPORT+3        ;Serial port


;       Controller function definitions

F$SPEC  equ     03      ;Specify
F$DSTS  equ     04      ;Drive status
F$WRT   equ     05      ;Write data
F$RDAT  equ     06      ;Read data
F$RECA  equ     07      ;recalibrate
F$RSTS  equ     08      ;Read status
F$DRID  equ     10      ;Read ID
F$SEEK  equ     15      ;Seek


;       Disk drive constants
;
;       To alter step rate times from the selected 8 ms to a
;       lower rate, simply comment out (";") the 8 ms rate and
```

```
;       remove the ";" from in front of the faster rate.
;       8 ms step rate is selected as shipped.

STEPR:  equ     8               ;Shugart SA 800 or other 8 ms drive

;       3 ms step rate for QUME and other drives with 3 ms step
;       rates should be set for 4 ms due to problem in 8272 and
;       765A controller chip.

;STEPR: equ     4               ;4 ms step rate
SRT:    equ     16-STEPR        ;Controller value
HUT:    equ     240/16          ;Head unload = 240 ms
HDLT:   equ     35              ;Head load = 35 ms
ND:     equ     0b              ;Set DMA mode
;
;                  7   6   5   4   3   2   1   0
;               +---+---+---+---+---+---+---+---+
;       ST0     |    IC   | SE | EC | NR | HD |    US   |
;               +---+---+---+---+---+---+---+---+
;       ST1     | EN | 0 | DE | OR | 0 | ND | NW | MA |
;               +---+---+---+---+---+---+---+---+
;       ST2     | 0 | CM | DD | WC | SH | SN | BC | MD |
;               +---+---+---+---+---+---+---+---+
;       ST3     | FT | WP | RY | T0 | TS | HD |    US   |
;               +---+---+---+---+---+---+---+---+
;
;       IC - Interrupt code.
;               00 - Normal termination by TC signal.
;               01 - Abnormal termination.
;               10 - Invalid command.
;               11 - Abnormal termination (READY dropped).
;
;       SE - Seek end, inticates end of seek.
;
;       EC - Equipment Check.
;
;       NR - Not ready.
;
;       HD - State of the head select.
;
;       US - State of the unit select.
;
;       EN - End of Cylinder, Read EOT sector.
;
;       DE - CRC error in ID or data fields.
;
;       OR - Over run.
;
;       ND - No Data.
;
;       NW - Not writable (write protect detected)
;
;       MA - Missing address mark.
;
;       CM - Control Mark (deleted data address mark).
```

```
;
;       DD - CRC error in data field.
;
;       WC - Wrong cylinder.
;
;       SH - Scan equal hit.
;
;       SN - Scan not satisfied.
;
;       BC - Bad cylinder.
;
;       MD - Missing data mark.
;
;       FT - Fault.
;
;       WP - Write protect signal.
;
;       RY - Ready.
;
;       TO - Track zero.
;
;       TS - Two sided disk is inserted.
;
;       Morrow Designs Hard Disk Controller Equates.

HDPORT: EQU     050h            ;Base port address
HDSTAT: EQU     HDPORT+0        ;Status and control
HDCMD:  EQU     HDPORT+1        ;Command channel
HDFNC:  EQU     HDPORT+2        ;Function
HDDATA: EQU     HDPORT+3        ;Data port

;       IN  HDSTAT          7   6   5   4   3   2   1   0
;                           |   |   |   |   |   |   |   |
;       Halt <--------------+   |   |   |   |   |   |   |
;       ILEVEL <----------------+   |   |   |   |   |   |
;       NREADY <--------------------+   |   |   |   |   |
;       NFAULT <------------------------+   |   |   |   |
;       TIMOUT <----------------------------+   |   |   |
;       COMPLT <--------------------------------+   |   |
;       OPDONE <------------------------------------+   |
;       TRACK0 <----------------------------------------+
;
;       IN  HDCMD           7   6   5   4   3   2   1   0
;                                           |   |   |   |
;       R1 <--------------------------------+   |   |   |
;       R0 <------------------------------------+
;       RETRY <-----------------------------------+   |
;       SDONE <---------------------------------------+
;
;       OUT  HDSTAT         7   6   5   4   3   2   1   0
;                                           |   |   |   |
;       WPROT <-----------------------------+   |   |   |
;       DSKCLK <----------------------------------+   |   |
;       RUN <-----------------------------------------+   |
;       FRENBL <------------------------------------------+
```

```
;
;       OUT  HDFNC          7   6   5   4   3   2   1   0
;                           |   |   |   |   |   |   |   |
;       NHEAD <-------------+---+---+---+   |   |   |   |
;       DIR   <-----------------------------+   |   |   |
;       NSTEP <---------------------------------+   |   |
;       DRIVE <-------------------------------------+---+

;       Hard disk commands  (OUT  HDCMD).

H$RBDT: EQU     00h             ;reset buffer pointer (data)
H$RSDT: EQU     01h             ;Read sector data
H$RSHD: EQU     03h             ;Read sector header
H$WSDT: EQU     05h             ;Write sector data
H$WSHD: EQU     07h             ;Write sector header
H$RHPT: EQU     08h             ;Reset header pointer
        ENDIF
;
STEPMS: DB      (SRT SHL 4)+0

;       Sector Translation Tables.

XTABLE: DW      XLTS            ;Single 128
        DW      XLTD1           ;Double 256
        DW      XLTD2           ;Double 512
        DW      XLTD3           ;Double 1024

XLTS:   DB      0,6,12,18,24,4,10,16,22,2,8,14,20
        DB      1,7,13,19,25,5,11,17,23,3,9,15,21

XLTD1:  DB       0, 1,18,19,36,37, 2, 3,20,21,38,39
        DB       4, 5,22,23,40,41, 6, 7,24,25,42,43
        DB       8, 9,26,27,44,45,10,11,28,29,46,47
        DB      12,13,30,31,48,49,14,15,32,33,50,51
        DB      16,17,34,35

XLTD2:  DB       0, 1, 2, 3,16,17,18,19
        DB      32,33,34,35,48,49,50,51
        DB       4, 5, 6, 7,20,21,22,23
        DB      36,37,38,39,52,53,54,55
        DB       8, 9,10,11,24,25,26,27
        DB      40,41,42,43,56,57,58,59
        DB      12,13,14,15,28,29,30,31
        DB      44,45,46,47

XLTD3:  DB       0, 1, 2, 3, 4, 5, 6, 7
        DB      24,25,26,27,28,29,30,31
        DB      48,49,50,51,52,53,54,55
        DB       8, 9,10,11,12,13,14,15
        DB      32,33,34,35,36,37,38,39
        DB      56,57,58,59,60,61,62,63
        DB      16,17,18,19,20,21,22,23
        DB      40,41,42,43,44,45,46,47
```

```
;       Disk selection masks.
;               A   B   C   D   E   F   G   H
DSKMSK: DB      00h,01h,02h,03h,00h,00h,00h,00h,00h
;               I   J   K   L   M   N   O   P
;       Control Blocks for disk drives

DPBASE:
ndsk    set     ndsk+1
        dw      xlts,0,0,0,dirbuf,dpbs1+1,csv+csvs,alv+alvs
nofdd   set     nofdd+1
csvs    set     csvs+(256/4)
alvs    set     alvs+((d6dsm+7)/8)
ndsk    set     ndsk+1
        dw      xlts,0,0,0,dirbuf,dpbs1+1,csv+csvs,alv+alvs
nofdd   set     nofdd+1
csvs    set     csvs+(256/4)
alvs    set     alvs+((d6dsm+7)/8)
ndsk    set     ndsk+1
        dw      xlts,0,0,0,dirbuf,dpbs1+1,csv+csvs,alv+alvs
nofdd   set     nofdd+1
csvs    set     csvs+(256/4)
alvs    set     alvs+((d6dsm+7)/8)
ndsk    set     ndsk+1
        dw      xlts,0,0,0,dirbuf,dpbs1+1,csv+csvs,alv+alvs
nofdd   set     nofdd+1
csvs    set     csvs+(256/4)
alvs    set     alvs+((d6dsm+7)/8)


        IF      HARD

        ENDIF
;       Disk type definition blocks for each particular mode.

DPBS1:          ;Single density, single sided.

        db      dsks1
        dw      26
        db      3,7,0
        dw      s1dsm-1,64-1
        db      11000000b,00000000b
        dw      (64+3)/4
        dw      2

DPBS2:          ;Single density, double sided.

        db      dsks2
        dw      26
        db      4,15,1
        dw      s2dsm-1,128-1
        db      11000000b,00000000b
        dw      (128+3)/4
        dw      2*2
```

```
DPBD1:          ;Double density, single sided.

        db      dskd1
        dw      2*26
        db      4,15,0
        dw      d1dsm-1,128-1
        db      11000000b,00000000b
        dw      (128+3)/4
        dw      2

DPBD2:          ;Double density, double sided.

        db      dskd2
        dw      2*26
        db      4,15,0
        dw      d2dsm-1,256-1
        db      11110000b,00000000b
        dw      (256+3)/4
        dw      2*2

DPBD3:          ;Double density, single sided.

        db      dskd3
        dw      4*15
        db      4,15,0
        dw      d3dsm-1,128-1
        db      11000000b,00000000b
        dw      (128+3)/4
        dw      2

DPBD4:          ;Double density, double sided.

        db      dskd4
        dw      4*15
        db      4,15,0
        dw      d4dsm-1,256-1
        db      11110000b,00000000b
        dw      (256+3)/4
        dw      2*2

DPBD5:          ;Double density, single sided.

        db      dskd5
        dw      8*8
        db      4,15,0
        dw      d5dsm-1,128-1
        db      11000000b,00000000b
        dw      (128+3)/4
        dw      2

DPBD6:          ;Double density, double sided.

        db      dskd6
        dw      8*8
        db      4,15,0
```

```
        dw      d6dsm-1,256-1
        db      11110000b,00000000b
        dw      (256+3)/4
        dw      2*2

        IF      HARD
DPBS81:         ;Shugart SA 1000, first half.
DPBS82:         ;Shugart Sa 1000, last half.

DPBM81:         ;Memorex 8 inch. first half.

DPBM82:         ;Memorex 8 inch. last half.

        ENDIF

;       CompuPro Interfacer board equates.

GBP0:   EQU     0               ;Serial port zero
GBP1:   EQU     2               ;Serial port one
GBP2:   EQU     4               ;Serial port two
GBDATA: EQU     0               ;Data on even I/O unit
GBSTAT: EQU     1               ;Status on odd I/O unit
GBTBMT: EQU     00000001b       ;Transmit buffer empty
GBDAV:  EQU     00000010b       ;Data available
GBOPT:  EQU     00000100b       ;Optional status line
GBPE:   EQU     00001000b       ;Parity error
GBOR:   EQU     00010000b       ;Overrun error
GBFE:   EQU     00100000b       ;Framing error
GBCC:   EQU     01000000b       ;RS 232 CC input
GBCB:   EQU     10000000b       ;RS 232 CB input
GBRIE:  EQU     00000001b       ;Receiver interrupt enable
GBTIE:  EQU     00000010b       ;Transmitter interrupt enable
GBCD:   EQU     00000100b       ;RS 232 CD output
GBCA:   EQU     00001000b       ;RS 232 CA output
GBTSB:  EQU     00010000b       ;Number of stop bits
GBNP:   EQU     00100000b       ;No parity
GBEPS:  EQU     01000000b       ;Even parity
GBNBI:  EQU     10000000b       ;number of bits/character
;
;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the
;       following values in the A register.
;
;       EXIT    A = 0 (zero), means no character
;               currently ready to read.
;
;               A = FFh (255), means character
;               currently ready to read.

CONST:  IN      GBP0+GBSTAT     ;Input from port
        ANI     GBDAV           ;Mask data available
        RZ                      ;If data not available
        ORI     0FFh
        RET
```

```
;
;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing
;       the high order bit.  If no character currently ready to
;       read then wait for a character to arrive before returning.
;
;       EXIT    A = character read from terminal.

CONIN:  IN      GBP0+GBSTAT
        ANI     GBDAV
        JZ      CONIN           ;If data not available
        IN      GBP0+GBDATA
        ANI     7Fh
        RET
;
;       C O N S O L E   O U T P U T
;
;       Send a character to the console.  If the console
;       is not ready to receive a character wait until
;       the console is ready.
;
;       ENTRY   C = ASCII character to output to console.

CONOUT: IN      GBP0+GBSTAT
        ANI     GBTBMT
        JZ      CONOUT          ;If transmit buffer not empty
        MOV     A,C
        OUT     GBP0+GBDATA
        RET
;
;       P u n c h   O u t p u t.
;
;       Send a character to the punch device.  If no punch
;       device exists then immediately return.
;
;       ENTRY   C = ASCII character to output.

PUNCH:  IN      GBP1+GBSTAT
        ANI     GBTBMT
        JZ      PUNCH           ;If transmit buffer full
        MOV     A,C
        OUT     GBP1+GBDATA
        RET
;
;       R e a d e r   I n p u t.
;
;       Read the next character from the currently assigned
;       reader device into the A register.
;
;       EXIT    A = character read from the reader device.

READER: IN      GBP1+GBSTAT     ;Input from port
        ANI     GBDAV           ;Mask data available
        JZ      READER          ;If data not available
```

```
        IN      GBP1+GBDATA
        RET
;
;       L i s t   O u t p u t.
;
;       Send a character to the list device.  If the list
;       device is not ready to receive a character wait
;       until the device is ready.
;
;       ENTRY   C = ASCII character to be output.

LIST:   LDA     IOBYTE          ;Get IOBYTE status
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        JZ      UL1
LIST1:  IN      GBP1+GBSTAT     ;Get status
        ANI     GBCC+GBTBMT
        SUI     GBTBMT
        JNZ     LIST1
        MOV     A,C
        OUT     GBP1+GBDATA
        RET
;
;       U L 1 :    L I S T   O U T P U T
;
;       Send a character to the second interfacer
;

UL1:    IN      GBP2+GBSTAT     ;Get status
        ANI     GBCC+GBTBMT
        SUI     GBTBMT
        JNZ     UL1
        MOV     A,C
        OUT     GBP2+GBDATA
        RET
;
;       L i s t   S t a t u s.
;
;       Return the ready status for the list device.
;
;       EXIT    A = 0 (zero), list device is not ready to
;               accept another character.
;               A = FFh (255), list device is ready to accept
;               a character.

LISTST: LDA     IOBYTE
        ANI     0C0H            ;Ceck for UL1:
        SUI     0C0H
        JZ      UL1ST
        IN      GBP1+GBSTAT
LSTAT:  ANI     GBCC+GBTBMT
        SUI     GBTBMT
        RZ                      ;If ready
        ORI     0FFh
        RET
```

```
UL1ST:  IN      GBP2+GBSTAT
        JMP     LSTAT

        IF      BANG
;
;
;+      TITLE   'Bit Banger for CompuPro DISK1'
;       Bit Banger for CompuPro DISK1.
;
; 81-03-28      phm


;       7 Data bits, no parity
;       For 2 MHz 8085, use 300 Baud.
;       For 5 MHz,      use 600 Baud.
;       Normal bit timings.
;                                   ___ TICK ___
;       Baud    uSec/bit            2MHz    5MHz
;       4800    208                 <<<     25
;       2400    416                 18      46
;       1200    833                 38      92
;       600     1667                75      216
;       300     3333                102     >>>
;       110     can't               >>>     >>>
;
;       Restriction on higher rates is due to the fact
;       that we can only adjust speed by integral TICK
;       counts: Unless number > 100, cannot tune in closer
;       than one percent.


;       Timing for the Bit Banger is via programmed delays,
;       so be very careful if changes are contemplated.
;       Baud =  Bit per second of serial data.
;       Cycle = Machine cycle (clock period, T-State).
;       Tick =  1/n of a bit time (PERIOD of sample).
;
;       1200 baud       = 833 uSec/bit
;       * 2 MHz         = 1667 cycles/bit
;       / 16 sample rate= 104  cycles/Tic

yBANG:  EQU     SER     ;Serial bit latch port


samp:   db      0ffh
        DS      10      ;buckets for Space counts
        db      0ffh
;
;       CONSOLE STATUS
;       BitBanger has no status available, so always
;               replies NO.  This means that Ctrl-S will
;               not work when using the BitBanger.
;       Exit    A= FFh  means character available.
```

```
KONST:
        xra     a
        RET
;
;       Output 1 Character.
;       Entry   C= Character to output.
;               Line assumed marking.
;       Exit    Line marking, but stop time not elapsed.
;       Transmission format:
;               Data bits inverted;
;               Start(0), D0, D1, ..., D6, Stop(1), Leaves marking
;       Note: Cannot destroy DE or HL.
;       Uses    AF, BC.
;
KONOUT:
        push    h
        push    d
        MOV     A,C
        ani     7fh     ;use Bit7 as Start bit (0)
        xri     0FFh    ;invert data
        RLC             ;adjust
        MOV     C,A
        mvi     B,7+1+1 ;7 bit data, 1 Start, 1 fudge

;       Write 8 bits.
; outer loop time:
;       N = 8
;       NT = (N * MT+6) + (N+1)*(37) - 3
;       N = 8
;       NT = 8*MT + 48+333-3
;               = 378 + 8*(43*TICK-3)
;               = 378 - 24 + (344*TICK)
;               = 354 + 344*TICK
;               = 10 + 344*(TICK+1)
; for each bit:
;       N1 = 43*M-3 + 6 + 37 - 3
;               = 37 + 43*M
;               = -6 + 43*(TICK+1)
proc1$n:
        lda     tick    ;(7     7
        MOV     E,A     ;(5     4
        MOV     A,C     ;(5     4
        RRC             ;(4     4
        MOV     C,A     ;(5     4
        dcr     B       ;(5     4
        JZ      proc1$6 ;if enuf bits read
        inx     h       ;(5     6

;       Begin output loop for this bit.
proc1$m:
        OUT     yBANG   ;(10    10      11)
        ani     80h     ;(7     7
        RLC             ;(4     4
        RRC             ;(4     4
        MOV     D,A     ;(5     4
```

```
        dcr     E       ;(5     4       4)
        JNZ     proc1$m ;(10    10(7)   10)
;                       ;(22*M
; inner loop=                   43*M-3
;                                       21*M)
        JMP     proc1$n
proc1$6:
        mvi     B,2     ;2 Stop bits
proc1$62:
        lda     TICK
        MOV     E,A
        inr     e
proc1$65:
        mvi     A,0     ;Stop bit= Mark= 1
        OUT     yBANG   ;(10    10      11)
        ani     80h     ;(7     7
        RLC             ;(4     4
        RRC             ;(4     4
        MOV     D,A     ;(5     4
        dcr     E       ;(5     4       4)
        JNZ     proc1$65        ;(10    10(7)   10)
        dcr     B
        JNZ     proc1$62        ;if more stoppers
        POP     D
        POP     H
        RET
;
;       Bit Banger Input.
;
;       Exit    A= Character read.
;                       Bit7 clear.
;       Uses    AF, BC, DE, HL.

;       Timing for      8080    8085    z80
KONIN:
        lxi     H,SAMP
        mvi     B,9+1

;       Wait for Start bit.
proc2$10:
        IN      yBANG
        RLC
        JNC     proc2$L0        ;if line still Marking

;       Now take 7 uniform samples.
;       The number of peeks in each determines the sample width.
proc2$n:
        lda     TICK    ;(13    13
        MOV     E,A     ;(5     4
        mov     m,d     ;(7     7
        mvi     d,0     ;(7     7
        dcr     B       ;(5     4
        JZ      proc2$6 ;(10)   if enuf bits read
        inx     H       ;(5     6
```

```
;       Begin sampling loop for this bit.
proc2$m:
        IN      yBANG   ;(10    10      11)
        ani     80h     ;(7     7
        RLC             ;(4     4
        ADD     D       ;(4     4
        MOV     D,A     ;(5     4
        dcr     E       ;(5     4       4)
        JNZ     proc2$M ;(10    10(7)   10)
        JMP     proc2$n ;(10    10      10)
;                       ;(22*M
; inner loop=                   43*M-3
;                                       21*M)


;       Reduce sample counts to data bits.
;       Note that due to DISK1 inversion,
;       0  Space =      Count[i]  >  Threshold
;       1  Mark  =      Count[i]  <  Threshold
;       Actually, all counts "near" mid-range are probably
;       errors.
proc2$6:
        lxi     h,SAMP+8        ;-> Data bit 6
        lda     TICK-1
        MOV     D,A     ;C= Threshold for Mark versus Space
        lxi     B,0700h
proc2$64:
        MOV     A,C
        RLC
        MOV     C,A
        mov     a,m
        CMP     D
        JNC     proc2$66        ;if large count
        inr     C       ;set bit for Mark
proc2$66:
        dcx     H
        dcr     B
        JNZ     proc2$64        ;if more bits to reduce
        MOV     A,C
        RET
;       Determine speed of terminal.
;       User must input a 'U'
;       Relies on being able to measure the width of the
;       Start bit.  Therefore, needs an odd-numbered Ascii
;       to be input.
;       Consecutive samples look like this:
;               ...1111111111110000..001xxxxxxx...
;               Mark            Space   Mark
;               ...Idle         Start   Data0   (ignore...)
baud:
        lxi     h,SAMP
        mvi     B,3
        lxi     D,1
```

```
;       Wait for Start bit.
proc3$10:
        IN      yBANG
        RLC
        JNC     proc3$L0        ;if line still Marking

;       Now measure width of next several pulses.
;       The number of peeks in each determines the sample width.
proc3$n:
        MOV     C,E     ;(5     4
        lda     TICK    ;(13    13
        mov     m,d     ;(7     7
        mvi     D,0     ;(7     7
        dcr     B       ;(5     4
        JZ      proc3$6 ;(10)   if enuf bits read
        inx     H       ;(5     6


;       Begin sampling loop for this bit.
proc3$m:
        inr     D       ;(5     4       4)
        IN      yBANG   ;(10    10      11)
        ani     80h     ;(7     7
        RLC             ;(4     4
        CMP     C       ;(4     4
        MOV     E,A     ;(5     4
        JZ      proc3$M ;(10    10(7)   10)
        JMP     proc3$N ;(10    10      10)
;                       ;(22*M
; inner loop=                   43*M-3
;                                       21*M)


;       Reduce sample counts to data bits.
proc3$6:
        lda     SAMP+1
        sta     TICK    ;set nominal bit width
        ora     a       ;clear carry
        RAR             ;width / 2 = threshold
        sta     TICK-1
        RET

        ENDIF

        IF      SYSSUP

;       CompuPro System Support 1 equates.

GBSS:   EQU     50h             ;System Support starting port
GBMP0:  EQU     GBSS+0          ;Master PIC port 0
GBMP1:  EQU     GBSS+1          ;Master PIC port 1
GBSP0:  EQU     GBSS+2          ;Slave PIC port 0
GBSP1:  EQU     GBSS+3          ;Slave PIC port 1
GBT0:   EQU     GBSS+4          ;Timer number 0
GBT1:   EQU     GBSS+5          ;Timer number 1
```

```
GBT2:   EQU     GBSS+6          ;Timer number 2
GBTC:   EQU     GBSS+7          ;Timer control port
GBFPPD: EQU     GBSS+8          ;Floating point processor data port
GBFPPC: EQU     GBSS+9          ;Floating point processor command port
GBCLKC: EQU     GBSS+10         ;Clock command port
GBCLKD: EQU     GBSS+11         ;Clock data port
GBUD:   EQU     GBSS+12         ;Uart data port
GBUS:   EQU     GBSS+13         ;Uart status port
GBUM:   EQU     GBSS+14         ;Uart modem port
GBUC:   EQU     GBSS+15         ;Uart command port

SSDAV:  EQU     00000010b       ;System Support Data Available
SSTBMT: EQU     00000001b       ;System Support Transmit Buffer Empty

;       page
;       C O N S O L E   I N I T I A L I Z A T I O N
;
;       This routine performs the initialization required by
;       the System Support UART.
;

sTINIT:
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, even, 2 stops
        OUT     GBUM            ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBUM            ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBUC            ;Set up command port
        RET
;
;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the
;       following values in the A register.
;
;       EXIT    A = 0 (zero), means no character
;               currently ready to read.
;
;               A = FFh (255), means character
;               currently ready to read.

sCONST:
        IN      GBUS            ;Input from port
        ANI     SSDAV           ;Mask data available
        RZ                      ;If data not available
        ORI     0FFH
        RET
;
;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing
;       the high order bit.  If no character currently ready to
;       read then wait for a character to arrive before returning.
;
```

```
;       EXIT    A = character read from terminal.

sCONIN:
        IN      GBUS            ;Get status from uart
        ANI     SSDAV
        JZ      sCONIN
        IN      GBUD
        ANI     7Fh
        RET
;
;       C O N S O L   O U T P U T
;
;       Send a character to the console.  If the console
;       is not ready to receive a character wait until
;       the console is ready.
;
;       ENTRY   C = ASCII character to output to console.

sCONOUT:
        IN      GBUS            ;Get uart status
        ANI     SSTBMT          ;Test if buffer empty
        JZ      sCONOUT
        MOV     A,C
        OUT     GBUD
        RET

        ENDIF

        IF      INTER3

;       CompuPro Interfacer 3 support routines

GBI3:   EQU     10h             ;Interfacer 3 Base address
GBI3D:  EQU     GBI3+0          ;Uart data location
GBI3S:  EQU     GBI3+1          ;Uart status
GBI3M:  EQU     GBI3+2          ;Uart mode register
GBI3C:  EQU     GBI3+3          ;Uart command register
GBI3U:  EQU     GBI3+7          ;Uart select register

GBI3DV: EQU     00000010b       ;Interfacer 3 Data Available
GBI3MT: EQU     00000001b       ;Interfacer 3 Transmit Buffer Empty
GBI3DS: EQU     10000000b       ;Interfacer 3 Data Set Ready
CON:    EQU     7               ;Interfacer 3 Console Select
PRN:    EQU     6               ;Interfacer 3 Printer Select
ULS:    EQU     5               ;Interfacer 3 UL1 Select

;       C O N S O L E   I N I T I A L I Z A T I O N
;
;       This routine performs the initialization required by
;       the Interfacer 3.
;

I3INIT:
        MVI     A,CON           ;Console select
        OUT     GBI3U           ;Select Uart 7
```

```
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, even, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port

        MVI     A,PRN           ;Printer Select
        OUT     GBI3U           ;Select Uart 0
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, even, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port

        MVI     A,ULS           ;User list 1 Select
        OUT     GBI3U           ;Select Uart 0
        MVI     A,11101110b     ;Async, 16x, 8 bits, no parity, even, 2 stops
        OUT     GBI3M           ;Set up mode register 1
        MVI     A,01111110b     ;9600 baud
        OUT     GBI3M           ;Set up mode register 2
        MVI     A,00100111b     ;Trans. on, dtr low, rec. on, no break,
                                ;       no reset, rts low
        OUT     GBI3C           ;Set up command port
        RET


;       C O N S O L E   S T A T U S
;
;       This routine samples the Console status and returns the
;       following values in the A register.
;
;       EXIT    A = 0 (zero), means no character
;               currently ready to read.
;
;               A = FFh (255), means character
;               currently ready to read.

I3CONST:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Input from port
        ANI     GBI3DV          ;Mask data available
        RZ                      ;If data not available
        ORI     0FFH
        RET


;       C O N S O L E   I N P U T
;
;       Read the next character into the A register, clearing
```

```
;       the high order bit.  If no character currently ready to
;       read then wait for a character to arrive before returning.
;
;       EXIT    A = character read from terminal.

I3CONIN:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Get status from uart
        ANI     GBI3DV
        JZ      I3CONIN
        IN      GBI3D
        ANI     7Fh
        RET



;       C O N S O L E   O U T P U T
;
;       Send a character to the console.  If the console
;       is not ready to receive a character wait until
;       the console is ready.
;
;       ENTRY   C = ASCII character to output to console.

I3CONOUT:
        MVI     A,CON
        OUT     GBI3U
        IN      GBI3S           ;Get uart status
        ANI     GBI3MT          ;Test if buffer empty
        JZ      I3CONOUT
        MOV     A,C
        OUT     GBI3D
        RET
;
;       L i s t   O u t p u t.
;
;       Send a character to the list device.  If the list
;       device is not ready to receive a character wait
;       until the device is ready.
;
;       ENTRY   C = ASCII character to be output.

I3LIST: LDA     IOBYTE          ;Get IOBYTE status
        ANI     0C0H            ;Check for UL1:
        SUI     0C0H
        MVI     A,ULS
        JZ      I3UL1
        MVI     A,PRN
I3UL1:  OUT     GBI3U
I3LST1: IN      GBI3S
        ANI     GBI3MT+GBI3DS
        SUI     GBI3MT+GBI3DS
        JNZ     I3LST1
        MOV     A,C
```

```
        OUT     GBI3D
        RET
;
;       L i s t   S t a t u s.
;
;       Return the ready status for the list device.
;
;       EXIT    A = 0 (zero), list device is not ready to
;               accept another character.
;               A = FFh (255), list device is ready to accept
;               a character.

I3LST:  LDA     IOBYTE
        ANI     0C0H            ;Ceck for UL1:
        SUI     0C0H
        MVI     A,ULS
        JZ      I3LS1
        MVI     A,PRN
I3LS1:  OUT     GBI3U
        IN      GBI3S
        ANI     GBI3MT+GBI3DS
        SUI     GBI3MT+GBI3DS
        MVI     A,0FFH
        RZ
        XRA     A
        RET

        ENDIF


;       File is BIOS4.asm
;       S E L E C T   D I S K   D R I V E
;
;       Select the disk drive for subsequent disk transfers and
;       return the appropriate DPB address.  This routine
;       diverges from the normal CP/M implementation of just
;       saving the disk selection value until the transfer is
;       performed.  This divergence is required because floppy
;       disks are a removable media and come in more than on
;       format.  This routine determines the correct format and
;       initializes the DPH with the appropriate values for the
;       format type.
;
;       ENTRY   C = disk delection value.
;               DE and 1 = 0, must determine disk type.
;                        = 1, drive type has been determined.
;
;       EXIT    HL = 0, if drive not selectable.
;               HL = DPH address if drive is selectable.
;
;               DPH is intialized for the appropriate floppy
;               disk format.

SELDSK: MOV     A,C
        CPI     NDSK
```

```
JNC     SELD1           ;If invalid drive
PUSH    D               ;Save drive selection mask
MVI     B,0

LXI     H,DSKMSK
DAD     B
MOV     A,M
STA     SEKDSK          ;Save selection code

MOV     L,C             ;Compute DPH address
MOV     H,B
DAD     H               ;*2
DAD     H               ;*4
DAD     H               ;*8
DAD     H               ;*16
LXI     D,DPBASE
DAD     D               ;HL = DPH address

LXI     D,5*2
XCHG
DAD     D               ;HL = DPH(DPB)
MOV     A,M
INX     H
MOV     H,M
MOV     L,A
XCHG                    ;DE = DPB address
DCX     D
LDAX    D
STA     SEKTYP          ;Save disk type
POP     D               ;Restore Drive selction mask
CPI     MAXFTP+1
RNC                     ;If hard disk

MOV     A,E
ANI     1               ;Mask selected bit
RNZ                     ;If drive previously selected

PUSH    H               ;Save DPH address
CALL    TREAD           ;Determine disk type
POP     H               ;Restore DPH address
JNZ     SELD1           ;If disk type not determined

STA     SEKTYP          ;Save disk type
XCHG
MOV     L,A             ;Move type
MVI     H,0
DAD     H               ;*2
DAD     H               ;*4
DAD     H               ;*8
DAD     H               ;*16
LXI     B,DPBS1+1
DAD     B
XCHG                    ;DE = DPB address
PUSH    H
LXI     B,5*2
```

```
        DAD     B               ;Compute DPH DPB address
        MOV     M,E             ;Set DPB address into DPH
        INX     H
        MOV     M,D
        ANI     0FEh            ;Remove sided bit
        MOV     E,A
        MVI     D,0
        LXI     H,XTABLE
        DAD     D
        XCHG
        POP     H
        LDAX    D               ;Set translation table
        MOV     M,A             ;address into DPH
        INX     H
        INX     D
        LDAX    D
        MOV     M,A
        DCX     H
        RET

SELD1:  LXI     H,0
        MOV     A,L
        STA     CDISK
        RET
;
;       H O M E
;
;       Return disk to home.  This routine sets the track number
;       to zero.  The current host disk buffer is flushed to the
;       disk.

HOME:
        CALL    FLUSH           ;Flush host buffer
        XRA     A
        STA     HSTACT          ;Clear host active flag
        STA     UNACNT          ;Clear sector count
        STA     SEKTRK
        STA     SEKTRK+1
        RET
;
;       S E T   T R A C K.
;
;       Set track number.  The track number is saved for later
;       use during a disk transfer operation.
;
;       ENTRY   BC = track number.

SETTRK:
        IF      Z80
        db      0edh,43h
        dw      sektrk
        ENDIF

        IF      C8080
        MOV     L,C
```

```
        MOV     H,B
        SHLD    SEKTRK
        ENDIF

        LHLD    UNATRK
        MOV     A,L
        XRA     C
        MOV     C,A
        MOV     A,H
        XRA     B
        ORA     C
        RZ                      ;If same track
;
;       Clear Unallocated block count (force pre-reads).

CUNACT: XRA     A               ;A = 0
        STA     UNACNT          ;Clear unallocated block count
        RET
;
;       Set the sector for later use in the disk transfer.  No
;       actual disk operations are perfomed.
;
;       Entry   BC = sector number.

SETSEC: MOV     A,C
        STA     SAVSEC          ;sector to seek
        RET
;
;       Set Disk memory address for subsequent disk read or
;       write routines.  This address is saved in DMAADR until
;       the disk transfer is performed.
;
;       ENTRY   BC = Disk memory address.
;
;       EXIT    DMAADR = BC.

SETDMA:
        IF      Z80
        db      0edh,43h
        dw      dmaadr
        RET
        ENDIF

        IF      C8080
        MOV     H,B
        MOV     L,C
        SHLD    DMAADR
        RET
        ENDIF
;
;       Translate sector number from logical to physical.
;
;       ENTRY   DE = 0, no translation required.
;               DE = translation table address.
;               BC = sector number to translate.
```

```
;
;       EXIT    HL = translated sector.

SECTRN:
        LDA     UNASEC
        CMP     C
        CNZ     CUNACT          ;If sectors do not match
        MOV     A,C
        STA     LOGSEC
        MOV     L,C
        MOV     H,B
        MOV     A,D
        ORA     E
        RZ                      ;If no translation
        DAD     D
        MOV     L,M
        MVI     H,0
        RET
;
;       S E T   S E C T O R   C O U N T
;
;       Set the number of continuous sectors to transfer.
;
;       ENTRY   C = Number of sectors to transfer.
;
;       EXIT    NUMSEC = C

SETNUM:
        MOV     A,C
        STA     NUMSEC
        RET
;
;       S E T   E X T E N D E D   B A N K
;
;       Set the extended bank data tranfer address.
;
;       ENTRY   C = Extended address bank.
;
;       EXIT    DMAADE = C.

SETXAD:
        MOV     A,C
        STA     DMAADE
        RET
;
;       B o o t   C P / M   f r o m   d i s k.
;
;       The CBOOT entry point gets control from the cold start
;       loader and is responsible for the basic system initial-
;       ization.  This includes outputting a sign-on message and
;       initializing the following page zero locations:
;
;          0,1,2: Set to the warmstart jump vector.
;              3: Set to the initial IOBYTE value.
;              4: Default and logged on drive.
```

```
;           5,6,7: Set to a jump to BDOS.
;              40: (Set by BOOT) Board switch options.
;
;       If BANG is true (DISK1 bit serial latch is to be
;       supported), then board switch option 1 means to
;       use the BitBanger for console I/O.
;       Register C must contain the selected drive, which is
;       zero to select the A drive.  The exit address is to
;       the CCP routine.
;
;
;       The WBOOT entry point gets control when a warm start
;       occurs, a ^C from the console, a jump to BDOS (function
;       0), or a jump to location zero.  The WBOOT routine reads
;       the CCP and BDOS from the appropriate disk sectors.
;       WBOOT must also re-initialize locations 0,1,2 and 5,6,7.
;       The WBOOT routines exits with the C register set to the
;       appropriate drive selection value.  The exit address
;       is to the CCP routine.
;
;
;       Disk layout Definition.
;       Cylinder 0   Head 0
;         0 thru 3       Boot program
;         4 thru 26      Reserved for CBIOS
;
;       256 byte sectors -- Cylinder 1  Head 0:
;         0 thru 7       CCP
;         8 thru 21      BDOS
;        22 thru 26      **Reserved for CP/M expansion**
;
;       512 byte sectors -- Cylinder 1  Head 0:
;         0 thru 3       CCP
;         4 thru 11      BDOS
;        12 thru 15      **Reserved for CP/M expansion**
;
;       1024 byte sectors -- Cylinder 1  Head 0:
;         0 thru 1       CCP
;         2 thru 5,7     BDOS
;         6              **Reserved for CP/M expansion**

CBOOT:  LXI     SP,DBUF
        lda     opts
        IF      BANG
        CPI     1
        JNZ     CBOOT4          ;if BitBanger not selected
        lxi H,KONST     ! shld J$CST+1
        lxi H,KONIN     ! shld J$CIN+1
        lxi H,KONOUT    ! shld J$COUT+1
        CALL    BAUD            ;get terminal speed
        JMP     CBOOTX
CBOOT4:
        ENDIF
        IF      SYSSUP
        CPI     2
```

```
        JNZ     CBOOT5          ;If System support not selected
        LXI H,sCONST    ! SHLD  J$CST+1
        LXI H,sCONIN    ! SHLD  J$CIN+1
        LXI H,sCONOUT   ! SHLD  J$COUT+1
        CALL    sTINIT          ;Initialize System Support
        JMP     CBOOTX
CBOOT5:
        ENDIF
        IF      INTER3
        CPI     3
        JNZ     CBOOT6          ;If Interfacer 3 not selected
        LXI H,I3CONST   ! SHLD  J$CST+1
        LXI H,I3CONIN   ! SHLD  J$CIN+1
        LXI H,I3CONOUT  ! SHLD  J$COUT+1
        LXI H,I3LIST    ! SHLD  J$LIST+1
        LXI H,I3LST     ! SHLD  J$LST+1
        CALL    I3INIT          ;Initialize Interfacer 3
CBOOT6:
        ENDIF
CBOOTX: LXI     H,SIGNON
        CALL    PRINT           ;Output Banner
        XRA     A
        STA     CDISK           ;Force A drive
        STA     IOBYTE          ;Clear I/O byte

WBOOT:                          ;Warm boot
        LXI     SP,DBUF
        CALL    BCPM            ;Boot CP/M
        JNZ     WBOOT           ;If error

        LXI     B,DBUF          ;Set default data transfer address
        CALL    SETDMA
        MVI     A,0C3h          ;Store jumps in low memory
        STA     0
        STA     5
        LXI     H,BIOS+3
        SHLD    1
        LXI     H,BDOS
        SHLD    6
        LDA     CDISK
        MOV     C,A
        JMP     CCP             ;Go to CPM

SIGNON:
        DB      CR,LF,LF
        DB      'CompuPro Systems '
        DB      CR,LF,MSIZE/10+'0',MSIZE mod 10 + '0'
        DB      'K CP/M ',VERS/10+'0','.',VERS mod 10 + '0'
        DB      CBIOSV+'@',CR,LF,0
;
;       Exit    A = 0, load sucessful.
;               Z bit = 1, load successful.

BCPM:   CALL    HOME            ;Force buffer flush
        mvi     C,'A'-'A'       ;Select Disk 'A'
```

```
        lxi     D,0             ;Force disk type determination
        CALL    SELDSK
        MOV     A,L
        ORA     H
        JZ      BCPME           ;If drive not selected
        LDA     SEKTYP          ;Get disk type
        RAR                     ;Remove sided bit
        DCR     A
        JM      BCPME           ;If invalid boot type

        MOV     C,A
        lxi     H,BSECT ;Boot sector table
        DAD     B
        mov     A,M             ;Get number of sectors
        STA     NUMSEC
        lxi     H,CCP           ;Set buffer address
        shld    DMAADR
        MVI     A,CXADR         ;Set extended address
        sta     DMAADE
        LDA     SEKTYP
        ani     1b              ;Mask sided bit
        Adi     1
        mvi     H,0
        MOV     L,A
        shld    SEKTRK  ;Set track
        MVI     A,0             ;Set boot sector
        sta     SAVSEC

        CALL    SETACT          ;Move SEK to ACT
        mvi     A,F$RDAT+040h   ;Read data
        CALL    FINAL
        JNZ     BCPME           ;If read erros
        mvi     A,1
        sta     NUMSEC
        LDA     SEKTYP
        cpi     dskd5
        JC      BCPM1           ;If not 1024 byte
        lxi     H,CCP+5*1024-512
        SHLD    BUFADR
        mvi     A,7
        STA     ACTSEC
        mvi     A,F$RDAT+040h   ;Read data
        JMP     FINAL

BCPM1:  XRA     A               ;Clear error indicator
        RET

BCPME:  lxi     B,500
        CALL    DELAY           ;Delay 500 milli-seconds
        ORI     1               ;Set error indicator
        RET

BSECT:  DB      22*256/256      ;double 256 byte sectors
        DB      22*256/512      ;double 512 byte sectors
        DB      22*256/1024     ;double 1024 byte sectors
```

```
        IF      HARD
        DB      22*256/256      ;SA 1000 hard disk
        DB      22*256/512      ;Memorex 8 inch
        ENDIF
;
;       Read a CP/M 128 byte sector.
;
;       EXIT    A = 0, successful read operation.
;               A = 1, unsucessful read operation.
;               Z bit = 1, successful read operation.
;               Z bit = 0, unsuccessful read operation.

READ:   CALL    CHKBKD          ;Check for blocked drive
        MVI     A,F$RDAT        ;Read from single density floppy
        JC      FINAL           ;If non-blocked transfer

        XRA     A               ;Set flag to force a read
        STA     UNACNT          ;Clear sector counter
        CALL    FILL            ;Fill buffer with data
        POP     H
        POP     D

        IF      C8080
        MVI     C,128
        CALL    MOVDTA          ;Move 128 bytes
        endif
        IF      Z80
        LXI     B,128           ;Move 128 bytes
        db      0edh,0b0h
        endif

        LDA     ERFLAG
        ORA     A
        RZ                      ;If no error
        XRA     A
        STA     HSTACT          ;Clear host active
        ORI     001h            ;Set error flag
        RET
;
;       Write the selected 128 byte CP/M sector.
;
;       ENTRY   C = 0, write to a previously allocated block.
;               C = 1, write to the directory.
;               C = 2, write to the first sector of unallocated
;               data block.
;
;       EXIT    A = 0, write was successful.
;               A = 1, write was unsucessful.
;               Z bit = 1, write was successful.
;               Z bit = 0, write was unsucessful.

WRITE:  CALL    CHKBKD          ;Check for blocked drive
        MVI     A,F$WRT         ;Write to single density floppy
        JC      FINAL           ;If non-blocked transfer
```

```
        MOV     A,C             ;Write type in c
        STA     WRTYPE
        CPI     WRUAL
        JNZ     WRIT2           ;If write to allocated
        LDA     SEKTYP
        CPI     MAXFTP
        MVI     A,2048/128
        JC      WRIT1           ;If floppy disk
        MVI     A,4096/128

WRIT1:  STA     UNACNT
        LHLD    SEKTRK
        SHLD    UNATRK          ;UNATRK = SEKTRK
        LDA     LOGSEC
        INR     A
        JMP     WRIT3

WRIT2:  LDA     UNACNT
        ORA     A
        JZ      WRIT4           ;If no unallocated records
        DCR     A
        STA     UNACNT
        LDA     SEKTYP
        RAR
        MOV     L,A
        MVI     H,0
        LXI     D,LSITT-1
        DAD     D
        LDA     UNASEC          ;Increment logical sector
        INR     A
        CMP     M               ;Last sector in track table
        JNZ     WRIT3           ;If not end of track
        LHLD    UNATRK
        INX     H
        SHLD    UNATRK
        XRA     A

WRIT3:  STA     UNASEC
        MVI     A,0FFh

WRIT4:  CALL    FILL
        POP     D
        POP     H

        IF      C8080
        MVI     C,128
        CALL    MOVDTA          ;Move 128 bytes
        endif
        IF      Z80
        LXI     B,128
        db      0edh,0b0h
        endif

        MVI     A,1
        STA     HSTWRT          ;HSTWRT = 1
```

```
        LDA     ERFLAG
        ORA     A
        RNZ                     ;If any errors occurred

        LDA     WRTYPE          ;write type
        CPI     WRDIR           ;to directory?
        CZ      FLUSH           ;Force write of directory
        LDA     ERFLAG
        ORA     A
        RET

LSITT:  DB      2*26            ;Double 256 byte
        DB      4*15            ;Double 512 byte
        DB       8*8            ;Double 1024 byte
        DB      2*32            ;Shugart 8 inch (256 byte)
        DB      4*21            ;Memorex 8 inch (512 bye)
;
;       TREAD - Determine floppy disk type.
;
;       ENTRY   C = Selected drive.
;
;       Exit    Zbit set = no error
;               A = disk type (0-3)

TREAD:  MOV     A,C
        ADI     'A'
        STA     NRDYM2          ;Set drive into message
        CALL    SPECIFY         ;Set disk parameters
        lxi     b,240           ;Time delay for selecting sides
        call    delay
        LDA     SEKDSK          ;Move drive to command buffer
        STA     ACTDSK          ;Set into ACTDSK
        lxi     H,DSTS
        mvi     B,DSTSL
        CALL    EXECP           ;Perform command
        mvi     B,1
        CALL    GCMPS           ;Get the one status byte
        ANI     020h            ;Mask ready bit
        JNZ     TRD1            ;If drive is ready
        lxi     H,NRDYM1
        CALL    PRINT
        ORI     0FFh            ;Clear zero flag
        RET

TRD1:   LDA     TEMPBF          ;Get status byte
        ANI     008h            ;Mask TS bit
        RRC
        RRC
        RRC
        STA     SEKTYP          ;Save sided flag
        lxi     H,RECAL ;Do a test seek
        mvi     B,LRECAL
        CALL    MOVETO          ;Process command
        RNZ                     ;If error
```

```
        mvi     A,2             ;Seek to track two
        CALL    DOSEEK          ;Do seek
        RNZ                     ;If error

        mvi     A,F$DRID
        STA     DRID
TRD2:   lxi     H,DRID
        mvi     B,DRIDL
        mvi     C,7
        CALL    EXECX           ;Process command
        JZ      TRD3            ;If read valid
        LDA     DRID
        XRI     040h            ;Compliment MFM bit
        STA     DRID
        ANI     040h
        JNZ     TRD2            ;If MFM not tried
        ORI     0FFh
        RET

TRD3:   LDA     TEMPBF+6        ;Get number of bytes
        ADD     A
        MOV     B,A
        LDA     SEKTYP
        ORA     B               ;Combine N with sided flag
        CMP     A               ;Set zero flag
        RET

DSTS:   DB      F$DSTS,0
DSTSL:  equ     $-DSTS

RECAL:  DB      F$RECA,0
LRECAL: equ     $-RECAL

DRID:   DB      F$DRID,0
DRIDL:  equ     $-DRID

NRDYM1: DB      CR,LF,'Drive '
NRDYM2: DB      'x'
        DB      ' not ready.',0
;
;       FILL - fill host buffer with approprite host sector.
;
;       ENTRY   A = 0, Read required if not in buffer.
;               Otherwise read not required.
;
;       EXIT    On exit the stack will contain the following
;               values:
;             POP     x         ;x = host record address.
;             POP     y         ;y = caller's buffer address.

FILL:   STA     RDFLAG          ;Save read flag
        LDA     SEKTYP          ;Get disk type
        CPI     MAXFTP+1
        JC      FILL1           ;If floppy disk
        SUI     DSK8S1-2
```

```
FILL1:  RRC                     ;divide by 2
        ANI     3h
        MOV     B,A             ;B = log base 2 (sector size) - 7
        LXI     D,HSTBUF        ;initial offset
        LXI     H,128           ;128 byte records
        LDA     SEKSEC          ;Get logical sector
FILL2:  XCHG
        RRC
        JNC     FILL3           ;If low bit not set
        DAD     D               ;Add bias to offset
FILL3:  XCHG
        DAD     H
        ANI     07Fh            ;Mask sector
        DCR     B
        JNZ     FILL2           ;If not all bits checked
        STA     SEKSEC
        LHLD    DMAADR
        XTHL                    ;Set return parameters
        PUSH    D
        PUSH    H               ;Set return address

        LXI     H,HSTACT        ;host active flag
        MOV     A,M
        MVI     M,1             ;always becomes 1
        ORA     A
        JZ      FILL6           ;If host buffer inactive
        LXI     H,HSTSEC
        LXI     D,SEKSEC
        MVI     C,SEKTYP-SEKSEC+1
FILL4:  LDAX    D
        CMP     M
        JNZ     FILL5           ;If mis-match
        INX     H
        INX     D
        DCR     C
        JNZ     FILL4           ;If all bytes not checked
        RET

FILL5:  CALL    FLUSH           ;Flush host buffer

FILL6:  LHLD    SEKDSK          ;Move disk and type
        SHLD    HSTDSK
        SHLD    ACTDSK
        LHLD    SEKTRK
        SHLD    HSTTRK
        SHLD    ACTTRK
        LDA     SEKSEC
        STA     HSTSEC
        STA     ACTSEC
        LDA     RDFLAG
        ORA     A
        RNZ                     ;If no read required

        MVI     A,F$RDAT+040h   ;Read double density
```

```
        JMP     BLKXFR
;
;       FLUSH - Write out active host buffer onto disk.

FLUSH:
        LXI     H,HSTWRT
        MOV     A,M
        ORA     A
        RZ                          ;If host buffer already on disk
        MVI     M,0
        LHLD    HSTDSK              ;Move disk and type
        SHLD    ACTDSK
        LHLD    HSTTRK
        SHLD    ACTTRK
        LDA     HSTSEC
        STA     ACTSEC
        MVI     A,F$WRT+040h        ;Write double density
;
;       BLKXFR -- blocked mode transfer.
;
;       ENTRY   A = command.

BLKXFR: MOV     C,A
        LXI     H,HSTBUF            ;Set buffer address
        SHLD    BUFADR
        MVI     A,BXADR
        STA     BUFADE
        MOV     A,C
;       JMP     FINAL
;
;       F I N A L   --  Preform final transfer processing.
;
;       ENTRY   A = Command.

FINAL:  CALL    PRCDCH              ;Process command, drive, cylinder
        lxi     H,CIOPB+0           ;Set buffer address
        mov     m,C                 ;Set command
        INX     H
        mov     m,B                 ;Set drive
        INX     H
        mov     m,E                 ;Set cylinder
        INX     H
        mov     m,D                 ;Set head
        INX     H
        MOV     E,A                 ;Save N field
        LDA     ACTSEC              ;Get sector
        MOV     C,A
        INR     A
        mov     m,A                 ;Set beginning sector
        INX     H
        MOV     A,E                 ;Get type
        CPI     4
        JP      HDFNL               ;If hard disk
        mov     m,A                 ;Set N field
        INX     H
```

```
        ADD     A               ;N*2
        ADI     CMDTYP and 0ffh
        MOV     E,A
        MVI     A,0
        ACI     CMDTYP/256
        MOV     D,A
        LDA     NUMSEC          ;Compute ending sector number
        ADD     C
        mov     m,A             ;Set EOT
        INX     H
        LDAX    d
        mov     m,A             ;Set GPL field
        INX     D
        INX     H
        LDax    d
        mov     m,A             ;Set DTL

        MVI     A,MRTRY         ;Set retry count
FNL1:   STA     RTRY            ;Clear retry count
        LDA     CIOPB+2         ;Get cylinder number
        CALL    DOSEEK          ;Seek to proper track
        JNZ     FNL3            ;If seek error

        lxi     H,BUFADE
        mvi     B,3
FNL2:   mov     A,m             ;get ext adr
        OUT     DMA
        Dcx     H               ;data is backward in memory
        dcr     B
        JNZ     FNL2            ;If not all 3 bytes

        lxi     H,CIOPB
        mvi     B,CIOPL         ;Set command buffer length
        mvi     C,7
        CALL    EXEC            ;perform operation
        Cpi     40h
        JNZ     FNL3            ;If error
        LDA     TEMPBF+1
        SUI     80h
        STA     ERFLAG
        RZ                      ;If no errors

FNL3:   LDA     RTRY            ;Get retry counter
        DCR     A
        JNZ     FNL1            ;If not permanent error
        ORI     01h
        STA     ERFLAG          ;Set error flag
        RET

;       HDFNL -- Hard disk final command processing.
;

HDFNL:
        IF      NOHRD
        CALL    HDSEL
```

```
        STA     ERFLAG
        RNZ                     ;If select error
        MVI     A,MRTRY         ;Set retry count
HDFNL1: STA     RTRY
        CALL    HDSEEK          ;Seek to correct track
        CALL    HDXFER          ;Perform hard disk transfer
        STA     ERFLAG
        RZ                      ;If no errors
        LDA     RTRY
        DCR     A
        JNZ     HDFNL1          ;If attempts left
        LDA     ACTDSK
        MVI     D,0
        MOV     E,A
        LXI     H,HDCYL
        DAD     D
        MVI     M,(-1)          ;Force track zero seek
        endif
        XRA     A
        ORI     001h
        STA     ERFLAG
        RET

RTRY:   DB      0
MRTRY:  EQU     10              ;Maximum retry count

;       Command buffer disk type dependent values.

CMDTYP: ;       GPL  DTL
        DB      007h,128        ;Single density
        DB      00Eh,255        ;Double density 256 bytes
        DB      01Bh,255        ;Double density 512 bytes
        DB      035h,255        ;Double density 1024 bytes
;
;       PRCDCH — Process Command, Drive, Cylinder, and Head.
;
;       ENTRY   A = command.
;
;       EXIT    A = N field (0..4).
;               B = drive.
;               C = command.
;               D = head.
;               E = cylinder.

PRCDCH: MOV     C,A             ;Save Command
        LDA     ACTDSK
        MOV     B,A
        LHLD    ACTTRK          ;Get track number
        LDA     ACTTYP          ;Get type
        CPI     MAXFTP+1
        JNC     CDCH2           ;If hard disk
        XCHG
        MOV     H,A             ;Save type
        ANI     1
        JZ      CDCH1           ;If single sided
```

```
        MOV     A,E
        ANI     1
        MOV     D,A             ;Set head
        RLC
        RLC
        ORA     B               ;Combine head with drive
        MOV     B,A
        MOV     A,E             ;Adjust track for cylinder
        RAR
        MOV     E,A

CDCH1:  MOV     A,H
        ANI     0FEh            ;Remove sided bit
        RRC
        RET

CDCH2:
        IF      NOHRD
        MOV     A,L
        ANI     003h
        MOV     D,A             ;Save head
        DAD     H               ;*2
        DAD     H               ;*4
        DAD     H               ;*8
        DAD     H               ;*16
        MOV     A,L             ;head * 16
        CMA
        ANI     030h
        ORA     B               ;Combine with drive
        MOV     B,A
        DAD     H               ;*32
        DAD     H               ;*64
        MOV     E,H             ;track*64/256 = track/4
        MOV     A,C
        ANI     00Fh
        CPI     F$RDAT
        MVI     A,4
        MVI     C,H$RSDT
        RZ                      ;If read command
        MVI     C,H$WSDT
        ENDIF
        RET
;
;       Seek to specified Track/Sector
;
;       Entry   A = Track

DOSEEK:
        sta     DSEKC+2
        lxi     H,DSEKC
        mvi     B,DSEKL


;       Move head according to command.
;
```

```
;       ENTRY   HL = address of command buffer.
;               B = length of command buffer.
;
;       Exit    Z bit set if no error.

MOVETO:
        CALL    EXECP           ;Perform seek
MVTO1:  IN      INTS
        ORA     A
        JP      MVTO1           ;if not complete
        mvi     A,F$RSTS
        OUT     FDCD            ;request status
        mvi     B,2
        CALL    GCMPS           ;Get status
        Cpi     20h
        RZ                      ;If seek complete
        LDA     TEMPBF          ;Get true status byte
        ANI     3h              ;Mask disk unit
        MOV     C,A
        LDA     ACTDSK
        CMP     C
        JNZ     MVTO1           ;If not proper unit
        ORI     001h            ;Clear zero flag
        RET
;
;       SPECIFY - Specify disk drive characteristics.

SPECIFY:
        lxi     H,SPEC+1
        mvi     B,LSPEC
        mvi     C,0
        LDA     STEPMS
        ORI     HUT
        mov     m,A
        dcx     H
        JMP     EXEC            ;Specify disk command

DSEKC   DB      F$SEEK,0,0
DSEKL:  EQU     $-DSEKC

SPEC    DB      F$SPEC
        db      (srt shl 4)+hut
        db      (hdlt shl 1)+nd
LSPEC   equ     $-SPEC
;
;       E X E C
;       Entry   HL = FWA of command buffer.
;               B  = # of bytes to output
;               C  = # of bytes for status
;
;       Exit    If C <> 0 then see GCMPS.

EXECP:  mvi     C,0             ;Set no status byte

EXECX:  INX     H
```

```
        LDA     ACTDSK          ;Set drive into command buffer
        MOV     M,A
        DCX     H

EXEC:
EXEC1:  IN      FDCS
        ORA     A
        JP      EXEC1           ;if no master ready bit
        mov     A,m             ;command byte
        OUT     FDCD            ;to controller
        INx     H
        DCR     B
        JNZ     EXEC1           ;if more bytes
        MOV     A,C             ;# of status bytes+1
        ORA     A
        RZ                      ;if no status bytes
        MOV     B,C             ;# of status bytes

EXEC2:  IN      INTS
        ORA     A
        JP      EXEC2           ;If operation not complete
;
;       Get completion status.
;
;       Entry   B= # of status bytes to read
;
;       Exit    TEMPBF = status bytes read in.
;               A = [TEMPBF] and 0F8h.
;               Flags set according to above value in A.

GCMPS:
        lxi     H,TEMPBF        ;Set status buffer address
GCMPS2: IN      FDCS
        ORA     A
        JP      GCMPS2          ;if not ready
        IN      FDCD            ;Get status byte
        mov     m,A
        INx     H
        Dcr     B               ;decrement counter
        JNZ     GCMPS2          ;wait until all done
        LDA     TEMPBF          ;Get first status byte
        ANI     0F8h
        RET
;
        If      C8080
;       MOVDTA  - Move data in memory.
;
;       ENTRY   C = number of bytes to move
;               DE = destination address.
;               HL = source address.

MOVDTA:
        MOV     A,M             ;Source character
        STAX    D               ;to destination
        INX     H
```

```
        INX     D
        DCR     C               ;loop 128 times
        JNZ     MOVDTA          ;If transfer not complete
        RET
        ENDIF
;
;       Check blocked disk transfer.
;
;       EXIT    Cbit set, unblocked device.
;               Cbit clear, blocked device.

CHKBKD:
        XRA     A
        STA     ERFLAG          ;Clear error flag
        LDA     SEKTYP
        CPI     DSKD1
        JC      CBKD2           ;If not blocked device
        CPI     MAXFTP+1
        JNC     CBKD1           ;If hard disk
        LHLD    SEKTRK
        MOV     A,H
        ORA     L
        MVI     A,DSKS1         ;A = Single density
        JZ      CBKD2           ;If zero force non-blocked

CBKD1   LDA     SAVSEC
        STA     SEKSEC
        XRA     A               ;Clear carry flag
        RET

SETACT: LDA     SEKTYP
CBKD2:  STA     ACTTYP          ;Set actual disk type
        LHLD    DMAADR
        SHLD    BUFADR
        LDA     DMAADE
        STA     BUFADE

        LDA     SEKDSK
        STA     ACTDSK
        LHLD    SEKTRK
        SHLD    ACTTRK
        LDA     SAVSEC
        STA     ACTSEC
        STC                     ;Set carry flag
        RET
;
;       Delay   Delay the millisecond count contained in BC.
;
;       Destroys  A and flags.

DELAY:  MVI     A,DELCNT/26
DLAY1:  DCX     B
        INX     B
        DCR     A
        JNZ     DLAY1           ;If not 1 millisecond
```

```
        DCX     B
        MOV     A,C
        ORA     B
        JNZ     DELAY           ;If not desired count
        RET
;
;       Print message terminated by zero byte.
;
;       ENTRY   HL -> message buffer, terminated by zero.
;
;       EXIT    HL -> zero byte + 1.
;               A = 0.
;               Z bit set.
;
;       Destroys only HL, Flags, and A registers.

PRINT:  MOV     A,M             ;Get a character
        ORA     A
        INX     H
        RZ                      ;If zero the terminate
        PUSH    B
        MOV     C,A
        CALL    J$COUT          ;Output to the console
        POP     B
        JMP     PRINT
;
;       Physical data buffer address ((DMAADR) or HSTBUF)

BUFADR: DW      0               ;Lower 16 bits (least, middle)
BUFADE: DB      0               ;Extended address

;       User data buffer address

DMAADR: DW      0               ;Lower 16 bits (least, middle)
DMAADE: DB      0               ;Extended address
;       space   4,10
;       BIOS blocking / deblocking flags.

HSTACT: DB      0               ;host active flag
HSTWRT: DB      0               ;host written flag
UNACNT: DB      0               ;unalloc rec CNT
UNATRK: DW      0               ;Track
UNASEC: DB      255             ;Sector
LOGSEC  DB      0               ;Logical sector
;       space   4,10
;       Area for storage of hard disk cylinders.

HDCYL:
        if      hard
        db      (-1),(-1),(-1),(-1)
        endif
;
;       CP/M disk work space.

ALV:    DS      ALVS
```

```
CSV:    DS      CSVS
;
;       Disk access information.
;       This area is organized into the following groups
;               sector number
;               track number
;               disk drive
;               drive type
;       Each of these groups has three cells for the
;       current disk request, ACTual disk transfer,
;       and active host disk.

SEKSEC: DS      1               ;Current request
SEKTRK: DS      2               ;Current request
SEKDSK: DS      1               ;Current request
SEKTYP: DS      1               ;Current disk's type

ACTSEC: DS      1               ;Actual transfer operation
ACTTRK: DS      2               ;Actual transfer operation
ACTDSK: DS      1               ;Actual transfer operation
ACTTYP: DS      1               ;Actual disk's type

HSTSEC: DS      1               ;Active host disk
HSTTRK: DS      2               ;Active host disk
HSTDSK: DS      1               ;Active host disk
HSTTYP: DS      1               ;Active disk's type
;       space   4,10
;       Disk transfer flags and counters.

RDFLAG: DS      1               ;Read flag
ERFLAG: DS      1               ;Error reporting
WRTYPE: DS      1               ;Write operation type

SAVSEC: DS      1               ;Save sector
NUMSEC: DS      1               ;Number of sectors

CIOPL:  EQU     9
CIOPB:  DS      CIOPL           ;Disk command buffer
TEMPBF: DS      8               ;Result status cells
ESPACE:


        ORG     LWAMEM-HSTSIZ-128

DIRBUF: DS      128             ;Directory buffer
HSTBUF: DS      HSTSIZ-1        ;Host buffer

END
```

```
;+      TITLE   'Sorcim ColdStart Boot for GBC DISK1.'
;BOOT   ColdStart Boot load for CP/m 2.2
;                       on CompuPro DISK1.
;
;       +----------------------+
;       |                      |
;       |       D i s k        |
;       |       B O O T        |
;       |                      |
;       +----------------------+
;       Copyright 1981, Sorcim Corp.
;
;       This product is a copyright program product of
;       SORCIM and is supplied for use with the GBC 171
;       series controllers.
;
;       CompuPro                        Sorcim Corp.
;       Oakland, CA                     Santa Clara, CA
;
;
;       Version number: 2.2F
;       Version date:   1981 March 31
;
;       The following code is supplied to customers who
;       purchase a hard/floppy disk system from CompuPro.
;       The following code is written onto track 0 sector 0
;       of the hard disk.  This routine is read into memory
;       at location 0 by the user's PROM.   This routine then
;       loads the rest of the system into memory.
;
;       The format of the Floppy Disk Boot sectors are as follows:
;
;                       Routine
;       T  Sector       Name
;
;       0  0 thru 3     Boot program (this routine)
;
;          4 thru 25    BIOS
;
;       1  0 thru 7     CCP
;
;          8 thru 21    BDOS
;
;         22 thru 25    reserved

;+OPARM:        EQU     *o              ;Capture 0 parameter
OPARM:  EQU     32                      ;Memory size in Kbytes, or
                                         Bios load address
VERS:   EQU     22

FALSE:  EQU     0
TRUE:   EQU     NOT FALSE
```

```
K:      EQU     1024
biosln: equ     1000h
;+      IF      OPARM < (64+2)  ;If absolute
option: set     false
        if      oparm/(64+1)
option: set     true
        endif
        if      not option

MSIZE:  EQU     OPARM           ;Size of CP/M memory
CBIOS:  EQU     MSIZE*K-biosln  ;Start of CP/M jump table
        ENDIF

;+      IF      OPARM > (64+1)  ;If PRL generation
        if      option
MSIZE:  EQU     (OPARM+0E00h)/K ;Size of CP/M memory
CBIOS:  EQU     OPARM           ;Start of CP/M jump table
        ENDIF
BDOS:   EQU     CBIOS-0E00h+6   ;Start of BDOS
CCP:    EQU     CBIOS-1600h     ;Start of CCP

OPTS:   EQU     40h             ;Option selections byte



;       Assembly Constants
FDPORT  EQU     0C0H            ;Base port address for Controller
FDCS    EQU     FDPORT          ;Status register
FDCD    EQU     FDPORT+1        ;Data register
DMA     EQU     FDPORT+2        ;Dma address (when write)
INTS    EQU     FDPORT+2        ;Status Register (when read)
;                               Input on port disables boot rom.
SER     EQU     FDPORT+3        ;Serial port

;+ DELCNT       =       5000    ;Delay count
DELCNT  EQU     5000            ;Delay count

;       Controller function definitions
;       Specify (00) command
;+ NSEC =       0               ;Sect verify number
;+ F.RTK        =       02      ;Read track
;+ F.SPEC       =       03      ;Specify
;+ F.DSTS       =       04      ;Drive status
;+ F.RDAT       =       06      ;Read data
;+ F.RECA       =       07      ;recalibrate
;+ F.RSTS       =       08      ;Read status
;+ F.SEEK       =       0Fh     ;Seek
NSEC    equ     0               ;Sect verify number
F$RTK   equ     02              ;Read track
F$SPEC  equ     03              ;Specify
F$DSTS  equ     04              ;Drive status
F$RDAT  equ     06              ;Read data
F$RECA  equ     07              ;recalibrate
F$RSTS  equ     08              ;Read status
F$SEEK  equ     0Fh             ;Seek
```

```
;+ SRT  =       16-8            ;= Shuggart 800s
SRT     equ     16-8            ;= Shuggart 800s
;               16-3            ;= Shuggart 850s
;               16-3            ;= Remex
;+ HUT: =       240/16          ;Head unload = 240 ms
;+ HLT: =       (35+1)/2        ;Head load = 35 ms
;+ ND:  =       00              ;Set DMA mode
HUT:    equ     240/16          ;Head unload = 240 ms
HDLT:   equ     (35+1)/2        ;Head load = 35 ms
ND:     equ     00              ;Set DMA mode
;+      space   4,10


;       Bootstrap load.
;       Do not change any addresses from here to START:
;       Entry   C= Board switches from ROM (0 .. 3)
        ORG     100h
BOOT:
        JMP     START           ;invariance
;+      space   4,10


;       Function data for controller to boot
DATA    DB      0               ;Extended
;+ ENTRY:       DB      high CBIOS
ENTRY:  DB      CBIOS shr 8
;+      DB      low CBIOS
        DB      CBIOS and 0ffh
;+ LDMA EQU     *-DATA
LDMA    EQU     $-DATA

;+ SPEC DB      F.SPEC
SPEC    DB      F$SPEC
;+      VFD     4\SRT,4\HUT
;+      VFD     7\HLT,1\ND
        db      srt shl (8-4) + hut
        db      (hdlt shl (8-7)) + nd
;+ LSPEC        =       *-SPEC
LSPEC   equ     $-SPEC


;+RECAL DB      F.RECA,0
;+LRECAL        =       *-RECAL
RECAL   DB      F$RECA,0
LRECAL  equ     $-RECAL


;+ READ:        DB      F.RDAT
READ:   DB      F$RDAT
        DB      0               ;hds,dsl,ds0
        DB      0               ;C = sector ID info
        DB      0               ;Head
        DB      5               ;Record (sector)
```

```
        DB      0               ;N
        DB      26              ;Read to end of track
        DB      7               ;GPL
        DB      128             ;DTL
;+LREAD =       *-READ
LREAD   equ     $-READ


START:
        MOV     A,C             ;save board options
;+      STO     A,OPTS
        sta     OPTS
RETRY:
;+      LDK     DE,DATA
        lxi     D,DATA
;+      LDK     B,LDMA
        mvi     B,LDMA

;       Output beginning DMA address

;+ ADDR:        LD      A,[de]
ADDR:   ldax    d
        OUT     DMA             ;set DMA
        INX     D
        DCR     B
        JNZ     ADDR            ;if NOT all 3 bytes


;       Load Specify Command

;+      LDK     B,LSPEC
        mvi     B,LSPEC
SPEC1:  IN      FDCS
;+      OR      A
        ora     A
        JP      SPEC1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
        DCR     B
        JNZ     SPEC1           ;if more bytes

;       Recalibrate drive

;+      LDK     B,LRECAL
        mvi     B,LRECAL
RCAL1:  IN      FDCS
;+      OR      A
        ora     A
        JP      RCAL1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
```

```
        DCR     B
        JNZ     RCAL1           ;if more bytes

RCAL2:  IN      INTS
        ORA     A
        JP      RCAL2           ;If not complete

;+      MVI     A,F.RSTS
        MVI     A,F$RSTS
        OUT     FDCD

RCAL3:  IN      FDCS
        ORA     A
        JP      RCAL3
        IN      FDCD
        SUI     20h
        MOV     C,A
RCAL4:  IN      FDCS
        ORA     A
        JP      RCAL4
        IN      FDCD
        ORA     C
        JNZ     START

;       Now set-up read command

;+      LDK     B,LREAD
        mvi     B,LREAD
READ1:  IN      FDCS
;+      OR      A
        ora     A
        JP      READ1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
        DCR     B
        JNZ     READ1           ;if more bytes

READ2:  IN      INTS
        ORA     A
        JP      READ2           ;If not complete

READ3:  IN      FDCS
        ORA     A
        JP      READ3
        IN      FDCD
        SUI     40h
        MOV     L,A
READ4:  IN      FDCS
        ORA     A
        JP      READ4
        IN      FDCD
        SUI     80h
        MOV     H,A
```

```
;+      LDK     B,7-2
        mvi     B,7-2
READ5:  IN      FDCS
;+      OR      A
        ora     A
        JP      READ5           ;if not ready
        IN      FDCD            ;read status
;+      DEC     B
        dcr     B
        JNZ     READ5           ;wait until all done

        MOV     A,L
        ORA     H
        JNZ     RETRY           ;if problems
;+      LD      HL,ENTRY
        lhld    ENTRY
        MOV     D,H
        MOV     H,L             ;reverse
        MOV     L,D
;+      JMP     [hl]            ;enter CBIOS
        pchl                    ;enter CBIOS


        END
```

```
;+      TITLE   'ROM Boot for CompuPro DISK1 Controller.
;GBCROM - Sorcim ROM Boot for CompuPro Disk Controller.
;
;       +----------------------+
;       |                      |
;       |    R O M   B O O T   |
;       |                      |
;       +----------------------+
;
;       COMPUPRO                          SORCIM Corp.
;       Oakland, CA                       Santa Clara, CA
;
;       Copyright 1981, Sorcim Corporation.
;
;       This product is a copyright program product of
;       Sorcim and is supplied for use with the
;       CompuPro IEEE 696 Floppy Disk Controller.
;
;       Version number: 2.2F
;       Version date:   81 March 31


;+ ROMFWA        EQU     *o * 100h
romfwa  equ     0
;       Assembly Constants
FDPORT  EQU     0C0H            ;Base port address for Controller
FDCS    EQU     FDPORT          ;Status register
FDCD    EQU     FDPORT+1        ;Data register
DMA     EQU     FDPORT+2        ;Dma address (when write)
INTS    EQU     FDPORT+2        ;Status Register (when read)

;       Input on port disables boot rom.
SER     EQU     FDPORT+3        ;Serial port

;+ DELCNT        =       5*1000          ;5 MHz processor
delcnt  equ     5*1000          ;5 Mhz processor

;       Controller function definitions
;       Specify (00) command
;+ NSEC =        0               ;Sect verify number
;+ F.RTK         =       02      ;Read track
;+ F.SPEC        =       03      ;Specify
;+ F.DSTS        =       04      ;Drive status
;+ F.RDAT        =       06      ;Read data
;+ F.RECA        =       07      ;recalibrate
;+ F.RSTS        =       08      ;Read status
;+ F.SEEK        =       0Fh     ;Seek
NSEC    equ     0               ;Sect verify number
F$RTK   equ     02              ;Read track
F$SPEC  equ     03              ;Specify
F$DSTS  equ     04              ;Drive status
F$RDAT  equ     06              ;Read data
F$RECA  equ     07              ;recalibrate
F$RSTS  equ     08              ;Read status
```

```
F$SEEK  equ     0Fh             ;Seek

;+ SRT  =       16-8            ;= Shuggart 800s
SRT     equ     16-8            ;= Shuggart 800s
;               16-3            ;= Shuggart 850s
;               16-3            ;= Remex
;+ HUT: =       240/16          ;Head unload = 240 ms
;+ HLT: =       (35+1)/2        ;Head load = 35 ms
;+ ND:  =       00              ;Set DMA mode
HUT:    equ     240/16          ;Head unload = 240 ms
HDLT:   equ     (35+1)/2        ;Head load = 35 ms
ND:     equ     00              ;Set DMA mode


        ORG     ROMFWA
;+      LOC     0
START:  JMP     ROM1            ;forced jump to location 3

;+ ROM1:        LDK     BC,20   ;Wait 20 MSec
ROM1:   lxi     b,20            ;Wait 20 MSec
;+      LDK     HL,ROM2         ;Return from delay
        lxi     h,ROM2          ;Return from delay
        JMP     DELAY

ROM2:   LXI     D,DATA

;       Output beginning DMA address

        MVI     B,LDMA
;+ ADDR:        LD      A,[de]
ADDR:   ldax    d
        OUT     DMA             ;set DMA
        INX     D
        DCR     B
        JNZ     ADDR            ;if NOT all 3 bytes


;       Load Specify Command

        MVI     B,LSPEC
SPEC1:  IN      FDCS
;+      OR      A
        ora     A
        JP      SPEC1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
        DCR     B
        JNZ     SPEC1           ;if more bytes

;       Recalibrate drive

        MVI     B,LRECAL
RCAL1:  IN      FDCS
```

```
;+      OR      A
        ora     A
        JP      RCAL1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
        DCR     B
        JNZ     RCAL1           ;if more bytes

RCAL2:  IN      INTS
        ORA     A
        JP      RCAL2           ;If not complete

        MVI     A,F$RSTS
        OUT     FDCD

;+      LDK     BC,250          ;Leave light on .25 seconds
        lxi     b,250           ;Leave light on .25 seconds
;+      LDK     HL,RCAL3        ;set return address
        lxi     h,RCAL3         ;set return address
        JMP     DELAY

RCAL3:  IN      FDCS
        ORA     A
        JP      RCAL3
        IN      FDCD
        SUI     20h
        MOV     C,A
RCAL4:  IN      FDCS
        ORA     A
        JP      RCAL4
        IN      FDCD
        ORA     C
        JNZ     ERROR           ;If error in recalibrate

;       Now set-up read command

        MVI     B,LREAD
READ1:  IN      FDCS
;+      OR      A
        ora     A
        JP      READ1           ;if no master ready bit
;+      LD      A,[de]          ;load command byte
        ldax    d               ;load command byte
        OUT     FDCD            ;to controller
        INX     D
        DCR     B
        JNZ     READ1           ;if more bytes

READ2:  IN      INTS
        ORA     A
        JP      READ2           ;If not complete

READ3:  IN      FDCS
```

```
        ORA     A
        JP      READ3
        IN      FDCD
        SUI     40h
        MOV     L,A
READ4:  IN      FDCS
        ORA     A
        JP      READ4
        IN      FDCD
        SUI     80h
        MOV     H,A

        MVI     B,7-2
READ5:  IN      FDCS
;+      OR      A
        ora     A
        JP      READ5           ;if not ready
        IN      FDCD            ;read status
;+      DEC     B
        dcr     B
        JNZ     READ5           ;wait until all done

        MOV     A,L
        ORA     H
        JZ      GOBOOT          ;If no error during read
;+      space   4,10


;       Error during read or recalibrate.

;+ ERROR:       LDK    BC,1000  ;Wait 1 second before retry
;+      LDK     HL,ROM2
ERROR:  lxi     b,1000          ;Wait 1 second before retry
        lxi     h,ROM2
;       JMP     DELAY
;+      space   4,10


;       Delay a period of time.
;
;       ENTRY   BC = number of milliseconds to delay.
;               HL = return address.

;+ DELAY:       LDK     A,DELCNT/26
DELAY:  mvi     A,DELCNT/26
DLAY1:  INX     B
        DCX     B
        DCR     A
        JNZ     DLAY1
        DCX     B
        MOV     A,B
        ORA     C
        JNZ     DELAY
        PCHL                    ;return to caller
;+      space   4,10
```

```
;       Function data for controller to boot
DATA    DB      0               ;Extended
;+      DB      high BOOT
        DB      boot shr 8
;+      DB      low BOOT
        DB      boot and 0ffh
;+ LDMA EQU     *-DATA
LDMA    EQU     $-DATA

;+ SPEC DB      F.SPEC
;+      VFD     4\SRT,4\HUT
;+      VFD     7\HLT,1\ND
SPEC    DB      F$SPEC
        db      (srt shl 4)+hut

        db      (hdlt shl (8-7))+nd
;+ LSPEC        =       *-SPEC
LSPEC   equ     $-SPEC

;+ RECAL        DB      F.RECA,0
;+ LRECAL       =       *-RECAL
RECAL   DB      F$RECA,0
LRECAL  equ     $-RECAL

;+ READ:        DB      F.RDAT
READ:   DB      F$RDAT
        DB      0               ;hds,ds1,ds0
        DB      0               ;C = sector ID info
        DB      0               ;Head
        DB      1               ;Record (first sector)
        DB      0               ;N
        DB      4               ;EOT (last sectors)
        DB      7               ;GPL
        DB      128             ;DTL
;+ LREAD        =       *-READ
LREAD   equ     $-READ


;       Next instruction disables rom, shadow
;       of next instruction must be jump to
;       execute boot code just loaded at 100h
;       ORG     ROMFWA+100h-4           ;Leave exact room
;+      DS      (100h-4) - *
        DS      (100h-4) - $
GOBOOT:
;+      LDK     C,high ROMFWA   ;pass board switch value to Boot
        mvi     c,romfwa shr 8  ;pass board switch value to Boot
        OUT     SER             ;must preceed Boot immediately
BOOT:
;+      ASSERT  BOOT = 100h


        END
```

IF YOU NEED ASSISTANCE ALWAYS CONTACT
YOUR **COMPUPRO DEALER FIRST**

CUSTOMER SERVICE INFORMATION

Our paramount concern is that you be satisfied with any Godbout CompuPro product. If this product fails to operate properly, it may be returned to us for service; see warranty information below.
If you need further information feel free to write us at:

Box 2355, Oakland Airport, CA 94614-0355

LIMITED WARRANTY INFORMATION

Godbout Electronics will repair or replace, at our option, any parts found to be defective in either materials or workmanship for a period of 1 year from date of invoice. Defective parts *MUST* be returned for replacement.

If a defective part causes a Godbout Electronics product to operate improperly during the 1 year warranty period, we will service it free (original owner only) if delivered and shipped at owner's expense to and from Godbout Electronics. If improper operation is due to an error or errors on the part of the purchaser, there may be a repair charge. Purchaser will be notified if this charge exceeds $50.00.

We are not responsible for damage caused by the use of solder intended for purposes other than electronic equipment construction, failure to follow printed instructions, misuse or abuse, unauthorized modifications, use of our products in applications other than those intended by Godbout Electronics, theft, fire, or accidents.

Return to purchaser of a fully functioning unit meeting all advertised specifications in effect as of date of purchase is considered to be complete fulfillment of all warranty obligations assumed by Godbout Electronics. This warranty covers only products marketed by Godbout Electronics and does not cover other equipment used in conjunction with said products. We are not responsible for incidental or consequential damages.

Prices and specifications are subject to change without notice, owing to the volatile nature and pricing structure of the electronics industry.